

Codes Over Absorption Channels

Zuo Ye and Ohad Elishco

Abstract

In this paper, we present a novel communication channel, called the absorption channel, inspired by information transmission in neurons. Our motivation comes from in-vivo nano-machines, emerging medical applications, and brain-machine interfaces that communicate over the nervous system. Another motivation comes from viewing our model as a specific deletion channel, which may provide a new perspective and ideas to study the general deletion channel.

For any given finite alphabet, we give codes that can correct absorption errors. For the binary alphabet, the problem is relatively trivial and we can apply binary (multiple-) deletion correcting codes. For single-absorption error, we prove that the Varshamov-Tenengolts codes can provide a near-optimal code in our setting. When the alphabet size q is at least 3, we first construct a single-absorption correcting code whose redundancy is at most $3 \log_q(n) + O(1)$. Then, based on this code and ideas introduced in [1], we give a second construction of single-absorption correcting codes with redundancy $\log_q(n) + 12 \log_q \log_q(n) + O(1)$, which is optimal up to an $O(\log_q \log_q(n))$.

Finally, we apply the syndrome compression technique with pre-coding to obtain a subcode of the single-absorption correcting code. This subcode can combat multiple-absorption errors and has low redundancy. For each setup, efficient encoders and decoders are provided.

I. INTRODUCTION

The field of molecular or chemical communication, which involves the use of chemical signals for communication, has gained popularity in recent years due to advances in nano-technology and the development of nano-machines. These small devices can perform various tasks such as computing, storing data, transmitting information, and measuring physical quantities, and can be connected together to form a nano-network. Nano-networks are expected to have significant potential in future medical technologies, such as being used as an effective drug delivery system [2], [3] or for detecting infections through monitoring the values of different molecules [4]–[6], [15].

However, the small size of nano-machines presents challenges for traditional forms of communication [7], leading to the development of chemical communication as an alternative [8], [9]. This allows nano-machines to directly communicate with and across the human nervous system using chemical signals [10]–[12]. There have been several communication models proposed and studied in this field [12]–[15], and in one practical application, researchers transferred information through an in-vivo nervous system and observed the response of nerves to different voltages and frequencies [16].

In this paper, we propose a new type of transmission channel called **absorption channels**, which are inspired by neural and chemical communication systems. Our goal is to model a communication channel between nano-machines located within a living organism that utilize the organism's nervous system for communication and data collection. While chemical communication systems have been analyzed from an information-theoretic perspective, no coding-theoretic framework has been proposed. Therefore, the models we present in this paper are adapted to a coding-theoretic framework and are analyzed from a coding-theoretic perspective.

An absorption error can be defined as follows: given a finite alphabet $\Sigma_q = \{0, 1, \dots, q-1\}$ and an n -length sequence $\mathbf{x} = x_1 x_2 \dots x_n \in \Sigma_q^n$, the transmission of \mathbf{x} through a single-absorption channel (which results in a single absorption error) produces an $(n-1)$ -length sequence $x_1 \dots x_{i-1} (x_i \oplus x_{i+1}) x_{i+2} \dots x_n \in \Sigma_q^{n-1}$ for some $1 \leq i \leq n-1$, where $a \oplus b \triangleq \min\{a+b, q-1\}$.

To better demonstrate the connection between absorption channels and neural communication channels, we provide a brief explanation of neuron activity (for a more detailed explanation of neurons, see [17, Ch. 8-11]). Every cell, including nerve cells, consists of a fluid and particles encased in a membrane that allows certain materials and particles to pass through for communication with the surrounding environment. Neurons, or nerve cells, have several parts: dendrites, cell body, axon, and axon terminals (as shown in Figure 1). The dendrites are thin, branching extensions of the cell body that receive signals from other cells, the cell body contains the nucleus and other organelles, the axon is a long, thin projection that carries signals away from the cell body, and the axon terminals are the ending points of the axon that transmit signals to other cells.

Neurons are specialized cells that transmit electrical and chemical signals within the nervous system (see Figure 1 for an illustration). To transmit a signal, a neuron generates an electrical charge, known as an action potential, which travels along the surface of the cell. Action potentials typically begin at the dendrites of a neuron. When a neuron receives input from another neuron, it may trigger an action potential, which is generated by the movement of ions across the cell membrane. Once triggered, the action potential travels down the length of the neuron, passing through the cell body and axon, to the axon terminal. In response, the axon terminal releases chemical signals, called neurotransmitters, which bind to receptors on

The authors are with the School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer Sheva, Israel. Email: {zuoy,ohadeli}@bgu.ac.il.

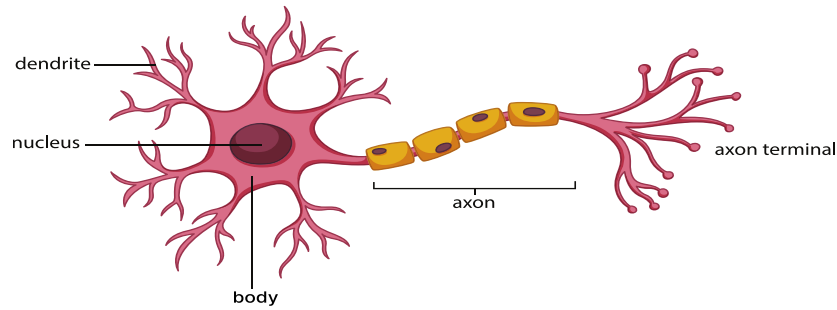


Figure 1. An illustration of a neuron (nerve cell) with its different parts: the dendrites, the cell body which contains the nucleus), the axon, and the axon terminals (downloaded from Vecteezy.com).

the dendrites of neighboring neurons. This transmission of the action potential from one neuron to another allows for the communication of information within the nervous system.

However, in some cases, an action potential may not be reached even if the neuron is depolarized by neurotransmitters. For example, if the neurotransmitters do not bind to enough receptors, the potential of the cell may increase, but not to the level required to trigger an action potential. This phenomenon is known as subthreshold stimulus¹.

In order for nano-machines to use nerve cells as communication channels, one machine should release neurotransmitters at the dendrites of a nerve cell, and another machine should detect the release of neurotransmitters from the nerve cell. The amount of neurotransmitters released can be used to represent symbols, such as small, medium, and large amounts representing 0, 1, and 2, respectively².

Neurotransmitters are chemical messengers that are produced within the cell body and then transported (by motor proteins) to the axon terminals, where they are stored until they are released in response to an action potential. However, there may be a shortage of neurotransmitters at the axon terminals due to their transport from the cell body, which can lead to a deficiency in the amount of neurotransmitters released when a neuron repeatedly fires [?]. This deficiency can result in the transmission of a different ("lower valued") symbol. Additionally, the production rate and quantity of neurotransmitters is influenced, among other things, by the depolarization of the cell, and an excess of neurotransmitters at the axon terminals may lead to the release of an excess amount and the transmission of a different ("higher") symbol.

The errors discussed above can occur in the context of communication between nano-machines using the nervous system as a transmission channel. If a symbol is to be transmitted while there is a deficiency of neurotransmitters, less neurotransmitters will be emitted and a "lower valued" symbol will be read. As a response to the deficiency, the neuron manufactures additional neurotransmitters. Thus, for the next transmission, an excess amount of neurotransmitters will be emitted and a "higher valued" symbol will be read. Similarly, if a transmission attempt depolarized the cell but not enough to reach an action potential, no neurotransmitters will be emitted (this corresponds to a deletion of the transmitted symbol). As a response to the depolarization, an additional amount of neurotransmitters is manufactured. Thus, in the next transmission attempt, an excess amount of neurotransmitters will be emitted and a "higher valued" symbol will be read. In this paper, we chose to focus only on the second error, in which a symbol is deleted and its value is added to the next transmission.

Mathematically, these observations give rise to a family of communication channels. Let us consider the transmission of a string $\mathbf{x} \in \Sigma_q^n$ and the received string \mathbf{y} . An error in the i th position can be described as follows: if the value of y_i is smaller ($y_i < x_i$), then the missing value is added to the next symbol, meaning $y_{i+1} = \min(q-1, x_{i+1} + (y_i - x_i))$. Alternatively, the i th symbol may be deleted completely ($\mathbf{y} \in \Sigma_q^{n-1}$) and its value added to the next symbol, so $y_i = \min(q-1, x_i + x_{i+1})$. In this work, we focus on the simplified case in which only the second error may occur, namely, the symbol is deleted and its entire value is added to the next symbol.

In addition to being motivated by neural communication systems, a single-absorption error can also be viewed as a deletion error followed by at most one substitution error. The study of codes that correct single-deletion and single-substitution errors was first introduced in the context of DNA-based data storage in [18] and further developed in [19]. More recently, codes that correct multiple-deletion and multiple-substitution errors were proposed in [20]. These results apply to our error model

¹In most mammals, the resting potential of a neuron is -70mV. This refers to the electrical potential across the cell membrane of the neuron when it is not actively transmitting an action potential. In order to fire, or transmit an action potential, a neuron must reach a potential of -50 mV. If this threshold is reached, the neuron will undergo a series of changes in ion concentrations that result in the rapid depolarization of the cell membrane. During this process, the potential of the cell increases to +30 mV before returning to the resting potential of -70 mV. This rapid change in potential, known as the action potential (or firing), allows for the transmission of information within the nervous system. If a neuron does not reach a potential of -50 mV, it will not fire an action potential. In this case, the neuron may be more excitable for a period of time after the failed attempt. This phenomenon is known as post-inhibitory rebound (see [?], [17]).

²Neurons are not found individually, but rather as a group or tissue. In order to utilize the communication capabilities of a neuron, it is necessary to isolate a single neuron from the tissue and use it as a standalone communication channel. If an entire (healthy) tissue is activated, it can result in unintended changes or effects on the body.

as well, but in this paper we demonstrate that it is possible to use specific absorption properties to achieve higher rates in our codes.

We also consider a variant of absorption errors called contraction errors, which we show are equivalent to deletion errors. The problem of constructing deletion-correcting codes dates back at least to the 1960s [21]. Recently, there has been renewed interest in this problem due to its potential applications in DNA-based data storage [22], [23] and document exchange [24], [25]. Despite significant progress, constructing deletion-correcting codes remains a challenging problem with no complete solution. Our new findings may provide new insights into this problem.

The paper is organized as follows: in Section II, we introduce the notation and definitions that will be used throughout the paper. In Section III, we present codes over the binary alphabet. Section IV contains the main results of this paper, which is the construction of absorption error-correcting codes for general alphabets. In Section V, we show that our single-absorption codes are asymptotically optimal in terms of redundancy. In Section VI, we study contraction errors as a variant of absorption errors and show that they are equivalent to deletion errors. Finally, in Section VII, we conclude the paper.

II. PRELIMINARY

For positive integers $m \leq n$, let $[m, n]$ denote the set $\{m, m+1, \dots, n\}$ and $[n] = \{1, \dots, n\}$. For an integer $q \geq 2$, let Σ_q denote the q -ary alphabet $\{0, 1, \dots, q-1\}$ and Σ_q^n denote the set consisting of all length- n sequences over Σ_q . For any sequence $\mathbf{x} \in \Sigma_q^n$, unless otherwise stated, we let x_i be the i th component of \mathbf{x} . In other words, $\mathbf{x} = x_1 \cdots x_n$. Suppose that two positive integers n and n' satisfy $n \geq n'$. Let $\mathbf{x} \in \Sigma_q^n$ and $\mathbf{y} \in \Sigma_q^{n'}$. If there are integers $1 \leq i_1 < i_2 < \dots < i_{n'} \leq n$ such that $y_j = x_{i_j}$ for each $1 \leq j \leq n'$, we say that \mathbf{y} is a *subsequence* of \mathbf{x} . If $I = \{i_1, i_2, \dots, i_{n'}\}$ (keep the order of $i_1, i_2, \dots, i_{n'}$), we also denote this subsequence by \mathbf{x}_I . Furthermore, if $i_{j+1} = i_j + 1$ for all $1 \leq j < n'$, we call \mathbf{y} a *substring* of \mathbf{x} . A *run* of \mathbf{x} is a maximal substring consisting of identical symbols from Σ_q . If a run consists of symbol a , we say it is an *a-run*. In this paper, the length of a sequence \mathbf{x} is denoted by $|\mathbf{x}|$.

Example II.1 Let $\mathbf{x} = 001112 \in \Sigma_3^6$, $\mathbf{y} = 012$ and $\mathbf{z} = 0111$. Then \mathbf{y} is a subsequence of \mathbf{x} and \mathbf{z} is a substring of \mathbf{x} . Specifically, we have $\mathbf{y} = \mathbf{x}_I$ and $\mathbf{z} = \mathbf{x}_J$, where $I = \{1, 3, 6\}$ and $J = \{1, 3, 4, 5\}$. There are exactly three runs in \mathbf{x} : 00, 111 and 2. They are 0-run, 1-run and 2-run, respectively.

For $a, b \in \Sigma_q$, we define $a \oplus b = \min\{a + b, q - 1\}$. Notice that \oplus is an associative operation thus the order in which it is performed does not affect the result. Suppose $\mathbf{x} \in \Sigma_q^n$. We say that the sequence $\mathbf{y} \in \Sigma_q^{n-1}$ is obtained from \mathbf{x} by an *absorption* if \mathbf{y} is either one of the following two cases:

- (1) $\mathbf{y} = x_1 \cdots x_{i-1}(x_i \oplus x_{i+1})x_{i+2} \cdots x_n$ for some $1 \leq i \leq n-1$;
- (2) $\mathbf{y} = x_1 \cdots x_{n-1}$.

When the second case happens, we say that x_n is *missing*. Otherwise, we say that x_n is *not missing*.

For multiple absorptions, the situation becomes a little more complicated. For example, let $\mathbf{x} \in \Sigma_q^n$, then $\mathbf{y}_1 = x_1 \cdots x_{i-1}(x_i \oplus x_{i+1})x_{i+2} \cdots x_{j-1}(x_j \oplus x_{j+1})x_{j+2} \cdots x_n$ where $i+2 \leq j < n$, and $\mathbf{y}_2 = x_1 \cdots x_{i-1}(x_i \oplus x_{i+1})x_{i+2} \cdots x_{n-1}$ where $i < n-1$ are both obtained from \mathbf{x} by two absorptions. Now let $\mathbf{y}_3 = x_1 \cdots x_{i-1}(x_i \oplus x_{i+1} \oplus x_{i+2})x_{i+3} \cdots x_n$ where $i < n-1$. It is clear that \mathbf{y}_3 can be obtained from \mathbf{x} by first absorbing x_i and x_{i+1} , and then absorbing $(x_i \oplus x_{i+1})$ and x_{i+2} .³ Therefore, the sequence \mathbf{y}_3 is also obtained from \mathbf{x} by two absorptions. In general, we have the following definition.

Definition II.2 Let $\mathbf{x} \in \Sigma_q^n$ and $\mathbf{y} \in \Sigma_q^{n-t}$ where $n > t \geq 1$. We say that \mathbf{y} is obtained from \mathbf{x} by t absorptions, if there is an integer $t' \in [0, t]$ and positive integers k, s_l ($1 \leq l \leq k$) and i_l ($1 \leq l \leq k$) satisfying $i_{l+1} - i_l > s_l$ for all $1 \leq l < k$ such that $\sum_{l=1}^k s_l = t - t'$, $i_k + s_k \leq n - t'$ and

$$y_i = \begin{cases} x_i, & \text{if } i < i_1, \\ x_{i+\sum_{j=1}^l s_j}, & \text{if } i_l - \sum_{j=1}^{l-1} s_j < i < i_{l+1} - \sum_{j=1}^l s_j \\ & \text{for } 1 \leq l < k, \\ x_{i+\sum_{j=1}^k s_j}, & \text{if } i > i_k - \sum_{j=1}^{k-1} s_j, \\ x_{i+s_l}, & \\ \bigoplus_{j=i_l} x_j, & \text{if } i = i_l - \sum_{j=1}^{l-1} s_j \text{ for } 1 \leq l \leq k. \end{cases}$$

Here, the substring $\mathbf{x}_{[n-t'+1, n]}$ is deleted.

In Definition II.2, the starting positions of the absorptions are denoted by the i_l s, while the number of symbols absorbed with x_{i_l} is denoted by s_l .

Example II.3 Let Σ_3 be the ternary alphabet and let $\mathbf{x} = 01101111$. Assume there are $t = 3$ absorptions, with $t' = 0$, $k = 2$, $s_1 = 2$, $s_2 = 1$ and $i_1 = 2$, $i_2 = 6$. The resulting sequence is $\mathbf{y}_1 = 021211$.

Now assume $t' = 1$, $k = 1$, $s_1 = 2$, and $i_1 = 2$, then $\mathbf{y}_2 = 021111$ with x_n missing.

³or by first absorbing x_{i+1} and x_{i+2} , and then absorbing x_i and $(x_{i+1} \oplus x_{i+2})$.

For a sequence $\mathbf{x} \in \Sigma_q^n$ and a positive integer t satisfying $t < n$, we define the set

$$\mathcal{B}_t^{ab}(\mathbf{x}) \triangleq \{\mathbf{y} \in \Sigma_q^{n-t} : \mathbf{y} \text{ is obtained from } \mathbf{x} \text{ by } t \text{ absorptions}\} \quad (1)$$

and call it the t -absorption ball centered at \mathbf{x} . Note that $\mathcal{B}_t^{ab}(\mathbf{x})$ depends on the alphabet Σ_q . We omit q in this notation since the alphabet will be clear from the context.

Definition II.4 Let t be a positive integer. Let \mathcal{C} be a nonempty subset of Σ_q^n . If $\mathcal{B}_t^{ab}(\mathbf{x}) \cap \mathcal{B}_t^{ab}(\mathbf{y}) = \emptyset$ for any distinct $\mathbf{x}, \mathbf{y} \in \mathcal{C}$, we call it a t -absorption correcting code. The redundancy of \mathcal{C} is defined to be $n - \log_q(|\mathcal{C}|)$. In other words, the redundancy is measured in q -ary symbols.

In this paper, we aim to construct t -absorption correcting codes with low redundancy, for any t . Throughout this paper, the number of errors t and the alphabet size q are assumed to be fixed constants.

III. CODES OVER BINARY ALPHABET

In this section we present a construction of a binary code that can repair multiple absorptions. The construction relies on the following simple observation.

Observation III.1 Suppose that \mathbf{y} is obtained from $\mathbf{x} \in \Sigma_2^n$ by absorbing x_i and x_{i+1} . If $x_i x_{i+1} \in \{00, 01, 10\}$, then \mathbf{y} is obtained from \mathbf{x} by deleting one 0. If $x_i x_{i+1} = 11$, then \mathbf{y} is obtained from \mathbf{x} by deleting one 1. Therefore, no matter whether the last symbol x_n is lost or not, \mathbf{y} is obtained from \mathbf{x} by deleting one symbol.

Notice that when at most one absorption occurs, an isolated 1 cannot be deleted, i.e., any 1 that both of its neighbors are 0, will not be deleted.

From Observation III.1, we have that every binary single-deletion correcting code is a binary single-absorption correcting code. The opposite, however, is not necessarily true, as shown in the next example.

Example III.2 We give an example to show that a single-absorption correcting code is not necessarily a single-deletion correcting code. Consider the code $\{011000, 011010\}$. A single absorption on 011000 yields 3 possible outputs as before: 11000, 01000, 01100. A single absorption on 011010 yields also 3 possible outputs: 11010, 01010, 01110. However, the sequence 01100 can be obtained from both codewords by deleting the one-before-last symbol. Thus, the code cannot correct a single deletion.

To construct a single-absorption correcting code, we can use Observation III.1 and apply single-deletion correcting codes. The best-known class of binary single-deletion correcting codes are the famous Varshamov-Tenengolts (VT) codes [26], which are defined as

$$\text{VT}_a(n) = \{\mathbf{c} \in \Sigma_2^n : \text{Syn}(\mathbf{c}) \equiv a \pmod{n+1}\}, \quad (2)$$

where a is an integer between 0 and n , and $\text{Syn}(\mathbf{c}) \triangleq \sum_{i=1}^n i c_i$ is the VT syndrome of \mathbf{c} . The smallest redundancy of $\log_2(n+1)$ is attained when $a = 0$ [27, Corollary 2.3]. A linear-time decoding algorithm of the VT codes to correct a single deletion was provided in [21]. In [28] the authors gave a linear-time systematic encoder with redundancy $\lceil \log_2(n+1) \rceil$.

Remark III.3 By Example III.2, one can deem that there might be a single-absorption correcting code of length n with a larger size than that of the VT code $\text{VT}_0(n)$. In Section V, we will show that for single-absorption, the redundancy of the code $\text{VT}_0(n)$ is optimal up to a constant.

By Definition II.2, it is not difficult to see that Observation III.1 can be generalized to the case when multiple absorptions happen. To be specific, if \mathbf{y} is obtained from \mathbf{x} by t absorptions, then it is obtained from \mathbf{x} by t deletions. So we can apply multiple-deletion correcting codes for our setting. There are already a myriad of works on binary multiple-deletion correcting codes (see, for example, [20], [22], [23], [29]–[31]). For $t = 2$, the best known result was given in [23], where an explicit binary 2-deletion correcting code of length n with redundancy at most $4 \log_2(n) + O(\log_2 \log_2(n))$ was constructed. This code is polynomial-time encodable and decodable. For general $t \geq 3$, the best known result was contributed in [20], where the authors proved that there is a binary systematic t -deletion correcting code of length n with redundancy at most $(4t - 1) \log_2(n) + o(\log_2(n))$. The encoding and decoding complexities are $O(n^{2t+1})$ and $O(n^{t+1})$ respectively.

IV. CODES OVER NON-BINARY ALPHABETS

In Section III, we showed that a single-absorption error is a special case of single-deletion error. The situation is different when the alphabet size is at least 3. Throughout this section, we always assume that the alphabet is Σ_q , where $q \geq 3$. This section contains three parts. At first, we present a basic code construction that can correct a single absorption. In the second part, we improve upon the basic construction and present a construction with smaller redundancy that can correct a single absorption error. In the last part, we study codes that can correct multiple absorptions.

A. A basic construction

We begin with a construction of a single absorption correcting code.

For a sequence $\mathbf{x} \in \Sigma_q^n$ and a symbol $a \in \Sigma_q$, we let $N_a(\mathbf{x})$ to be the number of a appearing in \mathbf{x} , that is,

$$N_a(\mathbf{x}) \triangleq |\{i : x_i = a\}|.$$

Let $\mathbf{y} = x_1 \cdots x_{i-1}(x_i \oplus x_{i+1})x_{i+2} \cdots x_n$ be the received sequence, where $1 \leq i \leq n-1$.

Observation IV.1 *Let $a, b \in \Sigma_q$ and $0 < a, b < q-1$.*

- *If $x_i x_{i+1} \in \{0a, a0, 00\}$, then $N_0(\mathbf{x}) = N_0(\mathbf{y}) + 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq 0$. In other words, \mathbf{y} is obtained from \mathbf{x} by deleting one 0.*
- *If $x_i = x_{i+1} = q-1$, then $N_{q-1}(\mathbf{x}) = N_{q-1}(\mathbf{y}) + 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq q-1$. In other words, \mathbf{y} is obtained from \mathbf{x} by deleting one $q-1$.*
- *If $x_i x_{i+1} \in \{(q-1)a, a(q-1)\}$, then $N_a(\mathbf{x}) = N_a(\mathbf{y}) + 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq a$. In other words, \mathbf{y} is obtained from \mathbf{x} by deleting one a .*
- *If $x_i x_{i+1} = ab$ and $c = a \oplus b$, then $N_a(\mathbf{x}) = N_a(\mathbf{y}) + 1$, $N_b(\mathbf{x}) = N_b(\mathbf{y}) + 1$, $N_c(\mathbf{x}) = N_c(\mathbf{y}) - 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq a, b, c$ (if $a = b$ then $N_a(\mathbf{x}) = N_a(\mathbf{y}) + 2$).*

From Observation IV.1, we can see that if $0 < x_i, x_{i+1} < q-1$, a single-absorption error is a single-deletion together with a single-substitution, which is different from the binary case. In general, a single-absorption error is a single-deletion together with at most a single-substitution (no matter whether the last symbol is missing or not). Therefore, if a code can combat a single-deletion together with at most a single-substitution, it can also correct a single-absorption error. In [19], the study of single-deletion single-substitution codes was initiated, and the authors gave a q -ary single-deletion single-substitution correcting code of redundancy at most $10 \log_2(n) + O(1)$ (measured in bits) [19, Corollary 17]. For more details about this kind of codes, we refer the interested readers to [19] and [20].

At this point, one may wonder if a single-absorption error correcting code is also a single-deletion single-substitution error correcting code. To answer this, we first notice that the substitution caused by an absorption is specific and depends on the absorbed symbol. Thus, it is reasonable to assume that a single-absorption error is a specific case of a single-deletion single-substitution error. Indeed, as shown in the next example, a single-absorption correcting code is not necessarily a single-deletion single-substitution correcting code.

Example IV.2 *Let $q = 3$ and consider the code $\{110110, 011010\}$. A single absorption error on 110110 yields one of the following words: 20110, 11110, 11020, 11011; a single absorption error on 011010 yields one of the following words: 11010, 02010, 01110, 01101. Therefore, this code can correct a single-absorption error. On the other hand, the sequence 11010 can be obtained from 110110 and from 011010 by a single deletion. So this code cannot correct a single-deletion and at most a single-substitution error.*

Thus, one may infer that there might be codes with lower redundancy for absorption channels. In this section, we show that indeed it is possible to obtain codes with less redundancy.

We begin with constructing a set of n -length words over Σ_q , which is defined by a vector \mathbf{s} of length $(q-1)$ over \mathbb{Z}_4 . Given $\mathbf{s} = (s_a)_{a \in [0, q-2]} \in \mathbb{Z}_4^{q-1}$, we define

$$\mathcal{C}_1(n; \mathbf{s}) \triangleq \{\mathbf{x} \in \Sigma_q^n : N_a(\mathbf{x}) \equiv s_a \pmod{4} \text{ for each } a \in [0, q-2]\}.$$

Since $N_{q-1}(\mathbf{x}) = n - \sum_{a=0}^{q-2} N_a(\mathbf{x})$, we can obtain $N_{q-1}(\mathbf{x}) \pmod{4}$ when given all $N_a(\mathbf{x}) \pmod{4}$ where $a \in [0, q-2]$. Assume a single absorption channel, and suppose that a transmitted sequence \mathbf{x} is in $\mathcal{C}_1(n; \mathbf{s})$. Denote the obtained sequence (the channel output) by \mathbf{y} . Since \mathbf{y} is obtained from \mathbf{x} by a single absorption, this absorption must be one of the four cases described in Observation IV.1. By calculating $N_a(\mathbf{y}) - s_a \pmod{4}$ for all $a \in \Sigma_q$,⁵ it is possible to know which one of the cases happened (without knowing the position in which the absorption happened). The details are shown in Table I. Hereafter, let $\{\{\cdot\}\}$ denote a multiset.

Thus, if a sequence $\mathbf{x} \in \mathcal{C}_1(n; \mathbf{s})$ is transmitted through a single absorption channel and \mathbf{y} is the output of the channel, it is possible to distinguish which one of the four absorption cases described in Observation IV.1 has occurred. However, more information is needed in order to recover \mathbf{x} from \mathbf{y} . For example, the order of the absorbed symbols (if $a \oplus b = c$ then also $b \oplus a = c$), or the exact position of the absorption. Therefore, we need to add additional redundancy layers to $\mathcal{C}_1(n; \mathbf{s})$ as explained next.

To account for the order of the absorbed symbols, let us first consider the case $x_i x_{i+1} \in \{\{ab, ba\}\}$ where $0 < a, b < q-1$ and a, b are not necessarily distinct. As mentioned above, by calculating $N_d(\mathbf{y}) - s_d \pmod{4}$ for all $d \in \Sigma_q$, one can deduce

⁴As will be clear later, our code can correct a single deletion. So we do not need to discuss the case $\mathbf{y} = \mathbf{x}_{[1, n-1]}$

⁵where $s_{q-1} \in \{0, 1, 2, 3\}$ and satisfies $s_{q-1} \equiv n - \sum_{a=0}^{q-2} s_a \pmod{4}$

TABLE I
THE RELATION BETWEEN $N_a(\mathbf{y}) - s_a$ ($a \in \Sigma_q$) AND THE VALUES OF x_i AND x_{i+1}

Cases	The values of x_i and x_{i+1}
$N_0(\mathbf{y}) - s_0 \equiv 3 \pmod{4}$ and $N_a(\mathbf{y}) - s_a \equiv 0 \pmod{4}$ for all $a \neq 0$	$0 \in \{\{x_i, x_{i+1}\}\}$
$N_a(\mathbf{y}) - s_a \equiv 3 \pmod{4}$ for some $a \neq 0$ and $N_b(\mathbf{y}) - s_b \equiv 0 \pmod{4}$ for all $b \neq a$	$\{\{x_i, x_{i+1}\}\} = \{\{a, q-1\}\}$
$N_a(\mathbf{y}) - s_a \equiv 2 \pmod{4}$, $N_c(\mathbf{y}) - s_c \equiv 1 \pmod{4}$ for some a, c , and $N_b(\mathbf{y}) - s_b \equiv 0 \pmod{4}$ for all $b \neq a, c$	$x_i = x_{i+1} = a$ $0 < a < q-1$
$N_a(\mathbf{y}) - s_a \equiv 3 \pmod{4}$, $N_b(\mathbf{y}) - s_b \equiv 3 \pmod{4}$, $N_c(\mathbf{y}) - s_c \equiv 1 \pmod{4}$ for some a, b, c , and $N_d(\mathbf{y}) - s_d \equiv 0 \pmod{4}$ for all $d \neq a, b, c$	$\{x_i, x_{i+1}\} = \{a, b\}$ $0 < a, b < q-1$ $a \neq b$

the values of a, b and $c = a \oplus b$, but cannot necessarily deduce their order (ab or ba). In order to distinguish between the two cases $x_i x_{i+1} = ab$ or $x_i x_{i+1} = ba$, we need the following notation: for any $\mathbf{z} \in \Sigma_q^n$, let

$$\text{Inv}(\mathbf{z}) \triangleq |\{(i, j) : 1 \leq i < j \leq n, z_i > z_j\}|.$$

Let \mathbf{x}' and \mathbf{x}'' be the sequences obtained from \mathbf{y} by replacing a specific c with ab and ba , respectively. Then $\text{Inv}(\mathbf{x}') - \text{Inv}(\mathbf{x}'') = \pm 1$. Therefore, if we fix $\text{Inv}(\mathbf{x}) \pmod{2}$ and this value is known, we obtain that at most one of \mathbf{x}' and \mathbf{x}'' equals \mathbf{x} .

Now, consider the case when $0 \in \{\{x_i, x_{i+1}\}\}$ or $q-1 \in \{\{x_i, x_{i+1}\}\}$. In this case, \mathbf{y} is obtained from \mathbf{x} by a single deletion. In order to correct such an error, we need a q -ary code that can correct a single deletion.

For each $\mathbf{z} \in \Sigma_q^n$, let $\alpha(\mathbf{z}) \in \Sigma_2^{n-1}$, where $\alpha(\mathbf{z})_i = 1$ if $z_{i+1} \geq z_i$, and 0 otherwise for each $i \in [n-1]$. For given $t_1 \in \mathbb{Z}_n$ and $t'_1 \in \mathbb{Z}_q$, it was shown in [32] that the following q -ary code can correct a single deletion:

$$T_{t_1, t'_1}(n; q) \triangleq \left\{ \mathbf{z} \in \Sigma_q^n : \text{Syn}(\alpha(\mathbf{z})) \equiv t_1 \pmod{n}, \sum_{i=1}^n z_i \equiv t'_1 \pmod{q} \right\}.$$

However, the only role of the constraint $\sum_{i=1}^n z_i \equiv t'_1 \pmod{q}$ is to determine the deleted symbol. In our setting, the deleted symbol is known by calculating $N_a(\mathbf{y}) - s_a \pmod{4}$, so we do not need this constraint (in fact, in our case this constraint is replaced with the constraint $N_a(\mathbf{x}) \equiv s_a \pmod{4}$).

Putting what we have so far together, we construct the following code. For a given $\mathbf{s} = (s_a)_{a \in [0, q-2]} \in \mathbb{Z}_4^{q-1}$ and $\mathbf{t} = (t_1, t_2) \in \mathbb{Z}_n \times \mathbb{Z}_2$, let

$$\mathcal{C}_2(n; \mathbf{s}, \mathbf{t}) \triangleq \{\mathbf{x} \in \mathcal{C}_1(n; \mathbf{s}) : \text{Syn}(\alpha(\mathbf{z})) \equiv t_1 \pmod{n}, \text{Inv}(\mathbf{x}) \equiv t_2 \pmod{2}\}.$$

Let $\mathbf{x} \in \mathcal{C}_2(n; \mathbf{s}, \mathbf{t})$ and let \mathbf{y} be the sequence received after transmitting \mathbf{x} through a single-absorption channel. By the discussions above, if $0 \in \{\{x_i, x_{i+1}\}\}$ or $q-1 \in \{\{x_i, x_{i+1}\}\}$, we can recover \mathbf{x} from \mathbf{y} by the decoder of $T_{t_1, t'_1}(n; q)$. If $x_i x_{i+1} \in \{\{ab, ba\}\}$ where $0 < a, b < q-1$, we can find the values of a and b using \mathbf{s} and Table I, and for the specific $c = a \oplus b$ in \mathbf{y} that was obtained by the absorption, we can determine whether $x_i x_{i+1} = ab$ or $x_i x_{i+1} = ba$ using $\text{Inv}(\mathbf{x})$. What we are still missing in order to be able to recover \mathbf{x} is the exact absorption position, i.e., the position of that $c = a \oplus b$.

Our next aim is to add another layer of redundancy that determines the position of absorption in the case that $x_i x_{i+1} \in \{\{ab, ba\}\}$ with $0 < a, b < q-1$. We will divide our discuss into two cases. Different methods will be applied to locate the error position.

(1) **The Case** $a + b \leq q-1$

Let \mathbf{x}' be the sequence obtained from \mathbf{y} by replacing the c located at position i with one of ab and ba , \mathbf{x}'' be the sequence obtained from \mathbf{y} by replacing the c located at position j with one of ab and ba , where $1 \leq i < j \leq n-1$. Recall that $\text{Syn}(\mathbf{z}) = \sum_{i=1}^{|\mathbf{z}|} iz_i$ for any sequence \mathbf{z} .

Lemma IV.3 $\text{Syn}(\mathbf{x}') \not\equiv \text{Syn}(\mathbf{x}'') \pmod{qn}$.

Proof: Since $a + b \leq q-1$, we have $a \oplus b = a + b$. Then it is easy to see that

$$\begin{aligned} \text{Syn}(\mathbf{x}') - \text{Syn}(\mathbf{y}) &= \alpha + \sum_{k=i+1}^{n-1} y_k, \\ \text{Syn}(\mathbf{x}'') - \text{Syn}(\mathbf{y}) &= \beta + \sum_{k=j+1}^{n-1} y_k, \end{aligned}$$

where $\alpha, \beta \in \{\{a, b\}\}$. These two equations imply that

$$\text{Syn}(\mathbf{x}') - \text{Syn}(\mathbf{x}'') = \alpha - \beta + \sum_{k=i+1}^j y_k = \begin{cases} y_j + \sum_{k=i+1}^{j-1} y_k, & \text{if } \alpha = \beta, \\ 2\alpha + \sum_{k=i+1}^{j-1} y_k, & \text{if } \alpha \neq \beta. \end{cases}$$

Noticing that $y_j = a + b$ and $a, b > 0$, we have

$$0 < \text{Syn}(\mathbf{x}') - \text{Syn}(\mathbf{x}'') < qn.$$

Now the proof is completed. \square

By Lemma IV.3, if we fix $\text{Syn}(\mathbf{x}) \pmod{qn}$, where $\mathbf{x} \in \mathcal{C}_2(n; \mathbf{s}, \mathbf{t})$, then we can find a unique c in \mathbf{y} such that \mathbf{x} is obtained from \mathbf{y} by replacing this c with ab or ba . Details will be shown in the proof of Theorem IV.5 below.

(2) **The Case** $a + b \geq q$

In this case, we have $a \oplus b = q - 1$. We want to locate in \mathbf{y} the position of the symbol $q - 1$ which is obtained by a single absorption. To this end, we define the location sequence of a sequence $\mathbf{x} \in \Sigma_q^n$ to be $P(\mathbf{x}) \in \Sigma_2^n$, where

$$P(\mathbf{x})_i = \begin{cases} 0, & \text{if } x_i \neq q - 1, \\ 1, & \text{if } x_i = q - 1. \end{cases}$$

Suppose \mathbf{y} is obtained from \mathbf{x} by absorbing x_i and x_{i+1} , where $0 < x_i, x_{i+1} < q - 1$ and $x_i + x_{i+1} \geq q$. It is easy to see that $P(\mathbf{y})$ is obtained from $P(\mathbf{x})$ by replacing two adjacent 0s with a single 1. We call this error type $00 \rightarrow 1$. Now locating the error position in \mathbf{x} is reduced to locating the error position in $P(\mathbf{x})$. For this, we have the following code.

For any $n \geq 3$ and any $d \in \mathbb{Z}_{2n-3}$, define

$$\mathcal{C}_3(n; d) \triangleq \{\mathbf{z} \in \Sigma_2^n : \text{Syn}(\mathbf{z}) \equiv d \pmod{2n-3}\}.$$

Lemma IV.4 *The binary code $\mathcal{C}_3(n; d)$ can correct the error type $00 \rightarrow 1$ and locate the error position.*

Proof: Suppose that \mathbf{z}' is obtained from a codeword $\mathbf{z} \in \text{VT}_d(2n-3)$ by the error $00 \rightarrow 1$. Let the two sequences \mathbf{u} and \mathbf{v} be obtained from \mathbf{z}' by replacing $z'_i = 1$ and $z'_j = 1$ with 00, respectively, where $1 \leq i < j \leq n - 1$. Then $\text{Syn}(\mathbf{u}) - \text{Syn}(\mathbf{v}) = j - i + \sum_{k=i+1}^j z'_k$. So we have

$$0 < j - i \leq \text{Syn}(\mathbf{u}) - \text{Syn}(\mathbf{v}) \leq 2(j - i) \leq 2n - 4 < 2n - 3. \quad (3)$$

Now we can recover \mathbf{z} from \mathbf{z}' by the following procedure. Scan the symbols from the beginning of \mathbf{z}' to its end. If the symbol 1 is encountered, conduct the following steps.

Step 1 Replace this 1 with 00 and denote the resulting sequence by \mathbf{u} . If $\text{Syn}(\mathbf{u}) \equiv d \pmod{2n-3}$, let \mathbf{z} and output \mathbf{z} .

Otherwise, go to Step 2.

Step 2 Move to the next 1 and go to Step 1.

Since \mathbf{z}' is obtained from \mathbf{z} by the error type $00 \rightarrow 1$, this \mathbf{u} does exist. On the other hand, Equation (3) ensures that such \mathbf{u} is unique and the error position can be uniquely determined. \square

Now we are ready to give a code that can correct a single-absorption error. Given $n \geq 3$, $\mathbf{s} = (s_a)_{a \in [0, q-2]} \in \mathbb{Z}_4^{q-1}$, $\mathbf{t} = (t_1, t_2) \in \mathbb{Z}_n \times \mathbb{Z}_2$ and $\mathbf{d} = (d_1, d_2) \in \mathbb{Z}_{qn} \times \mathbb{Z}_{2n-3}$, let

$$\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{d}) \triangleq \{\mathbf{x} \in \mathcal{C}_2(n; \mathbf{s}, \mathbf{t}) : \text{Syn}(\mathbf{x}) \equiv d_1 \pmod{qn}, P(\mathbf{x}) \in \mathcal{C}_3(n; d_2)\}.$$

Theorem IV.5 *The code $\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{d})$ can correct a single-absorption error.*

Proof: Let $\mathbf{x} \in \mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{d})$ be the transmitted sequence and $\mathbf{y} \in \Sigma_q^{n-1}$ be the received sequence. Suppose that \mathbf{y} is obtained from \mathbf{x} by replacing $x_i x_{i+1}$ with $c = x_i \oplus x_{i+1}$. Since $\mathbf{x} \in \mathcal{C}_1(n; \mathbf{s})$, we know if $0 \in \{\{x_i, x_{i+1}\}\}$ or $q - 1 \in \{\{x_i, x_{i+1}\}\}$ or neither of the cases. If $0 \in \{\{x_i, x_{i+1}\}\}$ or $q - 1 \in \{\{x_i, x_{i+1}\}\}$, \mathbf{y} is obtained from \mathbf{x} by a single-deletion, which can be recovered since $\mathbf{x} \in \mathcal{C}_2(n; \mathbf{s}, \mathbf{t})$.

If $0, q - 1 \notin \{\{x_i, x_{i+1}\}\}$, we can determine the multiset $\{\{x_i, x_{i+1}\}\}$ and thus know whether $x_i + x_{i+1} < q$ or not. We have two cases:

1) If $x_i + x_{i+1} < q$, the following algorithm can be used to recover \mathbf{x} . Scan the symbols from the beginning of \mathbf{y} to its end. If the symbol c is encountered, conduct the following steps.

Step 1 If $x_i x_{i+1} = aa$ for some a , replace this c with aa . Denote the resulting sequence by \mathbf{x}' and go to Step 4.

If $x_i \neq x_{i+1}$, we must have $x_i x_{i+1} \in \{ab, ba\}$ for some $a \neq b$. Go to Step 2.

Step 2 Replace this c with ab and denote the resulting sequence by \mathbf{x}' . If $\text{Inv}(\mathbf{x}') \equiv t_2 \pmod{2}$, go to Step 4. Otherwise, keep this c unchanged and go to Step 3.

Step 3 Replace this c with ba . Denote the resulting sequence by \mathbf{x}' and go to Step 4. Otherwise, keep this c unchanged and go to Step 5.

Step 4 If $\text{Syn}(\mathbf{x}') \equiv d_1 \pmod{qn}$, let $\mathbf{x} = \mathbf{x}'$ and output \mathbf{x} . Otherwise, keep this c unchanged and go to Step 5.

Step 5 Move to the next c and go to Step 1.

Since \mathbf{y} is obtained from \mathbf{x} by replacing an $x_i x_{i+1}$ with c , this \mathbf{x}' does exist. On the other hand, Lemma IV.3 ensures that such \mathbf{x}' is unique.

- 2) If $x_i + x_{i+1} \geq q$, we can recover \mathbf{x} by the following procedure. First, since $P(\mathbf{x}) \in \mathcal{C}_3(n; d_1)$, we can determine the error position i from $P(\mathbf{y})$ by the algorithm given in the proof of Lemma IV.4. If $x_i x_{i+1} = aa$ for some a , then replace $y_i (= q - 1)$ with aa and output the resulting sequence. If $x_i x_{i+1} \in \{ab, ba\}$ for some $a \neq b$, we can know whether $x_i x_{i+1} = ab$ or $x_i x_{i+1} = ba$ by $\text{Inv}(\mathbf{x}) \pmod{2}$. Once $x_i x_{i+1}$ is determined, replace y_i with $x_i x_{i+1}$ and output the resulting sequence. Lemma IV.4 ensures that the sequence \mathbf{x} can be uniquely recovered. \square

By the pigeonhole principle, there are some \mathbf{s} , \mathbf{t} and \mathbf{d} such that

$$|\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{u})| \geq \frac{q^n}{4^{q-1} \cdot n \cdot 2 \cdot (qn) \cdot (2n-3)}. \quad (4)$$

This lower bound means that the redundancy of $\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{u})$ is at most $3 \log_q(n) + O(1)$ for some choice of \mathbf{s} , \mathbf{t} and \mathbf{d} . If measured in binary bits, this redundancy is at most $3 \log_2(n) + O(1)$. Recall that in [19, Corollary 17], the authors gave a q -ary single-deletion single-substitution correcting code of redundancy at most $10 \log_2(n) + O(1)$. So the code $\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{d})$ performs better than the existing one in [19]. One may ask if $3 \log_q(n) + O(1)$ is the best redundancy that can be achieved. Based on $\mathcal{C}(n; \mathbf{s}, \mathbf{t}, \mathbf{d})$ and new ideas, we will show in next subsection that the redundancy can be further reduced to at most $\log_q(n) + O(\log_q \log_q(n))$.

B. An improved construction

In this subsection, we use Theorem IV.5 together with ideas from [1] and provide a code with redundancy $\log_q(n) + O(\log_q \log_q(n))$. We first outline the basic idea.

We begin with constructing a code with redundancy $\log_q(n) + O(1)$. This code has the property that when receiving a sequence which is a corrupted version of a codeword \mathbf{x} , it is possible to locate a window of length $L = \Theta(\log_q^2(n))$ that contains the erroneous position. That is to say, we only need to correct the absorption error within a shorter substring of \mathbf{x} . To this end, we should partition \mathbf{x} into consecutive disjoint intervals of length $2L + 1$ and then apply Theorem IV.5 to each of these intervals. As we will show next, this will only increase the redundancy by $O(\log_q \log_q(n))$ and so the overall redundancy of the resulted code is $\log_q(n) + O(\log_q \log_q(n))$. The details will be clear from the subsequent analysis.

For each $\mathbf{x} \in \Sigma_q^n$, which ends with 0011, we can segment \mathbf{x} and get a string $\mathbf{z}^{\mathbf{x}} = \mathbf{z}_1^{\mathbf{x}} \cdots \mathbf{z}_{l_{\mathbf{x}}}^{\mathbf{x}}$, where $1 \leq l_{\mathbf{x}} \leq n/4$, and each substring $\mathbf{z}_i^{\mathbf{x}}$ ends with 0011, and 0011 appears exactly once in $\mathbf{z}_i^{\mathbf{x}}$. For example, let $q = 3$ and $\mathbf{x} = 00111230320011$. Then $l_{\mathbf{x}} = 2$ and $\mathbf{z}_1^{\mathbf{x}} = 0011$, $\mathbf{z}_2^{\mathbf{x}} = 1230320011$.

Let $\delta = c_1 + c_2 \lceil \log_q(n) \rceil$, where constants c_1 and c_2 are both multiples of 4 and satisfy

$$\left(\frac{q^4}{q^4 - 1} \right)^{\frac{c_1}{4} - 1} \geq \frac{q}{q - 1}, \text{ and } \left(\frac{q^4}{q^4 - 1} \right)^{\frac{c_2}{4}} \geq q.$$

Since $\frac{q^4}{q^4 - 1} > 1$, the desired constants c_1 and c_2 do exist. For example, if $q = 3$, the smallest c_1 is 136, while the smallest c_2 is 356.

Lemma IV.6 Suppose that X is chosen uniformly at random from Σ_q^n . Then

$$\Pr(|\mathbf{z}_i^X| \leq \delta, i = 1, \dots, l_x) \geq \frac{1}{q}.$$

Proof: The probability that a fixed length-4 substring of X equals 0011 is $\frac{1}{q^4}$. Then for any i , the probability that $|\mathbf{z}_i^X| > \delta$ is at most

$$\left(\frac{q^4 - 1}{q^4} \right)^{\frac{\delta - 4}{4}} \leq \frac{q - 1}{qn},$$

where the inequality follows from the choices of c_1 and c_2 . Now the conclusion follows from the union bound. \square

Let $\mathcal{R}_{q,n}$ be the set of all strings $\mathbf{x} \in \Sigma_q^n$ which ends with 0011 and satisfies the condition that $|\mathbf{z}_i^X| \leq \delta$ for all $i = 1, \dots, l_x$. Then Lemma IV.6 implies $|\mathcal{R}_{q,n}| \geq q^{n-5}$. Next, we briefly explain how to construct $\mathcal{R}_{q,n}$. Let

$$\mathcal{Z} = \{ \mathbf{z} \in \Sigma_q^{\leq \delta} : \mathbf{z} \text{ ends with } 0011 \text{ and } 0011 \text{ does not appear elsewhere in } \mathbf{z} \}.$$

Since $\delta = c_1 + c_2 \lceil \log_q(n) \rceil$, the size of $\Sigma_q^{\leq \delta}$ is bounded above by $O(n^{c_2})^6$. This implies that \mathcal{Z} can be constructed by brute force searching. We can construct $\mathcal{R}_{q,n}$ by concatenating sequences in \mathcal{Z} . This process can be somewhat involved, but this

⁶More accurately, the capacity of the set of strings of length n that do not contain 0011, which can be calculated using constrained systems techniques, is $\log_q(1.839)$ which is roughly 0.87 in the binary case.

is a one-time pre-processing task. When $(c_1 - 4) \log_q(e)/(4q^4) \geq 5$ and $c_2 \log_q(e)/(4q^4) \geq 1$, we present an algorithm for encoding (and decoding) an arbitrary sequence of length n into a sequence in $\mathcal{R}_{q,n+5}$ (see Appendix A).

Observation IV.7 Let $\mathbf{x} \in \Sigma_q^n$ be a string, ending with 0011. If the last 0011 is destroyed due to an absorption error, it is easy to detect and correct that error. If the absorption error does not change the last 0011 and the received sequence is \mathbf{y} , we have $|\mathbf{y}| = |\mathbf{x}| - 1$ and $l_{\mathbf{y}} - l_{\mathbf{x}} \in \{0, -1, 1\}$ where $l_{\mathbf{x}}, l_{\mathbf{y}}$ denote the number of substrings that end with 0011 in \mathbf{x}, \mathbf{y} , respectively.

For any $n \in \mathbb{N}$ and any $\mathbf{x} \in \Sigma_q^n$, define

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^{l_{\mathbf{x}}} j |\mathbf{z}_j^{\mathbf{x}}| \pmod{2n}, \\ g(\mathbf{x}) &= l_{\mathbf{x}} \pmod{3}. \end{aligned}$$

If \mathbf{y} is obtained from \mathbf{x} by an absorption, the function $g(\mathbf{x})$ can help us to determine the exact value of $l_{\mathbf{y}} - l_{\mathbf{x}}$. For a given $\mathbf{r} = (r_1, r_2) \in \mathbb{Z}_{2n} \times \mathbb{Z}_3$, we define the code $\mathcal{D}_1(n; \mathbf{r}) \subseteq \Sigma_q^n$ as

$$\mathcal{D}_1(n; \mathbf{r}) = \{\mathbf{x} \in \mathcal{R}_{q,n} : f(\mathbf{x}) = r_1, g(\mathbf{x}) = r_2\}.$$

With suitable parameters, this code has redundancy at most $\log_q(n) + O(1)$.

Theorem IV.8 Let $\mathbf{x} \in \mathcal{D}_1(n; \mathbf{r})$ be a sequence and let \mathbf{y} be the sequence obtained from \mathbf{x} after a single absorption. Then there is a constant c_3 , which is a function of c_1 and c_2 , such that a window $W \subseteq [1, n-1]$ of size $c_3 \log_q^2(n)$ that contains the position where the absorption error has occurred in \mathbf{y} , can be detected. Furthermore, the window can be found in $O(n)$ time.

The proof of Theorem IV.8 is similar to the proof of [1, Theorem 4] and is deferred to Appendix B.

Let $L = c_3 \log_q^2(n)$. For simplicity, we assume $(2L+1) \mid n$ and let $t = n/(2L+1)$. All the following arguments can be generalized to the case $(2L+1) \nmid n$ in a straightforward way (see Remark IV.10 below). We partition $\{1, \dots, n\}$ into consecutive disjoint intervals $I_1^{(1)}, \dots, I_t^{(1)}$ of length $2L+1$. In other words,

$$I_i^{(1)} = [1 + (i-1)(2L+1), i(2L+1)] \quad (5)$$

for all $1 \leq i \leq t$. Furthermore, we define a family of shifted intervals $I_1^{(2)}, \dots, I_{t-1}^{(2)}$, where $I_i^{(2)} = I_i^{(1)} + L$.⁷ For given $\mathbf{x} \in \Sigma_q^n$, let $\mathbf{x}^{(1,i)} = \mathbf{x}_{I_i^{(1)}}$ and $\mathbf{x}^{(2,i)} = \mathbf{x}_{I_i^{(2)}}$. In other words, $\mathbf{x}^{(1,i)}$ is the substring corresponding to $I_i^{(1)}$ and $\mathbf{x}^{(2,i)}$ is the substring corresponding to $I_i^{(2)}$.

For a given sequence $\mathbf{z} \in \Sigma_q^{2L+1}$, we define

$$\hat{f}(\mathbf{z}) = (N_a(\mathbf{z}))_{a \in [0, q-2]} \times (\text{Syn}(\alpha(\mathbf{z})), \text{Inv}(\mathbf{z}), \text{Syn}(\mathbf{z}), \text{Syn}(P(\mathbf{z}))).$$

The values of $\hat{f}(\mathbf{z})$ are taken from $\mathbb{Z}_4^{q-1} \times \mathbb{Z}_{2L+1} \times \mathbb{Z}_2 \times \mathbb{Z}_{q(2L+1)} \times \mathbb{Z}_{4L-1}$. With the function $\hat{f}(\cdot)$ in hand, we define the functions:

$$\begin{aligned} \hat{g}_1(\mathbf{x}) &= \sum_{i=1}^t \hat{f}(\mathbf{x}^{(1,i)}), \\ \hat{g}_2(\mathbf{x}) &= \sum_{i=1}^{t-1} \hat{f}(\mathbf{x}^{(2,i)}), \end{aligned}$$

where the sums are performed position-wise over $\mathbb{Z}_4^{q-1} \times \mathbb{Z}_{2L+1} \times \mathbb{Z}_2 \times \mathbb{Z}_{q(2L+1)} \times \mathbb{Z}_{4L-1}$. Now we can give the desired code. For given $\alpha, \beta \in \mathbb{Z}_4^{q-1} \times \mathbb{Z}_{2L+1} \times \mathbb{Z}_2 \times \mathbb{Z}_{q(2L+1)} \times \mathbb{Z}_{4L-1}$ and $\mathbf{r} = (r_1, r_2) \in \mathbb{Z}_{2n} \times \mathbb{Z}_3$, let

$$\mathcal{D}(n; \mathbf{r}, \alpha, \beta) = \mathcal{D}_1(n; \mathbf{r}) \cap \{\mathbf{x} \in \mathcal{R}_{q,n} : \hat{g}_1(\mathbf{x}) = \alpha, \hat{g}_2(\mathbf{x}) = \beta\}.$$

Similar to Equation (4), there exists a choice of \mathbf{r}, α and β , such that

$$|\mathcal{D}(n; \mathbf{r}, \alpha, \beta)| \geq \frac{|\mathcal{R}_{q,n}|}{2n \cdot 3 \cdot [4^{q-1} \cdot (2L+1) \cdot 2 \cdot (q(2L+1)) \cdot (4L-1)]^2}.$$

Therefore, the redundancy of $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ is at most $\log_q(n) + 12 \log_q \log_q(n) + O(1)$ (recall that we require that c_1, c_2 and c_3 are constants and n is large compared to these constants).

Theorem IV.9 The code $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ can correct a single absorption error.

Proof: Let \mathbf{x} be the transmitted codeword and \mathbf{y} be the received sequence. The proof of Theorem IV.8 gives a method to locate the error position within a window $W = [i_1, i_1 + L - 1] \subseteq [n-1]$. By the constructions of $I_i^{(1)}$'s and $I_i^{(2)}$'s, there exists

⁷For a set A of integers and an integer m , we define $A + m = \{a + m : a \in A\}$.

some i such that W is contained in $I_i^{(1)}$ or $I_i^{(2)}$. The value of i can be determined in the following way (recall Equation (5) for the definitions of $I_i^{(1)}$'s and $I_i^{(2)}$'s).

Step 1 Find the largest $k \geq 0$ such that $k(2L+1) < i_1$ and $W \subseteq [k(2L+1)+1, k(2L+1)+2L]$. Then $i = k+1$. If such a k does not exist, go to Step 2.

Step 2 Find the largest $k \geq 0$ such that $k(2L+1)+L < i_1$ and $W \subseteq [k(2L+1)+L+1, k(2L+1)+3L]$. Then $i = k+1$. Since any window of length L must be contained in some $I_i^{(1)}$ or $I_i^{(2)}$, the above two steps can successfully find such an i . Now we can recover \mathbf{x} from \mathbf{y} by the following procedure.

Case (1) The value of i is found in Step 1. In this case, we have $x_j = y_j$ for all $j \leq (i-1)(2L+1)$ and $x_j = y_{j-1}$ for all $j > i(2L+1)$. In other words, we can recover $\mathbf{x}^{(1,j)}$ for all $j \neq i$ directly. Therefore, we can compute $\hat{f}(\mathbf{x}^{(1,j)})$ for all $j \neq i$. Then comparing α and $\sum_{j \neq i} \hat{f}(\mathbf{x}^{(1,j)})$, we can know $\hat{f}(\mathbf{x}^{(1,i)})$. Let $\mathbf{y}^{(i)} = \mathbf{y}_{[(i-1)(2L+1)+1, i(2L+1)-1]}$. Then $\mathbf{y}^{(i)}$ is the corrupted version of $\mathbf{x}^{(1,i)}$. Theorem IV.5 ensures that we can recover $\mathbf{x}^{(1,i)}$ from $\mathbf{y}^{(i)}$ with the help of $\hat{f}(\mathbf{x}^{(1,i)})$. Now the transmitted sequence \mathbf{x} is recovered.

Case (2) The value of i is found in Step 2. In this case, we have $x_j = y_j$ for all $j \leq (i-1)(2L+1)+L$ and $x_j = y_{j-1}$ for all $j > i(2L+1)+L$. In other words, we can recover x_j for all $j \notin [(i-1)(2L+1)+L+1, i(2L+1)+L]$ directly. Therefore, we can compute $\hat{f}(\mathbf{x}^{(2,j)})$ for all $j \neq i$. Then comparing β and $\sum_{j \neq i} \hat{f}(\mathbf{x}^{(2,j)})$, we can know $\hat{f}(\mathbf{x}^{(2,i)})$.

Let $\mathbf{y}^{(i)} = \mathbf{y}_{[(i-1)(2L+1)+L+1, i(2L+1)+L-1]}$. Then $\mathbf{y}^{(i)}$ is the corrupted version of $\mathbf{x}^{(2,i)}$. Theorem IV.5 ensures that we can recover $\mathbf{x}^{(2,i)}$ from $\mathbf{y}^{(i)}$ with the help of $\hat{f}(\mathbf{x}^{(2,i)})$. Now the transmitted sequence \mathbf{x} is recovered. \square

Remark IV.10 If $2L+1 \nmid n$, let $t = \lfloor n/(2L+1) \rfloor$ and $L' = n - t(2L+1)$. Then $0 < L' \leq 2L$. The $2t-1$ intervals $I_i^{(1)}$ ($1 \leq i \leq t$) and $I_i^{(2)}$ ($1 \leq i \leq t-1$) are defined as above. There are two cases.

- When $L' \leq L$, let $I_t^{(2)} = [(t-1)(2L+1)+L+1, n]$. Then $L+2 \leq |I_t^{(2)}| \leq 2L+1$. So we define $\hat{g}_1(\mathbf{x})$ as above and $\hat{g}_2(\mathbf{x}) = \sum_{i=1}^t \hat{f}(\mathbf{x}^{(2,i)})$.
- When $L < L' \leq 2L$, let $I_{t+1}^{(1)} = [t(2L+1)+1, n]$ and $I_t^{(2)} = [(t-1)(2L+1)+L+1, t(2L+1)+L]$. Then $L < |I_{t+1}^{(1)}| \leq 2L$ and $|I_t^{(2)}| = 2L+1$. So we define $\hat{g}_1(\mathbf{x}) = \sum_{i=1}^{t+1} \hat{f}(\mathbf{x}^{(1,i)})$ and $\hat{g}_2(\mathbf{x}) = \sum_{i=1}^t \hat{f}(\mathbf{x}^{(2,i)})$.

C. Codes correcting multiple errors

In this subsection, we study codes that can correct multiple absorption errors. Recall that the alphabet size q is at least 3, unless otherwise stated. We first claim that t -absorption is a special case of t -deletion- t -substitution and give two known results. After that, we explain the difference between t -absorption and t -deletion- t -substitution, which justifies our searching for better codes for our setting. Our construction is based on the single-absorption correcting code given in Theorem IV.9 and the syndrome compression technique with precoding developed recently [20].

In Observation IV.1, we have shown that a single-absorption error corresponds to a single-deletion together with at most a single-substitution. By Definition II.2, it is not difficult to see that this conclusion holds for multiple absorptions as well. In other words, a t -absorption error is the combination of t deletions and *at most* t substitutions. To see that, it suffices to notice

that the absorption error $\left(\bigoplus_{j=i_l}^{i_l+s_l} x_j \right)$ can be interpreted as firstly deleting s_l symbols x_j ($i_l \leq j < i_l + s_l$) and then substituting $x_{i_l+s_l}$ by $\left(\bigoplus_{j=i_l}^{i_l+s_l} x_j \right)$. If $\left(\bigoplus_{j=i_l}^{i_l+s_l} x_j \right) \neq x_{i_l+s_l}$, the second step is a substitution error. In other words, the absorption error $\left(\bigoplus_{j=i_l}^{i_l+s_l} x_j \right)$ of s_l+1 consecutive symbols can be interpreted as s_l deletions and *at most* one substitution.

Therefore, a t -deletion- t -substitution correcting code is naturally a t -absorption correcting code. We first introduce two classes of t -deletion- t -substitution correcting codes given in the literature. They will be used as a building block in our construction of t -absorption correcting codes.

By carefully checking the proof of [20, Lemma 9], we draw the following conclusion.

Lemma IV.11 Suppose that $q \geq 3$ and t are fixed positive integers. There exists a q -ary systematic⁸ t -deletion t -substitution correcting code $\mathcal{E}_q \subseteq \Sigma_q^N$ whose redundancy is at most $\frac{22t}{\log_q(2)} \log_q(N) + o(\log_q(N))$. The encoding and decoding complexities⁹ are $O(N^{6t+1})$ and $O(N^{3t+1})$, respectively.

When q is a prime power¹⁰, the authors of [20] obtained a better result.

⁸In the proof of [20, Lemma 9], an systematic encoder was defined.

⁹These two complexities follow from the construction of \mathcal{E}_q and [20, Theorem 1].

¹⁰When constructing the code in [20, Theorem 3], the authors used a BCH code over the finite field \mathbb{F}_q . This is the reason why we require that q is a prime power.

Lemma IV.12 [20, Theorem 3] *Let $q \geq 3$ be a prime power. There exists a q -ary systematic t -deletion t -substitution correcting code $\mathcal{E}_q \subseteq \Sigma_q^N$ with redundancy at most $\left(8t - 1 - \left\lfloor \frac{2t-1}{q} \right\rfloor\right) \log_q(N) + o(\log_q(N))$. The encoding and decoding complexities are $O(N^{4t+1})$ and $O(N^{2t+1})$, respectively.*

Furthermore, the codes \mathcal{E}_q in [20, Lemma 9] and [20, Theorem 3] can be expressed as

$$\mathcal{E}_q = \{(\mathbf{u}, \text{Red}_{q,n}(\mathbf{u})) : \mathbf{u} \in \Sigma_q^n\} \quad (6)$$

where \mathbf{u} is the information sequence and $\text{Red}_{q,n}(\mathbf{u})$ is the sequence of redundancy symbols. Note that $N = n + |\text{Red}_{q,n}(\mathbf{u})|$. Let

$$R_{q,n} = \begin{cases} \left(8t - 1 - \left\lfloor \frac{2t-1}{q} \right\rfloor\right) \log_q(N) + o(\log_q(N)) & \text{if } q \text{ is a prime power,} \\ \frac{22t}{\log_q(2)} \log_q(N) + o(\log_q(N)), & \text{if } q \text{ is arbitrary.} \end{cases}$$

Since $\frac{n}{N} \geq \frac{1}{2}$ when n is sufficiently large, we have

$$R_{q,n} = \begin{cases} \left(8t - 1 - \left\lfloor \frac{2t-1}{q} \right\rfloor\right) \log_q(n) + o(\log_q(n)) & \text{if } q \text{ is a prime power,} \\ \frac{22t}{\log_q(2)} \log_q(n) + o(\log_q(n)), & \text{otherwise.} \end{cases} \quad (7)$$

In the following, whenever $R_{q,n}$ is mentioned, we always refer to Equation (7).

From Lemma IV.11 and Lemma IV.12 we can see that $\text{Red}_{q,n}(\mathbf{u}) \in \Sigma_q^{R_{q,n}}$. For our purpose, we can also view $\text{Red}_{q,n}$ as a function $\text{Red}_{q,n} : \Sigma_q^n \rightarrow [0, q^{R_{q,n}} - 1]$. Let $\mathcal{B}_t^{DS}(\mathbf{u})$ be the t -deletion- t -substitution ball centered at \mathbf{u} , i.e.,

$$\mathcal{B}_t^{DS}(\mathbf{u}) = \left\{ \mathbf{z} \in \Sigma_q^{n-t} : \begin{array}{l} \mathbf{z} \text{ is obtained from } \mathbf{u} \text{ by } t \text{ deletions} \\ \text{and at most } t \text{ substitutions} \end{array} \right\}. \quad (8)$$

Then Lemma IV.11, Lemma IV.12 and Equation (6) imply the following corollary.

Corollary IV.13 *If $\mathcal{B}_t^{DS}(\mathbf{u}) \cap \mathcal{B}_t^{DS}(\mathbf{u}') \neq \emptyset$ and $\mathbf{u} \neq \mathbf{u}'$, then $\text{Red}_{q,n}(\mathbf{u}) \neq \text{Red}_{q,n}(\mathbf{u}')$.*

As discussed above, Lemma IV.11 and Lemma IV.12 provide us with two class of t -absorption correcting codes with low redundancy. However, the two kinds of error models differ in the following two aspects.

- In the t -deletion- t -substitution setup, the error positions are assumed to be arbitrary. But for the t -absorption channel, the deletion-positions and the substitution-positions are ‘‘close’’. For example, the absorption error $\left(\bigoplus_{j=i_l}^{i_l+s_l} x_j\right)$ leads to deletions in positions j ($i_l \leq j < i_l + s_l$) and a (possible) substitution in position $i_l + s_l$. Therefore, the deletions and substitution are constrained to within a window of length $s_l + 1$.
- In the t -deletion- t -substitution setup, a symbol $a \in \Sigma_q$ can be substituted by an arbitrary symbol $b \in \Sigma_q \setminus \{a\}$. However, for absorption channels, a symbol $a \in \Sigma_q$ can only be substituted by some $b > a$ and $b \in \Sigma_q$.

Therefore, it is reasonable to deem that there are better codes for absorption channels, which is the main goal of this subsection. In the rest of this subsection, we will apply the syndrome compression technique with precoding to show that for our setting, there are codes with even lower redundancy. The syndrome compression technique was first established in [22], [34] for designing t -deletion correcting codes, and then was further developed in [35] to a general method for obtaining low-redundancy error correcting codes. More recently, [20] further improved the syndrome compression technique by applying a precoding process.

To describe the syndrome compression technique, we need to introduce some notations. Let $\mathcal{B}(\mathbf{u})$ be a general error ball centered at the sequence $\mathbf{u} \in \Sigma_q^n$. The definition of such error balls is determined by the specific problem under consideration. For example, if we are studying t -deletion- t -substitution error correcting codes, then the error ball $\mathcal{B}(\mathbf{u})$ is defined as Equation (8). Consider some fixed error and its corresponding error ball $\mathcal{B}(\mathbf{u})$. For a given code $\mathcal{E} \subseteq \Sigma_q^n$ and $\mathbf{u} \in \mathcal{E}$, we define

$$\mathcal{N}_{\mathcal{E}}(\mathbf{u}) = \{\mathbf{u}' \in \mathcal{E} : \mathbf{u}' \neq \mathbf{u} \text{ and } \mathcal{B}(\mathbf{u}') \cap \mathcal{B}(\mathbf{u}) \neq \emptyset\}.$$

The following lemma, which is a variant of [34, Lemma 1] and [20, Lemma 3], is key to our purpose. We include its proof here because the proof reveals how the syndrome compression technique works.

Lemma IV.14 *Let $\mathcal{E} \subseteq \Sigma_q^n$ be a code and $N > \max\{|\mathcal{N}_{\mathcal{E}}(\mathbf{u})| : \mathbf{u} \in \mathcal{E}\}$. Suppose that the function $f : \Sigma_q^n \rightarrow [0, q^{R(n)} - 1]$ (where $R(n)$ is a function of n and $R(n) \geq 2$) satisfies the following property:*

(P1) *if $\mathbf{u} \in \Sigma_q^n$ and $\mathbf{u}' \in \mathcal{N}_{\Sigma_q^n}(\mathbf{u})$, then $f(\mathbf{u}) \neq f(\mathbf{u}')$.*

Then there exists a function $\bar{f} : \mathcal{E} \rightarrow \left[0, q^{2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)} - 1\right]$ such that $\bar{f}(\mathbf{u}) \neq \bar{f}(\mathbf{u}')$ for any $\mathbf{u} \in \mathcal{E}$ and $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$.

Proof: For any $\mathbf{u} \in \mathcal{E}$ and $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$, we have $1 \leq |f(\mathbf{u}) - f(\mathbf{u}')| < q^{R(n)}$ due to (P1). For any $\mathbf{u} \in \mathcal{E}$, let

$$D(\mathbf{u}) = \{p : p \text{ is a positive divisor of } |f(\mathbf{u}) - f(\mathbf{u}')| \text{ for some } \mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})\}.$$

By [35, Lemma 3], the number of positive divisors of $|f(\mathbf{u}) - f(\mathbf{u}')|$ is upper bounded by

$$q^{O\left(\frac{R(n)}{\log_q(R(n))}\right)},$$

for each $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$. So we have

$$|D(\mathbf{u})| \leq |\mathcal{N}_{\mathcal{E}}(\mathbf{u})| q^{O\left(\frac{R(n)}{\log_q(R(n))}\right)} < N q^{O\left(\frac{R(n)}{\log_q(R(n))}\right)} = q^{\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)}.$$

This implies that there is an integer $P(\mathbf{u}) \in \left[1, q^{\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)}\right]$ such that $f(\mathbf{u}) \not\equiv f(\mathbf{u}') \pmod{P(\mathbf{u})}$ for all $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$. Now for each $\mathbf{u} \in \mathcal{E}$, we define

$$\bar{f}(\mathbf{u}) = (\text{Expan}_q(f(\mathbf{u}) \pmod{P(\mathbf{u})}), \text{Expan}_q(P(\mathbf{u}))),$$

where $\text{Expan}_q(m)$ is the q -ary expansion of the integer m . Clearly, $\bar{f}(\mathbf{u})$ is a q -ary vector of length $2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)$ and thus we can view \bar{f} as a function $\bar{f} : \mathcal{E} \rightarrow \left[0, q^{2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)} - 1\right]$. By construction, it holds that $\bar{f}(\mathbf{u}) \neq \bar{f}(\mathbf{u}')$ for any $\mathbf{u} \in \mathcal{E}$ and $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$. \square

Remark IV.15 *In most cases, the number N is a polynomial in n . So if it holds that $O\left(\frac{R(n)}{\log_q(R(n))}\right) = O(\log_q(n))$, the function \bar{f} can be computed in polynomial time.*

Before moving on, we explain how Lemma IV.14 helps to compress the code redundancy. We follow the notations in Lemma IV.14. For a given $a_1 \in [0, q^{R(n)} - 1]$, the function f can be used to define a code

$$\mathcal{E}'(a_1) = \{\mathbf{u} \in \Sigma_q^n : f(\mathbf{u}) = a_1\},$$

where there exists some a_1 such that the redundancy of $\mathcal{E}'(a_1)$ is at most $R(n)$. If the conditions in Lemma IV.14 are satisfied, then the function \bar{f} can be used to define another code

$$\mathcal{E}''(a_2) = \{\mathbf{u} \in \mathcal{E} : \bar{f}(\mathbf{u}) = a_2\}$$

where $a_2 \in \left[0, q^{2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)} - 1\right]$, and there exists some a_2 such that the redundancy of $\mathcal{E}''(a_2)$ is at most $r(\mathcal{E}) + 2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)$, where $r(\mathcal{E})$ is the redundancy of \mathcal{E} . If $r(\mathcal{E}) + 2\log_q(N) + O\left(\frac{R(n)}{\log_q(R(n))}\right)$ is much smaller than $R(n)$, then the code redundancy is successfully compressed. If $\mathcal{E} = \Sigma_q^n$, we obtain the original syndrome compression technique in [35]. If \mathcal{E} is chosen to be a proper subset of Σ_q^n , then we obtain the syndrome compression technique with precoding in [20].

Now we are ready to derive the main result of this subsection, that is, t -absorption correcting codes ($t \geq 2$). In this case, the error ball $\mathcal{B}(\mathbf{u})$ is defined to be the t -absorption ball (see Equation (1)), i.e.,

$$\mathcal{B}(\mathbf{u}) = \mathcal{B}_t^{ab}(\mathbf{u}) = \{\mathbf{z} \in \Sigma_q^{n-t} : \mathbf{z} \text{ is obtained from } \mathbf{u} \text{ by } t \text{ absorption errors}\}.$$

We choose \mathcal{E} to be the code $\mathcal{D}(n; r, \alpha, \beta)$ in Theorem IV.9, and f to be the function $\text{Red}_{q,n}$ (see Equation (6)). From Corollary IV.13, f satisfies the property (P1) in Lemma IV.14 with $R(n) = R_{q,n}$, which is defined as in Equation (7). So we have $O\left(\frac{R(n)}{\log_q(R(n))}\right) = o(\log_q(n))$.

Firstly, we need to estimate an upper bound of $|\mathcal{N}_{\mathcal{E}}(\mathbf{u})|$ for any $\mathbf{u} \in \mathcal{E}$. For a given sequence \mathbf{z} , if we insert a symbol at the end of \mathbf{z} , or replace some z_i with ab such that $z_i = a \oplus b$, then we say we perform a *splitting* operation on \mathbf{z} .

Claim IV.16 *Let $t \geq 2$. Then for any $\mathbf{u} \in \mathcal{E}$, we have $|\mathcal{N}_{\mathcal{E}}(\mathbf{u})| < q^{2t-2}n^{2t-1}$.*

Proof: We should estimate the number of $\mathbf{u}' \in \mathcal{E}$ such that $\mathbf{u}' \neq \mathbf{u}$ and $\mathcal{B}_t(\mathbf{u}') \cap \mathcal{B}_t(\mathbf{u}) \neq \emptyset$. Each such \mathbf{u}' can be obtained through the following steps.

- Step 1** Obtain a sequence $\mathbf{u}^{(1)}$ from \mathbf{u} by sequentially performing t absorptions, which has at most $n(n-1)\cdots(n-t+1) < n^t$ possibilities.
- Step 2** For each $\mathbf{u}^{(1)}$, we perform a splitting operation on $\mathbf{u}^{(1)}$ to get a sequence $\mathbf{z}^{(1)}$. Then we perform a splitting operation on $\mathbf{z}^{(1)}$ to get a sequence $\mathbf{z}^{(2)}$. Repeat this process. after $t-1$ steps, we will get a sequence $\mathbf{z}^{(t-1)}$. For each $\mathbf{u}^{(1)}$, there are at most $q^{2t-2}(n-t+1)(n-t+2)\cdots(n-1) < q^{2t-2}n^{t-1}$ such $\mathbf{z}^{(t-1)}$'s.
- Step 3** For each $\mathbf{z}^{(t-1)}$, we perform a splitting operation on $\mathbf{z}^{(t-1)}$ to get a sequence $\mathbf{u}' \in \mathcal{E}$. Since \mathcal{E} is a single-absorption correcting code, there is at most one \mathbf{u}' for each $\mathbf{z}^{(t-1)}$.

Overall, the number of \mathbf{u}' is strictly less than $q^{2t-2}n^{2t-1}$ and thus $|\mathcal{N}_{\mathcal{E}}(\mathbf{u})| < q^{2t-2}n^{2t-1}$. \square

Now we choose $N = q^{2t-2}n^{2t-1}$. Then by Lemma IV.14, we have a function $\bar{f} : \mathcal{E} \rightarrow [0, q^{(4t-2)\log_q(n) + o(\log_q(n))} - 1]$ such that $\bar{f}(\mathbf{u}) \neq \bar{f}(\mathbf{u}')$ for any $\mathbf{u} \in \mathcal{E}$ and $\mathbf{u}' \in \mathcal{N}_{\mathcal{E}}(\mathbf{u})$. Combining the above discussions, we obtain the main result of this subsection.

Theorem IV.17 Let $q \geq 3$ and $t \geq 2$ be fixed integers. For given $\alpha, \beta \in \mathbb{Z}_4^{q-1} \times \mathbb{Z}_{2L} \times \mathbb{Z}_2 \times \mathbb{Z}_{q(2L+1)} \times \mathbb{Z}_{4L-1}$, $\mathbf{r} = (r_1, r_2) \in \mathbb{Z}_{2n} \times \mathbb{Z}_3$ and $0 \leq a < q^{(4t-2)\log_q(n) + o(\log_q(n))}$, let

$$\mathcal{E}(n; \mathbf{r}, \alpha, \beta, a) = \{ \mathbf{c} \in \mathcal{D}(n; \mathbf{r}, \alpha, \beta) : \bar{f}(\mathbf{c}) = a \}.$$

Then $\mathcal{E}(n; \mathbf{r}, \alpha, \beta, a)$ is a t -absorption correcting code. Furthermore, there is a choice of $\alpha, \beta, \mathbf{r}$ and a , such that the redundancy of $\mathcal{E}(n; \mathbf{r}, \alpha, \beta, a)$ is at most

$$(4t-1)\log_q(n) + o(\log_q(n)).$$

Let $\mathbf{c} \in \mathcal{E}(n; \mathbf{r}, \alpha, \beta, a)$ and $\hat{\mathbf{c}} \in \mathcal{B}_t^{ab}(\mathbf{c})$. By applying splitting operations on $\hat{\mathbf{c}}$, we can find at most $q^{2t-2}n^{t-1}$ sequences in $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ (see the proof of Theorem IV.9 and Steps 2–3 in the proof of Claim IV.16). Among these sequences, there is a unique sequence $\tilde{\mathbf{c}}$ such that $\bar{f}(\tilde{\mathbf{c}}) = a$, and thus $\mathbf{c} = \tilde{\mathbf{c}}$. Since finding the sequences $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ takes polynomial time, together with Remark IV.15 we obtain that the function \bar{f} can be computed in polynomial time. Therefore, we can recover \mathbf{c} from $\hat{\mathbf{c}}$ in polynomial time.

Remark IV.18 We do not know if there exists an efficient encoder that can encode an arbitrary sequence into $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ (or $\mathcal{E}(n; \mathbf{r}, \alpha, \beta, a)$). Based on the results in this section, we can provide two, polynomial-time encodable and decodable, codes \mathcal{E}_1 and \mathcal{E}_2 , which can combat single-absorption and multiple-absorption errors, respectively. Details are deferred to Appendix C. Recall that the code $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$ is defined with four functions f, g, \hat{g}_1 and \hat{g}_2 . For any two codewords \mathbf{c} and \mathbf{c}' in $\mathcal{D}(n; \mathbf{r}, \alpha, \beta)$, we have

$$(f(\mathbf{c}), g(\mathbf{c}), \hat{g}_1(\mathbf{c}), \hat{g}_2(\mathbf{c})) = (f(\mathbf{c}'), g(\mathbf{c}'), \hat{g}_1(\mathbf{c}'), \hat{g}_2(\mathbf{c}')).$$

However, for two codewords in \mathcal{E}_1 , the above equation is not necessarily true. The same phenomenon holds for \mathcal{E}_2 .

V. OPTIMALITY OF THE CODES

In this section, we always assume $q \geq 2$. Let $\mathcal{C}_{max} \subseteq \Sigma_q^n$ be a code of maximum size that can correct a single absorption error. Let \mathcal{B}_n denote the set of all n -length sequences over $\Sigma_q \setminus \{0\}$, i.e., the sequences that do not contain the symbol 0. From Observation III.1 and Observation IV.1, we know that the code $\mathcal{C}_{max} \setminus \mathcal{B}_n$ can correct a single deletion of zero. So $|\mathcal{C}_{max}| \leq A_{q,n} + |\mathcal{B}_n| = A_{q,n} + (q-1)^n$, where $A_{q,n}$ denotes the maximum size of a code in $\Sigma_q^n \setminus \mathcal{B}_n$ that can correct a single deletion of zero. In this section, we will prove an upper bound of $A_{q,n}$, which implies that the codes given in the last two sections are optimal or near optimal in terms of redundancy. To that end, we follow the method proposed in [36], of which the authors proved a nonasymptotic upper bound of the size of a deletion correcting code (rather than zero-deletion correcting codes which we are interested in). The basic idea is to interpret our problem of upper bounding the size of codes as a linear programming problem. Inspired by [36], several researchers further developed this method and obtained many important results (see, for example, [37], [38]).

We need to introduce some terminologies first. A *hypergraph* \mathcal{H} is a tuple (V, \mathcal{E}) , where V is a finite nonempty set and \mathcal{E} is a collection of nonempty subsets of V . The set V is the *vertex set* of \mathcal{H} and the elements in V are called *vertices*. The elements in \mathcal{E} are called *hyperedges*. A *matching* of \mathcal{H} is defined to be a collection of pairwise disjoint hyperedges of \mathcal{H} . The matching number, denoted by $\nu(\mathcal{H})$, is the maximum size of a matching.

For our purpose, we define a hypergraph $\mathcal{H}_{q,n} = (\Sigma_q^{n-1}, \mathcal{E}_{q,n})$, where $\mathcal{E}_{q,n} = \{D_1^{(0)}(\mathbf{x}) : \mathbf{x} \in \Sigma_q^n \setminus \mathcal{B}_n\}$. Here $D_1^{(0)}(\mathbf{x}) \subseteq \Sigma_q^{n-1}$ is the set of sequences obtained by deleting exactly one zero from \mathbf{x} . For example, if $\mathbf{x} = 0110010111$, then $D_1^{(0)}(\mathbf{x}) = \{110010111, 011010111, 011001111\}$. Obviously, a set $\mathcal{C} \subseteq \Sigma_q^n \setminus \mathcal{B}_n$ is a zero-deletion correcting code if and only if $\{D_1^{(0)}(\mathbf{x}) : \mathbf{x} \in \mathcal{C}\}$ is a matching of $\mathcal{H}_{q,n}$, and hence $A_{q,n} = \nu(\mathcal{H}_{q,n})$. Therefore, the problem boils down to estimating $\nu(\mathcal{H}_{q,n})$.

Suppose that $\mathcal{H} = (V, \mathcal{E})$ is a hypergraph with $V = \{v_1, \dots, v_n\}$ and $\mathcal{E} = \{E_1, \dots, E_m\}$. Then the *incidence matrix* A of \mathcal{H} is of size $n \times m$ and is defined as follows:

$$A_{i,j} = \begin{cases} 1, & \text{if } v_i \in E_j, \\ 0, & \text{otherwise.} \end{cases}$$

Here $A_{i,j}$ is the element in the i th row and j th column of A .

The following lemma gives an upper bound of $\nu(\mathcal{H})$.

Lemma V.1 [36, Lemma 2.4] Let notations be as above. Then $\nu(\mathcal{H}) \leq \tau^*(\mathcal{H})$, where

$$\tau^*(\mathcal{H}) = \min \left\{ \sum_{i=1}^n w_i : A^T \mathbf{w} \geq \mathbf{1}, \mathbf{w} \geq \mathbf{0} \right\}.$$

Here A^T denotes the transpose of the matrix A , $\mathbf{w} = (w_1, \dots, w_n)^T$ is a column vector whose components are all nonnegative reals, $\mathbf{1}$ denotes the column vector whose components are all 1, $\mathbf{0}$ denotes the column vector whose components are all 0, and the inequalities are defined component-wise.

According to Lemma V.1, we have $A_{q,n} = \nu(\mathcal{H}_{q,n}) \leq \tau^*(\mathcal{H}_{q,n})$. By definition,

$$\tau^*(\mathcal{H}_{q,n}) = \min \left\{ \sum_{\mathbf{y} \in \Sigma_q^{n-1}} w(\mathbf{y}) : \sum_{\mathbf{y} \in D_1^{(0)}(\mathbf{x})} w(\mathbf{y}) \geq 1, \forall \mathbf{x} \in \Sigma_q^n \setminus \mathcal{B}_n, \right. \\ \left. \text{and } w(\mathbf{y}) \geq 0, \forall \mathbf{y} \in \Sigma_q^{n-1} \right\}.$$

For a sequence $\mathbf{z} \in \bigcup_{i=1}^{\infty} \Sigma_q^i$ of finite length, we let $r_0(\mathbf{z})$ be the number of runs of zeros in \mathbf{z} . For example, if $\mathbf{z} = 0110010111$, then $r_0(\mathbf{z}) = 3$. It is clear that $r_0(\mathbf{y}) \leq r_0(\mathbf{x})$ if $\mathbf{y} \in D_1^{(0)}(\mathbf{x})$. If $\mathbf{y} \in \mathcal{B}_{n-1}$, we let $w(\mathbf{y}) = 1$; otherwise, let $w(\mathbf{y}) = \frac{1}{r_0(\mathbf{y})}$. Then $w(\mathbf{y}) \geq 0$ and

$$\sum_{\mathbf{y} \in D_1^{(0)}(\mathbf{x})} w(\mathbf{y}) \geq \sum_{\mathbf{y} \in D_1^{(0)}(\mathbf{x})} \frac{1}{r_0(\mathbf{x})} = \frac{|D_1^{(0)}(\mathbf{x})|}{r_0(\mathbf{x})} = 1$$

for any $\mathbf{x} \in \Sigma_q^n \setminus \mathcal{B}_n$. The last equality follows from the fact $|D_1^{(0)}(\mathbf{x})| = r_0(\mathbf{x})$. Let $\mathcal{S} = \Sigma_q^{n-1} \setminus \mathcal{B}_{n-1}$. Since

$$\sum_{\mathbf{y} \in \Sigma_q^{n-1}} w(\mathbf{y}) = (q-1)^{n-1} + \sum_{\mathbf{y} \in \mathcal{S}} \frac{1}{r_0(\mathbf{y})}, \quad (9)$$

it remains to calculate $\sum_{\mathbf{y} \in \mathcal{S}} \frac{1}{r_0(\mathbf{y})}$. Note that $1 \leq r_0(\mathbf{y}) \leq \lceil \frac{n-1}{2} \rceil = \lfloor \frac{n}{2} \rfloor$ for any $\mathbf{y} \in \mathcal{S}$. However, these bounds are too loose and will only lead to $A_{q,n} \leq q^{n-1}$. Thus, a better bound is needed.

Lemma V.2 *For a given positive integer N , the number of integer solutions to the following equation*

$$a_1 + \dots + a_t = N$$

under the condition that $a_i \geq 0$ for all $i = 1, \dots, t$, is $\binom{N+t-1}{N}$. More generally, the number of integer solutions to the above equation under the condition that $a_i \geq p_i$ for all $i = 1, \dots, t$, is $\binom{N+t-(\sum_{i=1}^t p_i)-1}{N}$, where p_1, \dots, p_t are nonnegative integers.

Proof: The first conclusion is [39, Proposition 1.5]. To prove the general conclusion, let $a'_i = a_i - p_i$ for each $1 \leq i \leq t$. Then each a'_i is a nonnegative integer. The proof follows from the first conclusion. \square

Lemma V.3 *Let $n \geq 2$ be a positive integer. For any $1 \leq k \leq \lfloor \frac{n}{2} \rfloor$, the number of sequences \mathbf{y} in \mathcal{S} with the property $r_0(\mathbf{y}) = k$ is $\binom{n-2}{2k}(q-1)^{k+1} + 2\binom{n-2}{2k-1}(q-1)^k + \binom{n-2}{2k-2}(q-1)^{k-1}$.*

Proof: For $\mathbf{y} \in \mathcal{S}$ with $r_0(\mathbf{y}) = k$, we can write \mathbf{y} in the form

$$\mathbf{y} = a_0^m 0^{l_1} a_1^{m_1} \dots 0^{l_k} a_k^{m_k},$$

where $a_0, a_1, \dots, a_k \in \Sigma_q \setminus \{0\}$, $m_0, m_k \geq 0$, $l_i, m_j \geq 1$ for all $1 \leq i \leq k$ and $1 \leq j \leq k-1$. Let S_n be the number of solutions to the equation $\sum_{i=1}^k l_i + \sum_{j=0}^k m_j = n-1$. The following conclusions are clear from Lemma V.2:

- if $m_0, m_k \geq 1$, $S_n = \binom{n-2}{2k}$;
- if $m_0 = 0, m_k = 1$ or $m_0 = 1, m_k = 0$, $S_n = \binom{n-2}{2k-1}$;
- if $m_0 = m_k = 0$, $S_n = \binom{n-2}{2k-2}$.

Therefore, the number of sequences is $|\mathcal{S}| = \binom{n-2}{2k}(q-1)^{k+1} + 2\binom{n-2}{2k-1}(q-1)^k + \binom{n-2}{2k-2}(q-1)^{k-1}$. \square

From Lemma V.3 and Equation (9), we have

$$\sum_{\mathbf{y} \in \Sigma_q^{n-1}} w(\mathbf{y}) = (q-1)^{n-1} + \sum_{\mathbf{y} \in \mathcal{S}} \frac{1}{r_0(\mathbf{y})} \\ = (q-1)^{n-1} + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k} (q-1)^{k+1} \\ + 2 \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-1} (q-1)^k + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-2} (q-1)^{k-1} \quad (10)$$

To derive our desired result, we need the following lemma.

Lemma V.4 [19, Claim 2] *For integers $q \geq 2$, $n \geq 5$ and $n \geq q$, it holds that*

$$\sum_{k=1}^n \frac{1}{k} \binom{n}{k} (q-1)^k \leq \frac{q^{n+1}}{(q-1)(n-2)}.$$

Putting everything together, we can now present the main theorem of this section.

Theorem V.5 *Let notations be as above. For integers $q \geq 2$, $n \geq 12$ and $n \geq q$, it holds that $A_{q,n} \leq (q-1)^{n-1} + 1 + \frac{8q^{n-1}}{(q-1)(n-4)}$. In particular, the redundancy of \mathcal{C}_{max} is at least $\log_q(n) - \log_q(C_q)$, where C_q is a constant dependent on q and independent of n .*

Proof: For any $k \geq 1$, we have $k+1 \leq 2k$. Therefore,

$$\sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k} (q-1)^{k+1} \leq 2 \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{2k} \binom{n-2}{2k} (q-1)^{2k} \leq 2 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k. \quad (11)$$

Since $1/k \leq 2/(2k-1)$ and $k \leq 2k-1$, we have

$$2 \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-1} (q-1)^k \leq 4 \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{2k-1} \binom{n-2}{2k-1} (q-1)^{2k-1} \leq 4 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k. \quad (12)$$

By Equation (11), we have

$$\begin{aligned} \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-2} (q-1)^{k-1} &= 1 + \sum_{k=2}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-2} (q-1)^{k-1} \\ &\leq 1 + \sum_{k=2}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k-1} \binom{n-2}{2k-2} (q-1)^{k-1} \\ &= 1 + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor - 1} \frac{1}{k} \binom{n-2}{2k} (q-1)^k \\ &\leq 1 + 2 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k. \end{aligned} \quad (13)$$

Now combining Equation (10), Lemma V.4 and Equations (11) to (13), we obtain

$$\begin{aligned} \sum_{\mathbf{y} \in \Sigma_q^{n-1}} w(\mathbf{y}) &= (q-1)^{n-1} + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k} (q-1)^{k+1} \\ &\quad + 2 \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-1} (q-1)^k + \sum_{k=1}^{\lfloor \frac{n}{2} \rfloor} \frac{1}{k} \binom{n-2}{2k-2} (q-1)^{k-1} \\ &\leq (q-1)^{n-1} + 1 + 2 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k \\ &\quad + 4 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k + 2 \sum_{k=1}^{n-2} \frac{1}{k} \binom{n-2}{k} (q-1)^k \\ &\leq (q-1)^{n-1} + 1 + \frac{8q^{n-1}}{(q-1)(n-4)}. \end{aligned}$$

By our discussion at the beginning of this section, we have $|\mathcal{C}_{max}| \leq (q-1)^{n-1} + 1 + \frac{8q^{n-1}}{(q-1)(n-4)} + (q-1)^n$. When n is large enough, this implies $|\mathcal{C}_{max}| \leq C_q \frac{q^n}{n}$, where C_q is a constant that depends on q and independent of n . Therefore, the redundancy of \mathcal{C}_{max} is at least $\log_q(n) - \log_q(C_q)$. \square

Corollary V.6 *The code in Equation (2) (when $a = 0$) is optimal up to a constant and the code in Theorem IV.9 is optimal up to an $O(\log_q \log_q(n))$, in terms of redundancy.*

VI. A VARIANT OF THE ABSORPTION CHANNEL AND ITS CONNECTION WITH DELETION CHANNELS

In this section, we briefly discuss a variant of the absorption channel, which we call the *contraction* channel. Interestingly, we find that it is equivalent to the deletion channel, which has been extensively studied in recent years. Throughout this section, we assume that q is a fixed positive integer great than 2.

Definition VI.1 *Suppose that $\mathbf{x} \in \Sigma_q^n$ is the transmitted sequence and $\mathbf{y} \in \Sigma_q^{n-1}$ is the received sequence, where*

- $\mathbf{y} = x_1 \cdots x_{i-1} (x_i \boxplus x_{i+1}) x_{i+2} \cdots x_n$ for some $1 \leq i \leq n-1$, or
- $\mathbf{y} = x_1 \cdots x_{n-1}$.

Here $x_i \boxplus x_{i+1}$ is defined to be $x_i + x_{i+1} \pmod{q}$. For simplicity, in the rest of this section we will say that \mathbf{y} is obtained from \mathbf{x} by a contraction if \mathbf{y} is obtained from \mathbf{x} in this way.

With Definition VI.1 in hand, multiple contractions can be defined in a similar way that we defined multiple absorptions (see Definition II.2).

For any $\mathbf{x} \in \Sigma_q^n$ and any integer $t \in [1, n-1]$, we define

$$D_t(\mathbf{x}) = \{\mathbf{z} \in \Sigma_q^{n-t} : \mathbf{z} \text{ is a subsequence of } \mathbf{x}\}.$$

Let \mathcal{C} be a nonempty subset of Σ_q^n . If $D_t(\mathbf{c}) \cap D_t(\mathbf{c}') = \emptyset$ for any two distinct sequences $\mathbf{c}, \mathbf{c}' \in \mathcal{C}$, we say \mathcal{C} is a q -ary t -deletion correcting code. There are some known results on nonbinary t -deletion correcting codes with low redundancy [20], [40].

Next, We construct a bijection that connects between contractions and deletions. To that end, we use the following notation. For any $t \geq 0$ and $n \geq t+1$, let

$$A_q(n, t) = \{\mathbf{x} \in \Sigma_q^n : x_i = 0 \text{ for all } 1 \leq i \leq t\}$$

and

$$B_q(n, t) = \{\mathbf{y} \in \Sigma_q^{n+1} : y_i = 0 \text{ for all } 1 \leq i \leq t+1\}.$$

We define a mapping $\Phi_{n,t}$ from $A_q(n, t)$ to $B_q(n, t)$ as following:

$$\begin{aligned} \Phi_{n,t} : A_q(n, t) &\rightarrow B_q(n, t) \\ \mathbf{x} &\mapsto \mathbf{y} \end{aligned}$$

where $y_1 = 0$ and $y_i = \boxplus_{j=1}^{i-1} x_j$ for each $i \geq 2$. Clearly, the mapping $\Phi_{n,t}$ is a bijection. Indeed, for any $\mathbf{y} \in B_q(n, t)$, we have $\Phi_{n,t}^{-1}(\mathbf{y}) = x_1 \cdots x_n$, where $x_i = y_{i+1} - y_i \pmod{q}$ for each $1 \leq i \leq n$.

Lemma VI.2 Let $\mathbf{x} \in A_q(n, t)$ and $\mathbf{y} = \Phi_{n,t}(\mathbf{x})$, where t is a positive integer and $n \geq t+1$ is an integer.

- (1) t contractions in \mathbf{x} corresponds t deletions in \mathbf{y} .
- (2) t deletions in \mathbf{y} corresponds t contractions in \mathbf{x} .

Before proving the lemma, we give a simple example to demonstrate the idea.

Example VI.3 Let $n = 7, t = 1$ and consider the sequence $\mathbf{x} = 0121201$ over the ternary alphabet $\Sigma_3 = \{0, 1, 2\}$. Applying the bijection, we obtain $\Phi_{7,1}(\mathbf{x}) = \mathbf{y} = 00101001$. Now assume a contraction occurred in \mathbf{x} in position $i = 2$, i.e., we obtain $\mathbf{x}' = x_1(x_2 \boxplus x_3)x_4 \dots x_7 = 001201$. The corresponding $\mathbf{y}' = \Phi(\mathbf{x}') = 0001001$ can be obtained from \mathbf{y} by deleting y_3 .

Considering 2 consecutive contractions, let $\mathbf{x}'' = 01201$ be obtained by contracting $x_2 \boxplus x_3 \boxplus x_4$. The corresponding \mathbf{y}' is $\Phi(\mathbf{x}'') = 001001$ which can also be obtained by deleting y_3 and y_4 from \mathbf{y} .

We now prove the lemma.

Proof: (1). Suppose that \mathbf{x}' is obtained from \mathbf{x} by t contractions. Then

$$x'_i = \begin{cases} x_i, & \text{if } i < i_1, \\ x_{i+\sum_{j=1}^l s_j}, & \text{if } i_l - \sum_{j=1}^{l-1} s_j < i < i_{l+1} - \sum_{j=1}^l s_j \\ & \text{for some } 1 \leq l < k, \\ x_{i+\sum_{j=1}^k s_j}, & \text{if } i > i_k - \sum_{j=1}^{k-1} s_j, \\ \boxplus_{j=i_l}^{i_l+s_l} x_j, & \text{if } i = i_l - \sum_{j=1}^{l-1} s_j \text{ for some } 1 \leq l \leq k. \end{cases} \quad (14)$$

Here $s_l \geq 1$ for each $1 \leq l \leq k$, the sum $\sum_{l=1}^k s_l = t - t'$, $i_1 \geq 1$, $i_k + s_k \leq n - t'$ and $i_{l+1} - i_l > s_l$ for each $1 \leq l < k$. Let $\mathbf{y}' = \Phi_{n-t,0}(\mathbf{x}')$. Then \mathbf{y}' is obtained from \mathbf{y} by deleting y_{i_l+r} ($1 \leq l \leq k, 1 \leq r \leq s_l$) and $\mathbf{y}_{[n-t'+2, n+1]}$. Therefore, \mathbf{y}' is obtained from \mathbf{y} by t deletions.

(2). Suppose that \mathbf{y}' is obtained from \mathbf{y} by t deletions. Then there exist integers i_l, s_l ($1 \leq l \leq k$) satisfying $i_1 \geq 0$, $s_l \geq 1$ for all $l \geq 1$, $i_{l+1} - i_l > s_l$ for all $1 \leq l < k$ and $i_k + s_k \leq n - t'$, such that \mathbf{y}' is obtained from \mathbf{y} by deleting y_{i_l+r} for all $1 \leq l \leq k$ and $1 \leq r \leq s_l$ (where $\sum_{l=1}^k s_l = t - t'$) and $\mathbf{y}_{[n-t'+2, n+1]}$. Notice that $y_1 = \cdots = y_{t+1} = 0$. So we can assume $i_1 \geq 1$ and hence $y'_1 = 0$. Let $\mathbf{x}' = \Phi_{n-t,0}^{-1}(\mathbf{y}')$. By construction, we can see that \mathbf{x}' is as in Equation (14). Therefore, \mathbf{x}' is obtained from \mathbf{x} by t contractions. \square

Lemma VI.2 suggests that a t -contraction error in sequences in $A_q(n, t)$ is equivalent to a t -deletion error in sequences in $B_q(n, t)$. Therefore, a t -contraction correcting code in $A_q(n, t)$ is equivalent to a t -deletion correcting code in $B_q(n, t)$.

Observation VI.4 Let $a, b \in \Sigma_q$, $a \neq b$, and $0 < a, b \leq q-1$.

- If $x_i x_{i+1} \in \{0a, a0, 00\}$, then $N_0(\mathbf{x}) = N_0(\mathbf{y}) + 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq 0$. In other words, \mathbf{y} is obtained from \mathbf{x} by deleting one 0.
- If $x_i x_{i+1} = aa$ and $c = a \boxplus a$, then $N_a(\mathbf{x}) = N_a(\mathbf{y}) + 2$, $N_c(\mathbf{x}) = N_c(\mathbf{y}) - 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq a, c$.
- If $x_i x_{i+1} = ab$ and $c = a \boxplus b$, then $N_a(\mathbf{x}) = N_a(\mathbf{y}) + 1$, $N_b(\mathbf{x}) = N_b(\mathbf{y}) + 1$, $N_c(\mathbf{x}) = N_c(\mathbf{y}) - 1$ and $N_d(\mathbf{x}) = N_d(\mathbf{y})$ for all $d \neq a, b, c$.

With Observation VI.4 in hand, it is easy to construct codes correcting contraction errors, as we did in Theorem IV.5, Theorem IV.9 and Theorem IV.17. On the other hand, we can also construct codes via deletion correcting codes. Since these two kinds of constructions are straightforward, we omit the details.

VII. CONCLUSION

In this paper, we introduced and studied absorption channels, which are closely related to neural communication systems. We constructed codes with near-optimal redundancy for single-absorption errors and codes with logarithmic redundancy for multiple-absorption errors. We also explored a variant of the absorption channels called contraction channels and showed that they are equivalent to deletion channels, which have numerous practical applications. We hope that this new finding will inspire new approaches to the construction of deletion-correcting codes.

In Section V, we derived an upper bound on the size of single-absorption-correcting codes based on the fact that such codes must be able to correct the deletion of zeros. This bound implies that the redundancy of our single-absorption codes is optimal up to a constant or a term of $O(\log_q \log_q(n))$. However, this upper bound is not tight because a code that can correct a deletion of zeros is not necessarily a single-absorption-correcting code. Improving this upper bound would require a better estimate of the size of the 1-absorption ball $\mathcal{B}_1^{ab}(\mathbf{x})$ (see Equation (1)) for each \mathbf{x} , which appears to be a difficult task because $|\mathcal{B}_1^{ab}(\mathbf{x})|$ depends on the structure of \mathbf{x} . This problem is left for future research. There are other interesting future research directions, which include

- deriving an upper bound on the size of multiple-absorption codes;
- finding new constructions of multiple-absorption codes;
- finding efficient encoders for $\mathcal{D}(n; r, \alpha, \beta)$ and $\mathcal{E}(n; r, \alpha, \beta, a)$;
- exploring the general error model, in which a symbol's value may be decreased and the next symbol's value increased.

APPENDIX A

ENCODING AND DECODING ALGORITHMS FOR THE SET $\mathcal{R}_{q,n+5}$

In this section, we will give an algorithm that encodes an arbitrary sequence $\mathbf{x} \in \Sigma_q^n$ into $\mathcal{R}_{q,n+5}$. Since the encoding process is reversible, a decoding algorithm arises naturally. Throughout this section, it is assumed that $(c_1 - 4) \log_q(e)/(4q^4) \geq 5$ and $c_2 \log_q(e)/(4q^4) \geq 1$. For two finite sets A and B , let $f : A \rightarrow B$ be an injective mapping (A , B and f will be clear from the context). Then f induces a bijection f_A from A to its image $f(A)$. By abuse of notations, we denote the inverse of f_A by f^{-1} .

The basic idea of the encoding algorithm can be outlined as follows.

- 1) Find two consecutive patterns 0011 of distance larger than δ .
- 2) Delete a substring of length $\delta - 4$ between these two patterns. This process aims to decrease the distance between these two patterns.
- 3) Encode the position of this deleted substring and a compressed version of this substring into a block.
- 4) Insert this block into another position to make sure that this insertion does not introduce two consecutive patterns of distance larger than δ .
- 5) Continue this process until there are no two consecutive patterns 0011 of distance larger than δ .

First, we present a method to compress a length $\delta - 4$ sequence that does not contain 0011, into a shorter sequence. The following lemma follows similar ideas in [?, Observation 1] and [?, Proposition 1].

Lemma A.1 *Let \mathcal{S} be the set of all sequences of length $\delta - 4$ that do not contain 0011 as a substring. Then there exists an injective mapping $g : \mathcal{S} \rightarrow \Sigma_q^{\delta - \lceil \log_q(n) \rceil - 9}$. Furthermore, the two mappings g and g^{-1} can be computed in $O(n)$ time.*

Proof: Divide each $\mathbf{s} \in \mathcal{S}$ into $(\delta - 4)/4$ segments, each of length 4. In other words, represent \mathbf{s} as $\mathbf{s} = \mathbf{s}_1 \mathbf{s}_2 \cdots \mathbf{s}_{(\delta-4)/4}$, where $\mathbf{s}_i \in \Sigma_q^4$ for each $1 \leq i \leq (\delta - 4)/4$. Since $\mathbf{s}_i \neq 0011$, there are at most $q^4 - 1$ choices of \mathbf{s}_i . This implies that each \mathbf{s}_i can be represented by a symbol from the alphabet Σ_{q^4-1} , and a sequence \mathbf{s} can be represented by a sequence $\mathbf{u} \in \Sigma_{q^4-1}^{(\delta-4)/4}$. Let $n_{\mathbf{u}}$ be the number of q -ary symbols to represent \mathbf{u} . Then

$$\begin{aligned} n_{\mathbf{u}} &\leq \left\lceil \log_q (q^4 - 1)^{\frac{\delta-4}{4}} \right\rceil \\ &= \left\lceil \delta - 4 + \frac{\delta - 4}{4q^4} \log_q \left(1 - \frac{1}{q^4} \right)^{q^4} \right\rceil \end{aligned}$$

$$\leq \left\lceil \delta - 4 - \frac{\delta - 4}{4q^4} \log_q(e) \right\rceil.$$

The last inequality follows from the fact that the function $(1 - 1/x)^x$ is increasing in x when $x > 1$ and $\lim_{x \rightarrow \infty} (1 - 1/x)^x = 1/e$. Since $(c_1 - 4) \log_q(e)/(4q^4) \geq 5$ and $c_2 \log_q(e)/(4q^4) \geq 1$, we have $(\delta - 4) \log_q(e)/(4q^4) \geq \lceil \log_q(n) \rceil + 5$. So $n_{\mathbf{u}} \leq \delta - \lceil \log_q(n) \rceil - 9$. Recall that c_1 and c_2 are integers. Thus, the sequence \mathbf{u} (and \mathbf{s}) can be represented by a q -ary sequence of length $\delta - \lceil \log_q(n) \rceil - 9$.

The construction of g (and g^{-1}) is straightforward. Since each \mathbf{s}_i corresponds to a symbol from Σ_{q^4-1} , we can obtain \mathbf{u} from \mathbf{s} by replacing each \mathbf{s}_i by the symbol from Σ_{q^4-1} that corresponds to the value of its base- q representation. We then transform \mathbf{u} to a q -ary sequence \mathbf{v} of length $\delta - \lceil \log_q(n) \rceil - 9$. This can be done, for example, using a lookup table. Overall, transforming $\mathbf{s} \in \mathcal{S} \subseteq \Sigma_q^{\delta-4}$ into $\mathbf{v} \in \Sigma_q^{\delta - \lceil \log_q(n) \rceil - 9}$ can be done in $O(n)$ time. This process is reversible and g^{-1} can be computed in $O(n)$ time. \square

With this lemma, we describe our encoding algorithm in Algorithm 1. We note that since $q^{\lceil \log_q(n) \rceil} \geq n$, there is an injective mapping from $[2, n+1]$ to $\Sigma_q^{\lceil \log_q(n) \rceil}$. Let b be such a mapping. By building a lookup table, the two mappings b and b^{-1} can be computed in $O(n)$ time.

Algorithm 1 works as follows. We scan the sequence for 0011 starting from the end of the sequence and going backward. If there is a block between two consecutive appearances of 0011 which is longer than $\delta - 4$, the length- $(\delta - 4)$ suffix of that block is removed, compressed, and placed at the beginning of the sequences together with a pointer to its position and with 0011 appended to it.

Algorithm 1: Encoding an arbitrary sequence of length n into $\mathcal{R}_{q,n+5}$

Input: $\mathbf{x} \in \Sigma_q^n$
Output: $\mathbf{c} = \text{Enc}(\mathbf{x}) \in \mathcal{R}_{q,n+5}$

- 1 **Initialization**
- 2 $\mathbf{c} \leftarrow 1\mathbf{x}0011$, $i \leftarrow n + 5$, $d \leftarrow 1$
- 3 **while** $i \geq d + \delta$ **do**
- 4 **if** there is no $j \in [d + 3, i - 4]$ such that $\mathbf{c}_{[j-3,j]} = 0011$ **then**
- 5 $j \leftarrow d - 1$
- 6 **else**
- 7 find the largest $j \in [d + 3, i - 4]$ such that $\mathbf{c}_{[j-3,j]} = 0011$
- 8 **end**
- 9 **if** $i - j \leq \delta$ **then**
- 10 $i \leftarrow j$
- 11 **else**
- 12 $\mathbf{c} \leftarrow 0b(i - 4)g(\mathbf{c}_{[i-\delta+1,i-4]})0011\mathbf{c}_{[1,i-\delta]}\mathbf{c}_{[i-3,n+5]}$
- 13 $d \leftarrow d + \delta - 4$
- 14 **end**
- 15 **end**
- 16 **return** \mathbf{c}

Algorithm 2: Decoding $\text{Enc}(\mathbf{x}) \in \mathcal{R}_{q,n+5}$ into \mathbf{x}

Input: $\text{Enc}(\mathbf{x}) \in \mathcal{R}_{q,n+5}$
Output: \mathbf{x}

- 1 **Initialization**
- 2 $\hat{\mathbf{x}} \leftarrow \text{Enc}(\mathbf{x})$
- 3 **while** $\hat{x}_1 = 0$ **do**
- 4 $ind \leftarrow b^{-1}(\hat{\mathbf{x}}_{[2, \lceil \log_q(n) \rceil + 1]})$
- 5 $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_{[\delta-3, ind]}g^{-1}(\hat{\mathbf{x}}_{[\lceil \log_q(n) \rceil + 2, \delta-8]})\hat{\mathbf{x}}_{[ind+1, n+5]}$
- 6 **end**
- 7 $\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}}_{[2, n+1]}$
- 8 **return** $\hat{\mathbf{x}}$

Theorem A.2 Given any sequence $\mathbf{x} \in \Sigma_q^n$, Algorithm 1 outputs a sequence $\text{Enc}(\mathbf{x}) \in \mathcal{R}_{q,n+5}$.

Proof: We start with a detailed explanation of the idea behind Algorithm 1. In the Initialization step, a pattern 0011 is appended to the end of the input sequence \mathbf{x} since each sequence in $\mathcal{R}_{q,n+5}$ ends with 0011, and 1 is appended to the beginning of \mathbf{x} . This appended 1 serves as a marker for the beginning of the information sequence (or, alternatively, when to finish the decoding process). The variable d is a pointer to the position of this symbol. The index i is initialized to be $n+5$, which is the position of the last pattern 0011 in \mathbf{c} . The condition for continuing the while loop is $i \geq d + \delta$. This is because we want to find two consecutive patterns 0011 of distance larger than δ .

The idea for the while loop is to search patterns 0011 in the sequence, starting from the end of the sequence and going backward. Once the pattern 0011 is encountered at position i (i.e., the position of the last symbol in 0011 is i), we search for the next pattern 0011 that is closest to the one at position i . Assume there is a 0011 pattern in position $j < i$ (the position of the last symbol is j). If the distance between these two patterns is at most δ , we set $j \rightarrow i$ and repeat the process. Otherwise, we delete the substring $\mathbf{c}_{[i-\delta+1, i-4]}$ of length $\delta-4$ and then insert a block $0b(i-4)g(\mathbf{c}_{[i-\delta+1, i-4]})0011$ at the beginning. This block contains the position $b(i-4)$ of the deleted substring and the compressed version $g(\mathbf{c}_{[i-\delta+1, i-4]})$ of the deleted substring. Notice that the length of the block $0b(i-4)g(\mathbf{c}_{[i-\delta+1, i-4]})0011$ is $\delta-4$. So the deletion-insertion process does not change the length of the input sequence.

Recall that in the Initialization step, a symbol 1 was inserted at the beginning of the sequence and the variable d denotes the position of this symbol. Since the inserted block is on the left of c_d and the deleted substring is on the right of c_d , the value of d should increase by $\delta-4$ in step 13. In steps 9–14, either i decreases to j or d increases by $\delta-4$. So the while loop will end after a finite number of cycles. In other words, the algorithm will terminate after finite steps. In each loop, if two consecutive patterns of distance larger than δ are encountered, then the distance between them will decrease since a length $\delta-4$ substring between them is deleted. The distance of two existing consecutive patterns does not increase after the insertion of a block $0b(i-4)g(\mathbf{c}_{[i-\delta+1, i-4]})0011$. Besides, the insertion of a block will not introduce two consecutive patterns of distance larger than δ since the length of each block is $\delta-4$ and each block ends with 0011. So in the output sequence $\text{Enc}(\mathbf{x})$, the distance between two consecutive patterns is at most δ and thus $\text{Enc}(\mathbf{x}) \in \mathcal{R}_{q,n+5}$. \square

The time for searching i and j are both $O(n)$. The time for computing g and b are both $O(n)$. For each pair (i, j) , there are at most $O(n/\log_q(n))$ substrings of length $\delta-4$ to be deleted. Therefore, the time complexity of Algorithm 1 is $O(n^4/\log_q(n))$.

It is easy to see that the encoding process of Algorithm 1 is reversible. The decoding algorithm is presented in Algorithm 2. We give a brief explanation of the correctness of Algorithm 2. In the Initialization step of Algorithm 1, a symbol 1 was inserted at the beginning. This 1 was not destroyed during the encoding process. Each inserted block $0b(i-4)g(\mathbf{c}_{[i-\delta+1, i-4]})0011$ begins with 0. So in Algorithm 2, the condition $\hat{x}_1 = 0$ implies that $\hat{\mathbf{x}}$ should be decoded. If $\hat{x}_1 = 1$ (this is exactly the inserted 1), we just need to delete the first and the last four symbols in $\hat{\mathbf{x}}$. The remaining substring $\hat{\mathbf{x}}_{[2, n+1]}$ is the original sequence \mathbf{x} . The time complexity of Algorithm 2 is $O(n^2)$.

APPENDIX B PROOF OF THEOREM IV.8

If the pattern 0011 in the end of $\mathbf{z}_{l_x}^{\mathbf{x}}$ was destroyed, then this error is easy to detect and correct, since each codeword $\mathbf{x} \in \mathcal{D}_1$ ends with 0011. Therefore, we always assume that the absorption error does not destroy the pattern 0011 in the end of $\mathbf{z}_{l_x}^{\mathbf{x}}$.

If $|\mathbf{y}| = n$, then \mathbf{y} is error-free. If $|\mathbf{y}| = n-1$, then a single absorption happened. Notice that by calculating $g(\mathbf{y}) - r_2 \pmod{3}$, we can find $l_{\mathbf{y}} - l_{\mathbf{x}}$ (see Observation IV.7).

Case (1): $g(\mathbf{y}) - r_2 \equiv 0 \pmod{3}$. In this case, we have $l_{\mathbf{y}} = l_{\mathbf{x}}$ and so we can assume $\mathbf{z}^{\mathbf{y}} = (\mathbf{z}_1^{\mathbf{x}}, \dots, \mathbf{z}_{i-1}^{\mathbf{x}}, \mathbf{z}'_i, \mathbf{z}_{i+1}^{\mathbf{x}}, \dots, \mathbf{z}_{l_x}^{\mathbf{x}})$ for some $i \leq l_{\mathbf{x}}$, where \mathbf{z}'_i is obtained from $\mathbf{z}_i^{\mathbf{x}}$ by an absorption error and so $|\mathbf{z}'_i| = |\mathbf{z}_i^{\mathbf{x}}| - 1$. Therefore, we have

$$f(\mathbf{x}) - f(\mathbf{y}) \equiv \sum_{j=1}^{l_{\mathbf{x}}} j |\mathbf{z}_j^{\mathbf{x}}| - \sum_{j=1}^{l_{\mathbf{y}}} j |\mathbf{z}_j^{\mathbf{y}}| \equiv i (|\mathbf{z}_i^{\mathbf{x}}| - |\mathbf{z}'_i|) \equiv i \pmod{2n}.$$

Since $1 \leq i \leq l_{\mathbf{x}} \leq n/4$ and $i \equiv f(\mathbf{x}) - f(\mathbf{y}) \equiv r_1 - f(\mathbf{y}) \pmod{2n}$, we can find the value of i from $(r_1 - f(\mathbf{y})) \pmod{2n}$. This gives a window W of length at most $\delta = O(\log_q(n))$ in which the absorption error has occurred. Furthermore, since $(r_1 - f(\mathbf{y})) \pmod{2n}$ can be computed in $O(n)$ time, this window can be found in $O(n)$ time.

Case (2): $g(\mathbf{y}) - r_2 \equiv 2 \pmod{3}$. In this case, we have $l_{\mathbf{y}} = l_{\mathbf{x}} - 1$ and so we can assume $\mathbf{z}^{\mathbf{y}} = (\mathbf{z}_1^{\mathbf{x}}, \dots, \mathbf{z}_{i-1}^{\mathbf{x}}, \mathbf{z}'_i, \mathbf{z}_{i+2}^{\mathbf{x}}, \dots, \mathbf{z}_{l_x}^{\mathbf{x}})$ for some $i < l_{\mathbf{x}}$, where \mathbf{z}'_i is obtained from $\mathbf{z}_i^{\mathbf{x}}$ and $\mathbf{z}_{i+1}^{\mathbf{x}}$ by an absorption error which destroyed the 0011 in $\mathbf{z}_i^{\mathbf{x}}$ and so $|\mathbf{z}'_i| = |\mathbf{z}_i^{\mathbf{x}}| + |\mathbf{z}_{i+1}^{\mathbf{x}}| - 1$. Therefore, we have

$$\begin{aligned} f(\mathbf{x}) - f(\mathbf{y}) &= \sum_{j=1}^{l_{\mathbf{x}}} j |\mathbf{z}_j^{\mathbf{x}}| - \sum_{j=1}^{l_{\mathbf{y}}} j |\mathbf{z}_j^{\mathbf{y}}| \pmod{2n} \\ &= i |\mathbf{z}_i^{\mathbf{x}}| + (i+1) |\mathbf{z}_{i+1}^{\mathbf{x}}| - i |\mathbf{z}'_i| + \sum_{j=i+2}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| \pmod{2n} \end{aligned}$$

$$= i + |\mathbf{z}_{i+1}^{\mathbf{x}}| + \sum_{j=i+2}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| \pmod{2n}.$$

Since $0 < i + |\mathbf{z}_{i+1}^{\mathbf{x}}| + \sum_{j=i+2}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| < \sum_{j=1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| = n$, we can obtain the value of $i + |\mathbf{z}_{i+1}^{\mathbf{x}}| + \sum_{j=i+2}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}|$ from $(r_1 - f(\mathbf{y})) \pmod{2n}$.

For each $i \leq i' \leq l_{\mathbf{y}}$, we define

$$\Phi(i') = \sum_{j=i'+1}^{l_{\mathbf{y}}} |\mathbf{z}_j^{\mathbf{y}}| + i' = \sum_{j=i'+2}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| + i'.$$

Then we have

$$|\Phi(i) - (f(\mathbf{x}) - f(\mathbf{y}))| = |\mathbf{z}_{i+1}^{\mathbf{x}}| \leq \delta. \quad (15)$$

Besides, since $|\mathbf{z}_j^{\mathbf{x}}| \geq 4$ for all j , it holds that

$$\Phi(i' - 1) - \Phi(i') = |\mathbf{z}_{i'+1}^{\mathbf{x}}| - 1 \geq 3, \quad (16)$$

whenever $i' - 1 \geq i$, which in turn, implies that for $k \in \mathbb{N}$ such that $i' - k \geq i$,

$$\Phi(i' - k) - \Phi(i') = \sum_{j=i'-k+2}^{i'+1} |\mathbf{z}_j^{\mathbf{x}}| - k \geq 3k. \quad (17)$$

Now we can recover the desired window W in the following way. Sequentially compute $\Phi(i')$ for i' starting at $l_{\mathbf{y}}$ until we find an $i_0 \geq i$ such that $|\Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \leq \delta$. This i_0 does exist due to Equation (15). We claim that $i_0 - i \leq \frac{2}{3}\delta$. Otherwise, Equation (17) implies that

$$\begin{aligned} |\Phi(i) - (f(\mathbf{x}) - f(\mathbf{y}))| &= |\Phi(i) - \Phi(i_0) + \Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \\ &\geq |\Phi(i_0) - \Phi(i)| - |\Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \\ &> 3\frac{2}{3}\delta - \delta = \delta, \end{aligned}$$

which contradicts Equation (15). Since $|\mathbf{z}_i^{\mathbf{y}}| \leq 2\delta - 1$, $|\mathbf{z}_j^{\mathbf{y}}| \leq \delta$ for each $j \neq i$ and $i_0 - i \leq \frac{2}{3}\delta$, obtaining i_0 gives a window W of length $|W| \leq (i_0 - i + 1)\delta + \delta - 1 \leq \frac{2}{3}\delta^2 + 2\delta - 1 \leq c_4 \log_q^2(n)$ for some constant c_4 depending on c_1 and c_2 . This window contains the position where the absorption error happened.

Case (3): $g(\mathbf{y}) - r_2 \equiv 1 \pmod{3}$. In this case, $l_{\mathbf{y}} = l_{\mathbf{x}} + 1$ and so we can assume $\mathbf{z}^{\mathbf{y}} = (\mathbf{z}_1^{\mathbf{x}}, \dots, \mathbf{z}_{i-1}^{\mathbf{x}}, \mathbf{z}'_i, \mathbf{z}''_i, \mathbf{z}_{i+1}^{\mathbf{x}}, \dots, \mathbf{z}_{l_{\mathbf{x}}}^{\mathbf{x}})$ for some $i \leq l_{\mathbf{x}}$, where \mathbf{z}'_i and \mathbf{z}''_i are obtained from $\mathbf{z}_i^{\mathbf{x}}$ by an absorption error which created a new 0011 in $\mathbf{z}_i^{\mathbf{x}}$ and so $|\mathbf{z}'_i| + |\mathbf{z}''_i| = |\mathbf{z}_i^{\mathbf{x}}| - 1$. Therefore, we have

$$\begin{aligned} f(\mathbf{x}) - f(\mathbf{y}) &= \sum_{j=1}^{l_{\mathbf{x}}} j |\mathbf{z}_j^{\mathbf{x}}| - \sum_{j=1}^{l_{\mathbf{y}}} j |\mathbf{z}_j^{\mathbf{y}}| \pmod{2n} \\ &= i |\mathbf{z}_i^{\mathbf{x}}| - i |\mathbf{z}'_i| - (i+1) |\mathbf{z}''_i| - \sum_{j=i+1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| \pmod{2n} \\ &= i - |\mathbf{z}''_i| - \sum_{j=i+1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| \pmod{2n}. \end{aligned}$$

Since $1 \leq i \leq l_{\mathbf{x}} \leq n/4$, $4 \leq |\mathbf{z}''_i| \leq |\mathbf{z}_i^{\mathbf{x}}| - 5$ and $4 \leq |\mathbf{z}_j^{\mathbf{x}}|$, we have

$$-(n-6) \leq i - |\mathbf{z}''_i| - \sum_{j=i+1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| \leq n/4 - 4.$$

Here, $f(\mathbf{x}) - f(\mathbf{y})$ is chosen to be the unique integer $-n + 6 \leq a \leq n/4 - 4$ such that $f(\mathbf{x}) - f(\mathbf{y}) \equiv a \pmod{2n}$. In fact, we have $a = i - |\mathbf{z}''_i| - \sum_{j=i+1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}|$.

Similar to Case (2), for each $i \leq i' < l_{\mathbf{y}}$, we define

$$\Phi(i') = - \sum_{j=i'+2}^{l_{\mathbf{y}}} |\mathbf{z}_j^{\mathbf{y}}| + i' = - \sum_{j=i'+1}^{l_{\mathbf{x}}} |\mathbf{z}_j^{\mathbf{x}}| + i'.$$

Then we have

$$|\Phi(i) - (f(\mathbf{x}) - f(\mathbf{y}))| = |\mathbf{z}_i''| = |\mathbf{z}_i^x| - 1 - |\mathbf{z}_i'| \leq \delta - 5. \quad (18)$$

Besides, since $|\mathbf{z}_j^x| \geq 4$ for all j , it holds that

$$\Phi(i') - \Phi(i' - 1) = |\mathbf{z}_{i'}^x| + 1 \geq 5. \quad (19)$$

whenever $i' - 1 \geq i$,

Now we can recover the desired window W in the following way. Sequentially compute $\Phi(i')$ for i' starting at $l_y - 1$ until we find an $i_0 \geq i$ such that $|\Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \leq \delta - 5$. This i_0 does exist due to Equation (18). We claim that $i_0 - i \leq \frac{2}{5}\delta$. Otherwise, Equation (19) implies that

$$\begin{aligned} |\Phi(i) - (f(\mathbf{x}) - f(\mathbf{y}))| &= |\Phi(i) - \Phi(i_0) + \Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \\ &\geq |\Phi(i_0) - \Phi(i)| - |\Phi(i_0) - (f(\mathbf{x}) - f(\mathbf{y}))| \\ &> 2\delta - \delta = \delta, \end{aligned}$$

which contradicts Equation (18). Since $|\mathbf{z}_j^x| \leq \delta$ for each j and $i_0 - i \leq \frac{2}{5}\delta$, obtaining i_0 gives a window W of length $|W| \leq (i_0 - i + 1)\delta \leq \frac{2}{5}\delta^2 + \delta \leq c_5 \log_q^2(n)$ for some constant c_5 depending on c_1 and c_2 . This window contains the position where the absorption error happened.

In Case (2) and Case (3), $f(\mathbf{x}) - f(\mathbf{y})$ can be computed in linear time as the process for searching an i_0 . Therefore, the window W can be found in $O(n)$ time. Now let $c_3 = \max\{c_4, c_5\}$ and the proof is completed.

APPENDIX C

NON-BINARY ABSORPTION-CORRECTING CODES WITH EFFICIENT ENCODERS AND DECODERS

In this section, by applying the results in Section IV-B and Section IV-C, we give two new absorption-correcting codes that are polynomial-time encodable and decodable.

For a set A of size m , there exists an injection \mathcal{Q} from A to $\Sigma_q^{\lceil \log_q(m) \rceil}$. Under this mapping, each element a in A can be represented as a q -ary sequence $\mathcal{Q}(a)$ of length $\lceil \log_q(m) \rceil$. By building a lookup table, \mathcal{Q} and \mathcal{Q}^{-1} can be computed in $O(m)$ time.

A. Single-absorption correcting codes

Let f, g, \hat{g}_1 and \hat{g}_2 be as in Section IV-B. A message $\mathbf{x} \in \Sigma_q^n$ is encoded into

$$\mathcal{E}_1(\mathbf{x}) = \text{Enc}(\mathbf{x}) 010 f(\text{Enc}(\mathbf{x})) g(\text{Enc}(\mathbf{x})) \hat{g}_1(\text{Enc}(\mathbf{x})) \hat{g}_2(\text{Enc}(\mathbf{x})),$$

where $\text{Enc}(\cdot)$ is the encoder in Algorithm 1. Here the sequence $f(\text{Enc}(\mathbf{x})) g(\text{Enc}(\mathbf{x})) \hat{g}_1(\text{Enc}(\mathbf{x})) \hat{g}_2(\text{Enc}(\mathbf{x}))$ is defined to be the sequence

$$\mathcal{Q}((f(\text{Enc}(\mathbf{x})), g(\text{Enc}(\mathbf{x})), \hat{g}_1(\text{Enc}(\mathbf{x})), \hat{g}_2(\text{Enc}(\mathbf{x}))))).$$

Therefore, $f(\text{Enc}(\mathbf{x})) g(\text{Enc}(\mathbf{x})) \hat{g}_1(\text{Enc}(\mathbf{x})) \hat{g}_2(\text{Enc}(\mathbf{x}))$ is a sequence of length $\log_q(n) + 12 \log_q \log_q(n) + O(1)$.

Proposition C.1 *The code $\{\mathcal{E}_1(\mathbf{x}) : \mathbf{x} \in \Sigma_q^n\}$ is a single-absorption correcting code with redundancy $\log_q(n) + 12 \log_q \log_q(n) + O(1)$.*

Proof: The redundancy is clear from construction. Denote the length of the code by N . Suppose that $\mathbf{c} = \mathcal{E}_1(\mathbf{x})$ is the transmitted codeword and $\hat{\mathbf{c}}$ is obtained from \mathbf{c} by a single-absorption. Recall that the length of $\text{Enc}(\mathbf{x})$ is $n + 5$. So $c_{[n+6, n+8]} = 010$. A single-absorption can not affect $c_{[1, n+7]}$ and $c_{[n+8, N]}$ simultaneously. Therefore, the decoder can recover \mathbf{x} by the following procedure.

- If $\hat{c}_{n+6} = 0$, no error occurred in $c_{[1, n+7]}$ and so $\text{Enc}(\mathbf{x}) = \hat{c}_{[1, n+5]}$. Then the message \mathbf{x} can be decoded from $\text{Enc}(\mathbf{x})$ by applying Algorithm 2.
- If $\hat{c}_{n+6} = 1$, an absorption occurred in $c_{[1, n+7]}$. If $\hat{c}_{n+5} \neq 0$, no error occurred in $c_{[1, n+5]}$ and so $\text{Enc}(\mathbf{x}) = \hat{c}_{[1, n+5]}$. If $\hat{c}_{n+5} = 0$, then \hat{c}_{n+5} is obtained from $\text{Enc}(\mathbf{x})$ by an absorption. Notice that no error occurred in $c_{[n+8, N]}$. So we have $\hat{c}_{[n+8, N-1]} = f(\text{Enc}(\mathbf{x})) g(\text{Enc}(\mathbf{x})) \hat{g}_1(\text{Enc}(\mathbf{x})) \hat{g}_2(\text{Enc}(\mathbf{x}))$. By Theorem IV.9, we can recover $\text{Enc}(\mathbf{x})$ from $\hat{c}_{[1, n+4]}$ when given $f(\text{Enc}(\mathbf{x})) g(\text{Enc}(\mathbf{x})) \hat{g}_1(\text{Enc}(\mathbf{x})) \hat{g}_2(\text{Enc}(\mathbf{x}))$. Again, the message \mathbf{x} can be decoded from $\text{Enc}(\mathbf{x})$ by applying Algorithm 2.

□

Since $\text{Enc}(\cdot)$ is a polynomial-time encoder and the four functions f, g, \hat{g}_1 and \hat{g}_2 can be computed in polynomial time, the code in Proposition C.1 provides a polynomial-time encoder. By Algorithm 2 and the proofs of Proposition C.1 and Theorem IV.9, we can see that this code can also be decoded in polynomial time.

B. Multiple-absorption correcting codes

The construction of multiple-absorption correcting codes is more complicated. We first need the following trivial observation. Recall that $\mathcal{B}_t^{ab}(\mathbf{x})$ denotes the t -absorption ball centered at \mathbf{x} .

Observation C.2 *Let $\mathbf{c} = \mathbf{c}_1\mathbf{c}_2$ be a sequence. We assume that $\mathbf{c}_1 = \mathbf{c}'_10^t$ and $\mathbf{c}_2 = 0^t\mathbf{c}'_2$, where \mathbf{c}'_1 and \mathbf{c}'_2 are substrings of length at least $t + 1$. Suppose $\mathbf{c}_1 = \mathbf{c}_{[n_0+1, n_1]}$ and $\mathbf{c}_2 = \mathbf{c}_{[n_1+1, n_2]}$, where $0 = n_0 < n_1 < n_2 = |\mathbf{c}|$. Then for any $\hat{\mathbf{c}} \in \mathcal{B}_t^{ab}(\mathbf{c})$, we have $\hat{\mathbf{c}}_{[n_{i-1}+1, n_i-t]} \in \mathcal{B}_t^{ab}(\mathbf{c}_i)$ for each $i = 1, 2$.*

Let $\mathcal{E}_1(\cdot)$ be the encoder given in Proposition C.1. Define

$$\mathcal{E} = \{\mathcal{E}_1(\mathbf{x})0^t : \mathbf{x} \in \Sigma_q^n\}.$$

Then Proposition C.1 ensures that \mathcal{E} is a single-absorption correcting code. Denote the length of this code by n_1 . Then $n_1 = n + \log_q(n) + o(\log_q(n))$. Claim IV.16 and the proof of Lemma IV.14 (here $R(n_1) = R_{q, n_1}$ as defined in Equation (7)) imply that there is a mapping \bar{f} from \mathcal{E} to $\Sigma_q^{(4t-2)\log_q(n) + o(\log_q(n))}$, such that $\bar{f}(\mathbf{u}) \neq \bar{f}(\mathbf{u}')$ for any $\mathbf{u} \neq \mathbf{u}' \in \mathcal{E}$ and $\mathcal{B}_t^{ab}(\mathbf{u}) \cap \mathcal{B}_t^{ab}(\mathbf{u}') \neq \emptyset$. Furthermore, Remark IV.15 asserts that \bar{f} can be computed in polynomial time.

Now we are ready to give our construction. In this construction, a message $\mathbf{x} \in \Sigma_q^n$ is encoded into

$$\mathcal{E}_2(\mathbf{x}) = \mathcal{E}_1(\mathbf{x})0^t0^th(\mathbf{x})\text{Red}_{q,m}(0^th(\mathbf{x}))$$

where $h(\mathbf{x}) = \bar{f}(\mathcal{E}_1(\mathbf{x})0^t)$, m is the length of $0^th(\mathbf{x})$ and $\text{Red}_{q,m}(\cdot)$ is defined as in Equation (6).

Proposition C.3 *Let $t \geq 2$ be fixed. The code $\{\mathcal{E}_2(\mathbf{x}) : \mathbf{x} \in \Sigma_q^n\}$ is a t -absorption correcting code with redundancy $(4t - 1)\log_q(n) + o(\log_q(n))$.*

Proof: The redundancy is clear from construction. Denote the length of this code by n_2 . Suppose that $\mathbf{c} = \mathcal{E}_2(\mathbf{x})$ is the transmitted codeword and $\hat{\mathbf{c}}$ is obtained from \mathbf{c} by a t absorptions. By Observation C.2, we have $\hat{\mathbf{c}}_{[1, n_1-t]} \in \mathcal{B}_t^{ab}(\mathcal{E}_1(\mathbf{x})0^t)$ and $\hat{\mathbf{c}}_{[n_1+1, n_2-t]} \in \mathcal{B}_t^{ab}(0^th(\mathbf{x})\text{Red}_{q,m}(0^th(\mathbf{x})))$. According to Lemma IV.11 and Lemma IV.12, we can first recover $h(\mathbf{x})$ from $\hat{\mathbf{c}}_{[n_1+1, n_2-t]}$. Then Claim IV.16 (this claim holds for any single-absorption code) and the property of \bar{f} ensures that we can recover $\mathcal{E}_1(\mathbf{x})$ and thus \mathbf{x} in polynomial time by brute force searching. \square

Recall that $\mathcal{E}_1(\mathbf{x})$ and $h(\mathbf{x})$ can be computed in polynomial time. From Lemma IV.11 and Lemma IV.12, we know that $\text{Red}_{q,m}(0^th(\mathbf{x}))$ can be computed in polynomial time. Therefore, the code in Proposition C.3 provides a polynomial-time encoder. From the proof of Proposition C.3, we can see that this code can also be decoded in polynomial time.

REFERENCES

- [1] R. Gabrys, V. Guruswami, J. Ribeiro, and K. Wu, "Beyond Single-Deletion Correcting Codes: Substitutions and Transpositions," *IEEE Trans. Inf. Theory*, vol. Early Access, Aug. 2022. [Online]. Available: <https://ieeexplore.ieee.org/document/9869870>
- [2] S. K. Vashist, R. Tewari, I. Kaur, R. P. Bajpai, and L. M. Bharadwaj, "Smart-drug delivery system employing molecular motors," in *Proc. Int. Conf. Intell. Sens. Inf. Process. (ICISIP)*, Chennai, India, Jan. 2005, pp. 441–446.
- [3] S. Davis, "Biomedical applications of nanotechnology—implications for drug targeting and gene therapy," *Trends Biotechnol.*, vol. 15, no. 6, pp. 217–224, Jun. 1997.
- [4] J. M. Dubach, D. I. Harjes, and H. A. Clark, "Fluorescent Ion-Selective Nanosensors for Intracellular Analysis with Improved Lifetime and Size," *Nano Lett.*, vol. 7, no. 6, pp. 1827–1831, Jun. 2007.
- [5] J. Li, T. Peng, and Y. Peng, "A Cholesterol Biosensor Based on Entrapment of Cholesterol Oxidase in a Silicic Sol-Gel Matrix at a Prussian Blue Modified Electrode," *Electroanalysis*, vol. 15, no. 12, pp. 1031–1037, Jul. 2003.
- [6] P. Tallury, A. Malhotra, L. M. Byrne, and S. Santra, "Nanobioimaging and sensing of infectious diseases," *Adv. Drug Del. Rev.*, vol. 62, no. 4-5, pp. 424–437, Mar. 2010.
- [7] K. Yang, D. Bi, Y. Deng, R. Zhang, M. M. U. Rahman, N. A. Ali, M. A. Imran, J. M. Jornet, Q. H. Abbasi, and A. Alomainy, "A comprehensive survey on hybrid communication in context of molecular communication and terahertz communication for body-centric nanonetworks," *IEEE Trans. Mol. Biol. Multi-Scale Commun.*, vol. 6, no. 2, pp. 107–133, Nov. 2020.
- [8] I. F. Akyildiz, F. Brunetti, and C. Blázquez, "Nanonetworks: A new communication paradigm," *Comput. Networks*, vol. 52, no. 12, pp. 2260–2279, Aug. 2008.
- [9] N. Farsad, H. B. Yilmaz, A. Eckford, C.-B. Chae, and W. Guo, "A Comprehensive Survey of Recent Advancements in Molecular Communication," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 1887–1919, Thirdquarter 2016.
- [10] W. Pan, X. Chen, X. Yang, N. Zhao, L. Meng, and F. H. Shah, "A Molecular Communication Platform Based on Body Area Nanonetwork," *Nanomaterials*, vol. 12, no. 4, p. 722, Feb. 2022.
- [11] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M. Leung, "Body Area Networks: A Survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, Apr. 2011.
- [12] D. Malak and O. B. Akan, "Molecular communication nanonetworks inside human body," *Nano Commun. Networks*, vol. 3, no. 1, pp. 19–35, Mar. 2012.
- [13] W. Gerstner and W. M. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge university press, 2002.
- [14] D. Malak and O. B. Akan, "Communication theoretical understanding of intra-body nervous nanonetworks," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 129–135, Apr. 2014.
- [15] O. B. Akan, H. Ramezani, T. Khan, N. A. Abbasi, and M. Kescu, "Fundamentals of Molecular Information and Communication Science," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 306–318, Feb. 2017.
- [16] N. A. Abbasi, D. Lafci, and O. B. Akan, "Controlled Information Transfer Through An In Vivo Nervous System," *Sci. Rep.*, vol. 8, no. 1, pp. 1–12, Feb. 2018.
- [17] D. U. Silverthorn, *Human Physiology : An Integrated Approach*, 8th ed. Pearson, 2019.
- [18] R. Heckel, G. Mikutis, and R. N. Grass, "A Characterization of the DNA Data Storage Channel," *Sci. Rep.*, vol. 9, no. 1, pp. 1–12, Jul. 2019.
- [19] I. Smagloy, L. Welter, A. Wachter-Zeh, and E. Yaakobi, "Single-Deletion Single-Substitution Correcting Codes," in *Proc. Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 775–780.

- [20] W. Song, N. Polyanski, K. Cai, and X. He, “Systematic Codes Correcting Multiple-Deletion and Multiple-Substitution Errors,” *IEEE Trans. Inf. Theory*, vol. 68, no. 10, pp. 6402–6416, Oct. 2022.
- [21] V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions and reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, Feb. 1966.
- [22] J. Sima and J. Bruck, “On Optimal k -Deletion Correcting Codes,” *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 3360–3375, Jun. 2021.
- [23] V. Guruswami and J. Håstad, “Explicit Two-Deletion Codes With Redundancy Matching the Existential Bound,” *IEEE Trans. Inf. Theory*, vol. 67, no. 10, pp. 6384–6394, Oct. 2021.
- [24] K. Cheng, Z. Jin, X. Li, and K. Wu, “Deterministic Document Exchange Protocols, and Almost Optimal Binary Codes for Edit Errors,” in *Proc. Annu. Symp. Found. Comput. Sci. (FOCS)*, Paris, France, Oct. 2018, pp. 200–211.
- [25] B. Haeupler, “Optimal Document Exchange and New Codes for Insertions and Deletions,” in *Proc. Annu. Symp. Found. Comput. Sci. (FOCS)*, Baltimore, MD, USA, Nov. 2019, pp. 334–347.
- [26] R. R. Varshamov and G. M. Tenengolts, “Code Correcting Single Asymmetric Errors (in Russian),” *Avtomat. i Telemekh.*, vol. 26, no. 2, pp. 288–292, 1965.
- [27] N. J. A. Sloane, “On single-deletion-correcting codes,” *Codes and Designs*, vol. 10, pp. 273–291, May 2002.
- [28] K. Abdel-Ghaffar and H. Ferreira, “Systematic encoding of the Varshamov-Tenengol’ts codes and the Constantin-Rao codes,” *IEEE Trans. Inf. Theory*, vol. 44, no. 1, pp. 340–345, Jan. 1998.
- [29] J. Brakensiek, V. Guruswami, and S. Zbarsky, “Efficient Low-Redundancy Codes for Correcting Multiple Deletions,” *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3403–3410, May 2018.
- [30] R. Gabrys and F. Sala, “Codes Correcting Two Deletions,” *IEEE Trans. Inf. Theory*, vol. 65, no. 2, pp. 965–974, Feb. 2019.
- [31] J. Sima, N. Raviv, and J. Bruck, “Two Deletion Correcting Codes From Indicator Vectors,” *IEEE Trans. Inf. Theory*, vol. 66, no. 4, pp. 2375–2391, Apr. 2020.
- [32] G. Tenengolts, “Nonbinary codes, correcting single deletion or insertion (corresp.),” *IEEE Trans. Inf. Theory*, vol. 30, no. 5, pp. 766–769, Sept. 1984.
- [33] J. Sima, R. Gabrys, and J. Bruck, “Optimal systematic t -deletion correcting codes,” in *Proc. Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 769–774.
- [34] J. Sima and J. Bruck, “Optimal k -Deletion Correcting Codes,” in *Proc. Int. Symp. Inf. Theory (ISIT)*, Paris, France, Jul. 2019, pp. 847–851.
- [35] J. Sima, R. Gabrys, and J. Bruck, “Syndrome Compression for Optimal Redundancy Codes,” in *Proc. Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 751–756.
- [36] A. A. Kulkarni and N. Kiyavash, “Nonasymptotic Upper Bounds for Deletion Correcting Codes,” *IEEE Trans. Inf. Theory*, vol. 59, no. 8, pp. 5115–5130, Aug. 2013.
- [37] A. Fazeli, A. Vardy, and E. Yaakobi, “Generalized Sphere Packing Bound,” *IEEE Trans. Inf. Theory*, vol. 61, no. 5, pp. 2313–2334, Mar. 2015.
- [38] D. Cullina and N. Kiyavash, “Generalized sphere-packing bounds on the size of codes for combinatorial channels,” *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4454–4465, May 2016.
- [39] S. Jukna, *Extremal Combinatorics*, 2nd ed., ser. Texts in Theoretical Computer Science. An EATCS Series. Springer Berlin, Heidelberg, 2011.
- [40] J. Sima, R. Gabrys, and J. Bruck, “Optimal Codes for the q -ary Deletion Channel,” in *Proc. Int. Symp. Inf. Theory (ISIT)*, Los Angeles, CA, USA, Jun. 2020, pp. 740–745.