Tail-Erasure-Correcting Codes

Boaz Moav, *Student, IEEE,* Ryan Gabrys, *Member, IEEE,* Eitan Yaakobi, *Senior Member, IEEE*

***Abstract*—The increasing demand for data storage has prompted the exploration of new techniques, with molecular data storage being a promising alternative. In this work, we develop coding schemes for a new storage paradigm that can be represented as a collection of two-dimensional arrays. Motivated by error patterns observed in recent prototype architectures, our study focuses on correcting erasures in the last few symbols of each row, and also correcting arbitrary deletions across rows. We present code constructions and explicit encoders and decoders that are shown to be nearly optimal in many scenarios. We show that the new coding schemes are capable of effectively mitigating these errors, making these emerging storage platforms potentially promising solutions.**

***Index Terms*—Coding theory, DNA data storage, deletions, tail-erasure, RT-metric.**

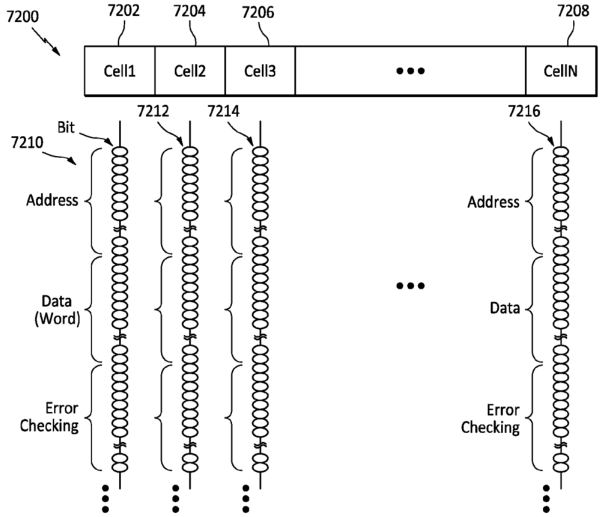


Figure 1. Illustration of a collection of Iridia's NMC [4, Fig. 72]

I. Introduction

THE DNA Data Storage market is rapidly growing and is expected to approach $2 billion (USD) in valuation by 2028 [1]. This rapid growth is driven by several promising technologies that depart from existing storage practices. Traditional approaches to DNA storage, including the pioneering works of [2], [3], typically represent data using individual base pairs of DNA (i.e., adenine (A) and thymine (T), cytosine (C) and guanine (G)).

More recently, companies and researchers have moved away from this paradigm, and in an effort to drive down the cost of synthesis and overcome some of the practical limitations of reading with traditional DNA sequencers, they have begun to investigate using collections of DNA molecules, sometimes referred to as cassettes, in order to encode logical '0's and '1's, such as in the Iridia system [4]. These DNA cassettes are sequentially chained together and stored within an atomic unit, which is referred to as a *nano-memory cell (NMC)* within a larger storage architecture.

In order to accommodate these new storage paradigms while also keeping the broader (traditional) storage technology stack intact, these memory cells are logically organized into larger collections of cells analogous to sectors in a hard drive. This organization is beneficial in several ways. First, it allows one to parallelize the read/write process enabling faster access times. In addition, this logical organization of NMCs can leverage addressing schemes that are typically employed in the context of traditional magnetic media storage thus reducing the potential cost or impact for transition.

Such systems can be modeled as collections of two dimensional binary arrays, which are referred to as a *DNA storage arrays*, whereby each row of the array represents a particular NMC and the sequence of bits in the row represents the logical information contained within that NMC. An illustration of such a system architecture is being shown in Figure 1 where each cell consists of an individual strand of DNA and this data can be partitioned into information pertaining to the address of the NMC, the data itself, and additional data such as ECC.

One of the key challenges, and one which is essential to making emerging DNA storage architectures practically viable, is constructing error correcting coding schemes. The design of such schemes not only further increases the already exceptional durability of DNA storage, but it also enables greater speeds and flexibility when it comes to reading information back from the underlying data storage medium. In this way, the problem of reading information from a DNA storage system is no longer equivalent to the challenges faced by existing biological DNA sequencers today which is to determine (exactly) the underlying sequence of nucleotides that comprises a given strand of DNA.

Designing error correcting coding schemes for DNA data storage systems is fundamentally different than the design of coding schemes for traditional storage media [5]. Traditional storage typically experiences errors in the form of substitutions whereas data stored within DNA can experience insertion, deletion, and substitution errors. Insertions can be the result of the improper blocking of some bits whereas deletions can be caused by a failure of the bit addition chemistry. On

B. Moav and E. Yaakobi are with the Henry and Marilyn Taub Faculty of Computer Science, Technion - Israel Institute of Technology, Haifa 3200003, Israel (e-mail: {boazmoav,yaakobi}@cs.technion.ac.il). R. Gabrys is with University of California, San Diego, California, USA (e-mail: rgabrys@ucsd.edu).

The research was funded by the European Union (ERC, DNAStorage, 865630). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. This work was also supported in part by NSF Grant CCF2212437.

top of this, cells within the DNA storage array can be lost either partially, which occurs when a defective bit prematurely terminates the chain, or the data within a cell can be corrupted completely. Initial experiments have reported error rates as high as 10% [6]. Despite these high numbers, we are unaware of any existing works that attempt to construct coding schemes for systems that can be modeled as DNA storage arrays, and this work represents the first effort towards the development of such a system.

Although the observed error rates are high, there appear to be certain commonly occurring error patterns, which may be useful in the design of future coding schemes. In particular, when a large number of errors occur, often only the tail end of the strand is affected. For shorthand, we say that an *e-tail-erasure* (TE for short) has occurred if we are unable to recover the last e symbols of the strand stored in the cell.

In the realm of error correction, we envision coding schemes that provide two levels of protection. The first level of protection corrects errors that may occur within each NMC, including insertions, deletions, and substitutions in DNA strands whereas the second level addresses more severe corruption. We propose a new class of codes, which we refer to as *tail erasure codes* (*TE codes* for short), for the second level of our envisioned two-level scheme. Our interest in codes with this structure is motivated by the fact that most of the errors that we are interested in correcting take the form of deletions and in particular as a burst of deletions. Since we know both the number of deletions that have taken place along with the location, we can treat the burst of deletions as a burst of erasures, which in general is a much easier problem to tackle than burst deletion correction.

The e-tail-erasure model that we study in this paper has also applications to the *slow-fading channel* as suggested in [7] and the *reliable-to-unreliable channel* as described in [8]. According to the reliable-to-unreliable channel model, data is transmitted over n channels in a distributed way, and each channel switches from reliable transmission into an unreliable one. This switch happens during the transmission, and the position where it occurs may be different in every channel. The rest of the transmission, since the moment where it became unreliable and until the end, is considered as an erasure. Therefore, each of these channels can be modeled as a row in a two dimensional array, where the last e bits of each row are considered as erasures since the channel became unreliable, and this results in a TE.

In this work, we initiate the study of TE codes. Our main result, which appears in Section III, is the development of novel error-correcting codes that we show in many cases are nearly optimal. We also consider a generalization of our problem to account for the setup where both deletions and TEs may occur and derive codes and bounds for this setup as well. This paper is organized as follows. In Section II we formally introduce our problem statement and discuss related work. In Section III we present a general construction for TE codes. Sections IV and V extend our problem to include the scenario where in addition to TEs, deletions are allowed to occur. In Section VI, bounds are derived for both TE codes and codes capable of correcting TEs and deletions. Finally, Section VII concludes the paper.

II. Notations, Definitions, Problems Statement, and Related Work

Throughout this paper, let $\mathbb{F}_2$ be the field of size two, denote $[n]=\{1,\dots,n\}$ and additionally let $\mathbb{Z}_{\geq 0}=\{m\in\mathbb{Z}\colon m\geq 0\}$. Also, for any vector $\boldsymbol{x}=(x_1,\dots,x_n)$, denote the subsequence of a vector $\boldsymbol{x}$, as $\boldsymbol{x}_{[i:j]}=(x_i,\dots,x_j)$. Moreover, denote a binary array as $\boldsymbol{X}=(\boldsymbol{x}_1,\dots,\boldsymbol{x}_n)\in{\mathbb{F}_2}^{n\times L}$, where $\{\boldsymbol{x}_i\}_{i=1}^n$ are binary row vectors of length L, which means $\boldsymbol{x}_i=(x_{i,1},\dots,x_{i,L})\in\mathbb{F}_2^L$ for any $i\in[n]$. Finally, for $\boldsymbol{x}=(x_1,\dots,x_n)\in\mathbb{Z}_{\geq 0}^n$, let $\|\boldsymbol{x}\|_1=\sum_{i=1}^n x_i$, and $\|\boldsymbol{x}\|_\infty=\max\{x_1,\dots,x_n\}$. Next, we lay the goals of this paper and the error models we study.

A. Definitions of the Error Models

If it is not possible to recover the last bits from a row in the array, it is said that a *tail-erasure* has occurred in this row. In this case, the row vector $\boldsymbol{x}_i\in\mathbb{F}_2^L$, $i\in[n]$, suffers from an *e-row-tail-erasure* if the bits $\{x_{i,L-e+1},\dots,x_{i,L}\}$ are all erased. This definition is generalized as follows.

Definition 1. Let $\boldsymbol{X}\in\mathbb{F}_2^{n\times L}$. It is said that $\boldsymbol{X}$ suffers from an *e-tail-erasure*, for a positive integer e, if there exists a positive integer $t\leq n$, positive integers $e_1,\dots,e_t$, such that $e_1+\cdots+e_t=e$, and t distinct row indices $i_1,i_2,\dots,i_t$, such that for $j\in[t]$, the i_j-th row in $\boldsymbol{X}$ suffers an e_j-row-tail-erasure.

For shorthand, this paper will refer to an e-tail-erasure simply as an *e-TE*. A code $\mathcal{C}\subseteq\mathbb{F}_2^{n\times L}$ capable of correcting any e-tail-erasure is referred as an *e-TE code*, or a *TE code* when e is not specified.

Subsequently, our model yields a definition to a distance function, which corresponds with the definition of the *m-metric*, as presented first in [9] and then further studied by [10]–[12]. Most of these papers focus on fields of large cardinality, and when reduced to the binary field, their constructions work for a very small number of rows, n.

In our paper we study specifically binary codes for arbitrary n, and present a new construction method, based on parity check matrices designed for this metric. This yields a new family of explicit codes, which in many cases is also better than the previous constructions in terms of the number of redundancy bits. This will be discussed more in detail in Sections II-B, and III. The first goal of the paper is as follows.

Problem 1. Construct optimal TE codes.

The results for this problem are presented in Sections III and VI-A and are summarized in Table I. Next, it is said that an array $\boldsymbol{X}\in\mathbb{F}_2^{n\times L}$ suffers from a *(t,s)-deletion* if at most $t\leq n$ rows suffer at most $s\leq L$ arbitrary deletions each. A code that can correct any (t,s)-deletion is referred to as a *(t,s)-DC* or a *DC code* for short when t,s are unspecified.

The second goal of this paper is summarized in the next problem.

Problem 2. Construct optimal DC codes.

The results for this problem are presented in Sections IV and VI-B. Finally, this paper studies the combined model of TEs and deletions. More specifically, a (t,s,e)-*tail-erasure-deletion*, abbreviated as a (t,s,e)-*TED*, is a combination of an e-TE and a (t,s)-deletion. It can be shown that the order of occurrence between the TEs and deletions in each row does not matter, and thus for simplicity it is assumed that first the e-TE has occurred, which is then followed by the (t,s)-deletion.

However, it is worth mentioning that since the output of this channel is just an array with some truncated rows, we cannot distinguish between the scenarios of a TE and an arbitrary deletion in a given row, based on the received row's length alone. For instance, assume that in a $(1,1,1)$-TED model, one receives an array with the first two rows each missing one bit. Then, any one of the following may hold: (i) The first and second rows each experience a single TE, (ii) The first row experiences a TE and the second row does not, or (iii) The second row experiences a TE and the first does not.

A code that can correct any (t,s,e)-TED is a (t,s,e)-*TED code*, or a *TED code* when t,s,e are unspecified. The third goal of the paper is described next.

Problem 3. Construct optimal TED codes.

The results for this problem are presented in Sections V and VI-C.

B. Related Work

Several previous works have proposed models and coding schemes for the emerging DNA storage system. In [13], the authors modeled the DNA storage system as an ordered set of sequences of a certain length where each sequence can experience insertions, deletions, and substitutions. Other works such as [14] and [15] considered the problem of developing constrained codes that result in input sequences that are less likely to experience errors that may result from the DNA storage channel. Recently, [16] developed codes for DNA storage that protect against errors but are also amenable to a broad class of coding constraints.

The motivation for our work stems from its connection to storage systems that fall under the DNA storage array where DNA molecules are stored within nano-memory cells such as [4] rather than as an unordered multiset. Under this setting, when a particular cell is severely corrupted the errors manifest themselves as erasures that occur typically at the very end of the strand. As discussed in Section I, one can model each cell as a row of a binary array, and therefore the entire system can be modeled as an $n \times L$ binary array, where n is the number of cells, and L is the length of the data in the strand that is saved in each cell. Moreover, since the cells are ordered by hardware design, there is no need to reserve bits to index the strands as in other DNA systems.

This model, where it is assumed that the indices are given, can also be applied to another common model of DNA storage systems, where all the information strands are stored in one container, which results a loss of ordering. Therefore, in this unordered information model, one usually dedicates part of the information for storing indices, as was suggested by [17] and studied by [18]–[20] and more.

Since it can either be assumed that the indices are error-free, or that there is a dedicated ECC for the indices, it can be verified easily that the final resulted read of the strands in the unordered model, after ordering and removing the indices, is a binary array as in our model.

The work on TE codes is strongly related to many works on codes over either the so-called m-metric (also called the *RT-metric* in other papers), where m corresponds to the number of rows in the array, which was introduced first in [9]. The m-metric is found to be useful in the case of evaluating codes that correct erasures in arrays, which occur at the end of each column. Therefore, it is easy to verify, using array transposition, that TE codes are equivalent to codes in the m-metric. Another relevant metric is the so-called ρ-*metric*, that was first studied in [10], where the metric corresponds to erasures that occur at the beginning of each row.

Moreover, in Section III-A, we define our ρ_{TE}-*metric* for evaluating TE codes. However, as will be discussed in this section, the m-metric, the ρ-metric and the ρ_{TE}-metric are all equivalent in terms of metric spaces isomorphism.

Lastly, we note that we prefer to define our new metric although it is equivalent to the previous one, since it better fits the error patterns studied in our paper, and will make the reading easier.

Next, a discussion is given, regarding the current state of the art results under the mentioned metrics. In [10], [11] the case of erasures at the beginning of the row was considered, and in [8], [9], [12] it was at the end of the columns. Moreover, in [9]–[12] mostly the case of large fields was considered, and several bounds were found, as will be described in Section VI-A. Also, two results of [11] for $q=2$ can be applied to Problem 1, but the problem the authors solved is more strict, in terms of linearity. Their model treats every row as a symbol in $\mathbb{F}_{q^L}$ and requires linearity of the length-n codewords over $\mathbb{F}_{q^L}$. Therefore, under our model, our construction is less restrictive, since linearity demands are only over $\mathbb{F}_q$. Lastly, in [8] a BCH code for the RT-metric is presented, and therefore a construction for a binary TE code can be derived where the length of each row is a power of two. However, their construction has concrete results only for TE codes of size 3×4, and for the general case there is no explicit lower bound on the cardinality of such codes. Therefore, one cannot derive a proper comparison between their results and the results in our paper.

Another related area of research involves the design of Universally Decodable Matrices (UDM), which is originated from the slow-fading channel problem, such as in [7], [11]. Specifically, in [11] a construction is given for the case where the rows are seen as a symbol over $\mathbb{F}_{q^L}$ and $q>n$, which does not generalize into the binary array model in this paper. Moreover, it is proved in [7, Theorem 4] that a necessary condition for the existence of $\{A_i\}_{i=0}^{n-1}$ UDM over $\mathbb{F}_q$ is that $q>n$, and therefore any construction based strictly on UDM, cannot be used in the binary case.

We note that our problem bears resemblance to the problem of coding for segmented edit channels [21], [22]. However, unlike the segmented edit model, we assume that the location of the tail-erasure is also known.

Nevertheless, we also face the model of (t, s)-deletions, that can be seen as a segmented edit channel, where at most t segments are erroneous and the separation to segments is given.

Our problem is also reminiscent of constructing unequal error protection (UEP) codes [23], [24], [25]. UEP codes are codes with the property that some information bits are more protected against a greater number of errors than other information symbols. Under the UEP model, each coordinate, say i, in a codeword is pre-assigned a protection which is referred to as f_i. Under this setting, if f errors occur in the underlying codeword, then we can determine the value of any coordinate j in the original codeword if $f_j \geq f$ regardless of whether the original codeword can be determined. One of the primary motivations for the development of UEP codes was to ensure that when errors occur, the values of coordinates with higher protection levels f_i can still be recovered [23]. As is the case across many existing storage solutions [26], our interest will be in the development of codes that ensure that so long as the number of errors that occur is below a certain threshold, all errors are correctable.

As will be discussed in the next section, the distance metric of interest for constructing TE codes is fundamentally different in the sense that our schemes aim to recover the entire codeword under the setting where the locations of the errors are non-independent and satisfy certain spatial properties in the underlying arrays.

III. Tail-Erasure Codes

In this section, constructions of TE codes, and also linear TE codes, are presented. The main result of the section is in Section III-C, where a construction of a linear e-TE code when $e \leq L$ is presented. Moreover, several results for the case where $e > L$ can be found in Section III-D.

A. The TE Distance

The focus of this subsection is introducing a suitable distance function for TE codes, while also providing a definition of a TE pattern. We begin with the definition of the ρ_{TE} distance, as follows.

Definition 2. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n), \boldsymbol{Y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n) \in \mathbb{F}_q^{n \times L}$. Then, for every $j \in [n]$,

$$\rho_{\text{TE}}(\boldsymbol{x}_j, \boldsymbol{y}_j) = \begin{cases} L - \min\{i : x_{j,i} \neq y_{j,i}\} + 1 & \text{if } \boldsymbol{x}_j \neq \boldsymbol{y}_j \\ 0 & \text{if } \boldsymbol{x}_j = \boldsymbol{y}_j \end{cases} .$$

And,

$$\rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) = \sum_{j=1}^{n} \rho_{\text{TE}}(\boldsymbol{x}_j, \boldsymbol{y}_j) \ .$$

This distance function is closely related to the ρ-metric, which was introduced in [10], while noting that the difference between the definitions is that $\rho(\boldsymbol{x}_j, \boldsymbol{y}_j) = \max\{i : x_{j,i} \neq y_{j,i}\}$ if $\boldsymbol{x}_j \neq \boldsymbol{y}_j$, and one can easily find the isomorphism between these two distance definitions, by reversing the order of each row. Moreover, it can be deduced that the ρ_{TE}-distance is a metric and from now on we will refer to it as the ρ_{TE}-metric or the TE-metric. The relation to the RT-metric is by transposing the array, and therefore, in terms of code parameters, all of the above mentioned metrics give equivalent codes and a comparison between the codes that are derived in each metric will be given at the end of Section III-C.

Next, a mathematical definition of a *TE pattern* is given. This definition resembles with the ones in [7], [11], but is more suitable for the notations of this paper.

Definition 3. For e, L, n, the *TE-pattern set* is defined to be

$$P(e, L, n) = \left\{ \boldsymbol{p} \in \mathbb{Z}_{\geq 0}^n : \|\boldsymbol{p}\|_1 \leq e, \|\boldsymbol{p}\|_\infty \leq L \right\} \ .$$

In other words, $P(e, L, n)$ is the set of all length-n vectors $\boldsymbol{p}$, which are called TE patterns, where the sum of entries in $\boldsymbol{p}$ is at most e and the value of each entry in $\boldsymbol{p}$ is at most L. A pattern $\boldsymbol{p} \in P(e, L, n)$ will be used next as an indicator, such that the i-th entry in $\boldsymbol{p}$ represents the number of erased bits in the i-th row of the array. Following this, the concept of a $\boldsymbol{p}$-pattern-TE is introduced.

Definition 4. Let $\boldsymbol{X} \in \mathbb{F}_2^{n \times L}$ and $\boldsymbol{p} \in P(e, L, n)$. Then, the *$\boldsymbol{p}$-pattern-TE* of $\boldsymbol{X}$, denoted by $\boldsymbol{X}^{(\boldsymbol{p})} \in (\mathbb{F}_2 \cup \{?\})^{n \times L}$, is defined as follows:

$$x_{i,j}^{(\boldsymbol{p})} = \begin{cases} ? & \text{if } p_i > 0 \text{ and } j \geq L - p_i + 1 \\ x_{i,j} & \text{else.} \end{cases} .$$

Conceptually, $\boldsymbol{X}^{(\boldsymbol{p})}$ represents the array that is obtained by replacing the last p_i symbols in the i-th row of $\boldsymbol{X}$ with the erasure symbol "?". An example is given next.

Example 1. Let $\boldsymbol{X} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$, $\boldsymbol{p} = (2, 1)$. Then, $\boldsymbol{X}^{(\boldsymbol{p})} = \begin{bmatrix} 1 & ? & ? \\ 0 & 0 & ? \end{bmatrix}$, is a $\boldsymbol{p}$-pattern-TE of $\boldsymbol{X}$.

Next, we provide a claim that connects Definitions 3 and 4 to the ρ_{TE}-metric definition.

Claim 1. Let $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{F}_2^{n \times L}$. Then,

$$\rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) = \min_{\boldsymbol{p} \in P(nL, L, n)} \left\{ \|\boldsymbol{p}\|_1 : \boldsymbol{X}^{(\boldsymbol{p})} = \boldsymbol{Y}^{(\boldsymbol{p})} \right\} .$$

Proof: By definition, for every row $j \in [n]$, the distance $\rho_{\text{TE}}(\boldsymbol{x}_j, \boldsymbol{y}_j)$ is exactly the minimal number of erasures required in the end of both $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ so they appear the same. In other words, this is the value of p_j for some vector $\boldsymbol{p} = (p_1, \ldots, p_n)$, such that $(\boldsymbol{X}^{(\boldsymbol{p})})_j = (\boldsymbol{Y}^{(\boldsymbol{p})})_j$, where $(\boldsymbol{X}^{(\boldsymbol{p})})_j$ denotes the j-th row of $\boldsymbol{X}^{(\boldsymbol{p})}$ and $(\boldsymbol{Y}^{(\boldsymbol{p})})_j$ denotes the j-th row of $\boldsymbol{Y}^{(\boldsymbol{p})}$. Thus, a vector $\boldsymbol{p}$ that obtains the minimum $\|\boldsymbol{p}\|_1$, for which $\boldsymbol{X}^{(\boldsymbol{p})} = \boldsymbol{Y}^{(\boldsymbol{p})}$ as in the left hand side, is constructed in a way such that every entry p_j, represents also the minimal number of erasures required in the end of both $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$, such that $(\boldsymbol{X}^{(\boldsymbol{p})})_j = (\boldsymbol{Y}^{(\boldsymbol{p})})_j$, or otherwise it results a contradiction to the minimality, since one can choose a smaller value for the j-th entry of $\boldsymbol{p}$. Hence, for the above minimal $\boldsymbol{p}$ it holds that

$$\|\boldsymbol{p}\|_1 = \sum_{i=1}^{n} p_i = \sum_{j=1}^{n} \rho_{\text{TE}}(\boldsymbol{x}_j, \boldsymbol{y}_j) = \rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) \ ,$$

which completes the proof.

■

A short example of computing the ρ_{TE}-metric by using a TE pattern is given next.

Example 2. Let $\boldsymbol{X},\boldsymbol{Y} \in \mathbb{F}_2^{2\times 3}$ be defined as follows,

$$\boldsymbol{X} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \ , \quad \boldsymbol{Y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} \ .$$

It can be readily verified that $\rho_{\mathrm{TE}}(\boldsymbol{X},\boldsymbol{Y}) = 3$ using $\boldsymbol{p} = (1,2)$, or using Definition 2.

Compared to the Hamming distance, note that under this model, erasures do not independently contribute to the ρ_{TE}-distance between two codewords. For example, consider the case where $\boldsymbol{X}$ is as in Example 2. Then, the vectors

$$\mathbf{Y} = \begin{bmatrix} 1 & \mathbf{1} & \mathbf{0} \\ 0 & 0 & \mathbf{0} \end{bmatrix} , \quad \mathbf{Z} = \begin{bmatrix} 1 & \mathbf{1} & 1 \\ 0 & 0 & \mathbf{0} \end{bmatrix} ,$$

are each at the same TE distance (of three) to $\mathbf{X}$ despite the fact that the set of coordinates in which $\mathbf{X}$ and $\mathbf{Z}$ differ is a subset of the set of coordinates where $\mathbf{X}$ and $\mathbf{Y}$ do not agree.

Finally, a definition of a minimum distance of a code under the ρ_{TE} distance follows.

Definition 5. Let $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$ be a TE code. Then, the *minimum ρ_{TE}-distance of* $\mathcal{C}$ is defined as follows

$$\rho_{\mathrm{TE}}(\mathcal{C}) = \min_{\boldsymbol{X},\boldsymbol{Y}\in\mathcal{C}:\boldsymbol{X}\neq\boldsymbol{Y}} \rho_{\mathrm{TE}}(\boldsymbol{X},\boldsymbol{Y}) \ .$$

Next, denote by $(n\times L, M, e+1)_{\mathrm{TE}}$ a code of cardinality M, and minimum ρ_{TE}-distance of $e+1$ over binary arrays of length $n\times L$. Then, a basic theorem regarding the minimum ρ_{TE}-distance of a code is given. This theorem illustrates one of the similarities between the properties of classical codes under the Hamming metric, and the TE codes in our paper, as we show that an $(n\times L, M, e+1)_{\mathrm{TE}}$ code can correct e erasures. Although this theorem is probably known, we present it here for the sake of completeness.

Theorem 1. A code $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$ is an e-TE-correcting code if and only if $\rho_{\mathrm{TE}}(\mathcal{C}) \geq e+1$.

The proof of Theorem 1 is provided in the appendix. Furthermore, Theorem 1 indicates that the constructions presented later in the paper are applicable to any type of TE pattern.

B. Tail-Erasure Linear Codes

Next, linear TE codes are constructed. First, the notation of a parity check matrix for linear codes over arrays under the ρ_{TE}-distance is presented.

Definition 6. Let $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$ be a linear code. A *TE parity check matrix* of $\mathcal{C}$ is a three-dimensional array $\boldsymbol{\mathcal{H}} \in \mathbb{F}_2^{r\times n\times L}$,

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \dots & \boldsymbol{h}_{1,L} \\ \vdots & \ddots & \vdots \\ \boldsymbol{h}_{n,1} & \dots & \boldsymbol{h}_{n,L} \end{bmatrix} , \boldsymbol{h}_{i,j} \in \mathbb{F}_2^r \ ,$$

where it holds that

$$\boldsymbol{X} = (x_{i,j})_{i\in[n],j\in[L]} \in \mathcal{C} \iff \sum_{i=1}^{n}\sum_{j=1}^{L} x_{i,j}\boldsymbol{h}_{i,j} = \mathbf{0} \ .$$

Denote $\boldsymbol{\mathcal{H}}^*$ as the set of vectors, which are the entries of $\boldsymbol{\mathcal{H}}$, i.e., $\{\boldsymbol{h}_{1,1},\dots,\boldsymbol{h}_{n,L}\}$. Then, $\dim(\mathcal{C}) = nL - \operatorname{rank}(\boldsymbol{\mathcal{H}}^*)$. Moreover, refer to a linear TE code $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$, as an $[n\times L, k, d]_{\mathrm{TE}}$ code, where $k = \dim(\mathcal{C})$ and $d = \rho_{\mathrm{TE}}(\mathcal{C})$.

Next, an expression of $\rho_{\mathrm{TE}}(\mathcal{C})$ for linear codes will be developed, using several preliminary definitions. First, let

$$\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p}) = \{\{\boldsymbol{h}_{i,j} \in \mathbb{F}_2^r : p_i > 0, \text{ and } j \geq L - p_i + 1\}\} \ ,$$

where $\boldsymbol{p} \in P(e,L,n)$. This multiset of columns represents the columns of $\boldsymbol{\mathcal{H}}$ which are located in the matching entries of the erasure pattern $\boldsymbol{p}$. The reason for choosing the multiset over a set, is the fact that there could be a repetition of columns in $\boldsymbol{\mathcal{H}}$, and it clearly affects the linear dependency of the multiset.

Example 3. Let $H = [\boldsymbol{h}_1,\dots,\boldsymbol{h}_7]$ be a parity check matrix of the $[7,4,3]$ Hamming code. Let

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \boldsymbol{h}_2 & \boldsymbol{h}_1 \\ \boldsymbol{h}_3 & \boldsymbol{h}_2 \\ \boldsymbol{h}_4 & \boldsymbol{h}_3 \\ \boldsymbol{h}_5 & \boldsymbol{h}_4 \\ \boldsymbol{h}_6 & \boldsymbol{h}_5 \\ \boldsymbol{h}_7 & \boldsymbol{h}_6 \\ \boldsymbol{h}_1 & \boldsymbol{h}_7 \end{bmatrix} \in \mathbb{F}_2^{3\times 7\times 2}, \ \boldsymbol{p} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2 \end{pmatrix} .$$

Then, $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p}) = \{\{\boldsymbol{h}_1,\boldsymbol{h}_1,\boldsymbol{h}_7\}\}$. Notice that there are repetitions of columns and therefore this multiset is linearly dependent. This is one of the cases where a TE pattern cannot be corrected, and this observation will be generalized in Claim 2.

In addition to the above, define the *TE-weight* of $\boldsymbol{X}$ as $w_{\mathrm{TE}}(\boldsymbol{X}) = \rho_{\mathrm{TE}}(\boldsymbol{X},\mathbf{0})$. It is possible to show that $\rho_{\mathrm{TE}}(\boldsymbol{X},\boldsymbol{Y}) = \rho_{\mathrm{TE}}(\boldsymbol{X}-\boldsymbol{Z},\boldsymbol{Y}-\boldsymbol{Z})$ for any $\boldsymbol{Z}$. Then, if $\boldsymbol{Z} = \boldsymbol{Y}$, $\rho_{\mathrm{TE}}(\boldsymbol{X},\boldsymbol{Y}) = w_{\mathrm{TE}}(\boldsymbol{X}-\boldsymbol{Y})$.

In [8] the authors indicated another similarity between the Hamming distance and the ρ_{TE}-distance, which is,

$$\rho_{\mathrm{TE}}(\mathcal{C}) = \min_{\boldsymbol{X}\in\mathcal{C}\setminus\{\mathbf{0}\}} w_{\mathrm{TE}}(\boldsymbol{X}) \ .$$

Lastly, a connection between the TE parity check matrix and the minimum ρ_{TE}-distance of its code is presented.

Claim 2. Let $\boldsymbol{\mathcal{H}}$ be a TE parity check matrix of a code $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$. Then, $\rho_{\mathrm{TE}}(\mathcal{C})$ is the largest integer d, such that for any $\boldsymbol{p} \in P(d-1,L,n)$, the vectors in $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ are linearly independent.

Proof: Let $\boldsymbol{X} \in \mathcal{C}$ with TE weight of $t > 0$. This implies the existence of $\boldsymbol{p} \in P(t,e,n)$, $\|\boldsymbol{p}\|_1 = t$ such that $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ is of cardinality t, and denote the column indices in $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ as J. From the definition of TE weight, any $(i,j) \notin J$ implies $x_{i,j} = 0$ and since $\boldsymbol{X}$ is a codeword,

$$\sum_{i=1}^{n}\sum_{j=1}^{e} \boldsymbol{h}_{i,j}x_{i,j} = \sum_{(i,j)\in J} \boldsymbol{h}_{i,j}x_{i,j} = \mathbf{0} \ ,$$

which implies $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ is a linearly dependent multiset by definition. Conversely, let $\mathcal{J}'(\boldsymbol{\mathcal{H}},\boldsymbol{p}')$ be a multiset of t' linearly dependent columns of $\boldsymbol{\mathcal{H}}$, for some $\boldsymbol{p}' \in P(t',e,n)$, and denote the column indices in $\mathcal{J}'(\boldsymbol{\mathcal{H}}, \boldsymbol{p}')$ as J', i.e., $\sum_{(i,j)\in J'} \alpha_{i,j}\boldsymbol{h}_{i,j} = \boldsymbol{0}$, which implies the existence of a codeword $\boldsymbol{Y}$ which is zero in every entry except the indices in J', thus $0 < w_{\text{TE}}(\boldsymbol{Y}) \leq t'$, and then $d \leq t'$, so we can conclude that there are no patterns $\boldsymbol{p}' \in P(t', L, n)$, for $t' < d$, such that the vectors in $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p}')$ are linearly dependent.

■

C. Linear Tail Erasure Code Construction

In this subsection, we give a construction for linear codes where $e \leq L$. Note that if $e \leq L$, then the first $L - e$ bits of a row cannot be erroneous. Thus, the subarray of size $n \times (L - e)$ of any word of size $n \times L$, taken as its first $L - e$ columns is error-free. Therefore, one can use the constructions in this subsection for the last e columns, and then prepend any first $L - e$ columns subarray to it, without affecting the bits of redundancy or the TE-correcting ability of the code. Hence, unless stated otherwise, it is assumed throughout this subsection $L = e$ so that the size of our arrays are $n \times (L = e)$. In Section III-D a discussion on codes where $e > L$ is given, and also a construction of a linear code for this case will be given in Section III-D.

First, since one can take a parity check matrix and construct a generator matrix from it, we can derive an efficient (polynomial-time) encoding scheme. For decoding, since the locations of the errors are known under this model, a system of linear equations can be used to correct erasures, using the same parity check matrix. Therefore, an efficient decoding scheme is also implied. Following the above discussion, a construction of an $[n \times (d-1), k, d]_{\text{TE}}$ code, $\mathcal{C}$, is presented, using a TE parity check matrix.

Construction 1. Let d be an odd positive integer and denote $t = \frac{d-1}{2}$. Also, let $\mathcal{C}_B$ be a binary $[nt, k_B, d]$ error-correcting code, referred as the *base code*, with a parity check matrix $H_B = [\boldsymbol{h}_1\, \boldsymbol{h}_2 \cdots \boldsymbol{h}_{nt}]$. Then,

$$\boldsymbol{\mathcal{H}} = \left[\begin{array}{ccc|ccc} \boldsymbol{h}_1 & \cdots & \boldsymbol{h}_t & \boldsymbol{h}_{2t} & \cdots & \boldsymbol{h}_{t+1} \\ \boldsymbol{h}_{t+1} & \cdots & \boldsymbol{h}_{2t} & \boldsymbol{h}_{3t} & \cdots & \boldsymbol{h}_{2t+1} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_{(n-1)t+1} & \cdots & \boldsymbol{h}_{nt} & \boldsymbol{h}_t & \cdots & \boldsymbol{h}_1 \end{array}\right],$$

is a TE parity check matrix of a TE-correcting code $\mathcal{C}_{\text{TE}}$.

Remark 1. For $n = 2$, this construction is degenerate, since it requires a $[2t, k_B, 2t+1]$ base code.

For example, the code with the TE parity check matrix $\boldsymbol{\mathcal{H}}$ in Example 3 is a $[7{\times}2, 11, 3]_{\text{TE}}$ code. The next claim emphasizes an important property of Construction 1.

Claim 3. Let $\boldsymbol{p} \in P(e, 2t, n)$, where $e \leq 2t$, and $\boldsymbol{\mathcal{H}}$ as in Construction 1. Then, the multiset $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$ does not contain any column vector with multiplicity greater than 1.

Proof: Assume, for the sake of contradiction, that there exists a column vector with multiplicity 2 in $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$ (according to the construction it cannot have a greater multiplicity). By definition, for an arbitrary $k \in [nt]$, $\boldsymbol{h}_k$ appears in $\boldsymbol{\mathcal{H}}$ at locations (i, j) and $(i-1, 2t-j+1)$ (up to modulo n and $2t$, respectively). But this implies that $\boldsymbol{h}_{i,j}$ and $\boldsymbol{h}_{i-1,2t-j+1}$ are both in $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$, which is not possible since $j + 2t - j + 1 = 2t + 1 > e = \|\boldsymbol{p}\|_1$, and $\boldsymbol{p} \in P(e, 2t, n)$.

■

The following presents a proof of the correctness of Construction 1, utilizing Claim 3.

Theorem 2. Let $\mathcal{C}_B$ be the $[nt, k_B, d]$ base code, where $t = \frac{d-1}{2}$. Let $\mathcal{C}$ be the resulting TE-correcting code from Construction 1. Then, $\mathcal{C}$ is an $[n \times 2t, k_B + nt, d]_{\text{TE}}$ code. Furthermore, the redundancy of the code $\mathcal{C}$ is $nt - k_B$.

Proof: One can verify that this construction yields a code of length $n \times 2t$ and that $\dim(\mathcal{C}) = 2nt - \text{rank}(\boldsymbol{\mathcal{H}}) = 2nt - (nt - k_B) = k_B + nt$, since $\boldsymbol{\mathcal{H}}$ is constructed from the columns of H_B, repeated two times, and therefore $\text{rank}(\boldsymbol{\mathcal{H}}) = \text{rank}(H_B) = nt - k_B$. It is left to prove that $\rho_{\text{TE}}(\mathcal{C}) = d$. Based on Claim 2, to show that $\rho_{\text{TE}}(\mathcal{C}) \geq d$, it is required to show that for any arbitrary $\boldsymbol{p} \in P(2t, 2t, n)$, the multiset $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$, is linearly independent. Since every $2t = d - 1$ columns in the base code are linearly independent, the fact that $\rho_{\text{TE}}(\mathcal{C}) \geq d$, immediately follows from Claim 3. Lastly, the fact that $\rho_{\text{TE}}(\mathcal{C}) = d$ can be easily verified, e.g. using the vector $\boldsymbol{p} = (2t, 0, \ldots, 0, 1)$, which results in $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$, which is a linearly dependent multiset.

■

Construction 1 results in codes with odd minimum ρ_{TE}-distance. The case of even minimum ρ_{TE}-distance is considered next. First, note that for minimum ρ_{TE}-distance 2, it is straightforward to see that a single parity bit suffices. For even minimum ρ_{TE}-distance greater than 2, denote again $t = \frac{d-1}{2}$, where d is odd. Then, let $\mathcal{C}_B$ be the $[nt+1, k_b, d+1]$ base code, i.e., a code with even minimum (Hamming) distance $d+1$ (w.l.o.g, based on the same base code from Construction 1, with odd minimum distance, after adding one parity bit, so that $d+1$ is even), where $H_B^* = [\boldsymbol{h}_1 \cdots \boldsymbol{h}_{nt+1}]$ is a parity check matrix of $\mathcal{C}_B$. Then,

$$\boldsymbol{\mathcal{H}}^* = \left[\begin{array}{ccc|c|ccc} h_1 & \cdots & h_t & h_{nt+1} & h_{2t} & \cdots & h_{t+1} \\ h_{t+1} & \cdots & h_{2t} & h_{nt+1} & h_{3t} & \cdots & h_{2t+1} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ h_{(n-1)t+1} & \cdots & h_{nt} & h_{nt+1} & h_t & \cdots & h_1 \end{array}\right].$$

Similarly to Construction 1, it can be shown that $\boldsymbol{\mathcal{H}}^*$ is a parity check matrix for a $[n \times (2t+1), k_b + n(t+1) - 1, d+1]_{\text{TE}}$ code. Therefore, the redundancy of such a code is $nt - k_b + 1$. In order to analyze the redundancy result of Construction 1, and its extension for the even minimum ρ_{TE}-distance case, let $R(n, d)$ be the minimum redundancy of any length-n linear code with minimum (Hamming) distance d. It is known that [27, p. 160-161],

$$R(n,d) \leq \begin{cases} \frac{d-1}{2}\lceil \log(n+1) \rceil, & d \text{ is odd} \\ 1 + \left(\frac{d}{2} - 1\right)\lceil \log(n+1) \rceil, & d \text{ is even.} \end{cases}$$

Next, for an $[n \times (d-1), k, d]_{\text{TE}}$ code that can be achieved by Construction 1, let $R_{\text{TE}}(n, d)$ be the minimum number of redundancy bits of the code. The next corollary considers the redundancy of Construction 1 and provides an upper bound on the redundancy of TE-codes.

Corollary 1. It holds that

1) $R_{\mathrm{TE}}(n,d) \le R\left(\frac{n(d-1)}{2}, d\right)$ and thus, when d is odd,
$$R_{\mathrm{TE}}(n,d) \le \frac{d-1}{2}\left\lceil \log\left(n\left(d-1\right)+2\right)\right\rceil - \frac{d-1}{2} \ .$$
2) $R_{\mathrm{TE}}(n,d) \le R\left(\frac{n(d-2)+2}{2}, d\right)$ and thus, when d is even and greater than 2,
$$R_{\mathrm{TE}}(n,d) \le \left(\frac{d}{2}-1\right)\left\lceil \log(n\left(d-2\right)+4)\right\rceil - \frac{d-4}{2} \ .$$

Note that a trivial construction of TE codes can be achieved by considering the array of size $n \times (d-1)$ as a vector of length $n(d-1)$, and then correcting $d-1$ TEs using a linear $[n(d-1),k,d]$ code. This was presented in [8] as the *vector construction*. The minimal number of redundancy bits of the vector construction will be denoted by $R\left(n(d-1),d\right)$. Then, from the previous discussion along with Corollary 1,
$$R\left(n(d-1),d\right) \ge R\left(\frac{n(d-1)}{2},d\right) \ge R_{\mathrm{TE}}(n,d) \ ,$$
for any n and odd d, and similarly for the even d case. More explicitly, the savings in Construction 1 is roughly $d/2$ bits of redundancy. Next, $[n \times (d-1),k,d]_{\mathrm{TE}}$ codes are presented, which imply some additional constructive upper bounds on the redundancy of TE codes.

Claim 4. Let n be a positive integer and $r = \lceil \log_2(n+1)\rceil$. Then,

1) An $[n \times 2, 2n-r, 3]_{\mathrm{TE}}$ code exists, and thus $R_{\mathrm{TE}}(n,3) \le \lceil \log_2(n+1)\rceil$.
2) An $[n \times 3, 3n-r-1, 4]_{\mathrm{TE}}$ code exists, and thus $R_{\mathrm{TE}}(n,4) \le \lceil \log_2(n+1)\rceil + 1$.

Proof: Let $\mathcal{C}_B$ be an $[n,n-r,3]$ base code, which is a shortening of a (2^r-1)-length Hamming code (where if $r = \log_2(n+1)$ exactly, no shortening is required), and $\mathcal{C}_B^*$ be a $[n+1,n-r,4]$ code, which is the extended version of $\mathcal{C}_B$, which one gets by adding a parity bit. Therefore,

1) Using Theorem 2, there exists a $[n \times 2, 2n-r, 3]_{\mathrm{TE}}$ code.
2) From the discussion on TE codes with even $\rho_{\mathrm{TE}}(\mathcal{C})$, there exists a $[n \times 3, 3n-r-1, 4]_{\mathrm{TE}}$ code.

■

By Claim 4 one derives an explicit upper bound on the redundancy of codes which correct 2 or 3 TEs, based on Construction 1. Next, another upper bound on the redundancy of codes that correct 4 TEs is derived from linear cyclic codes, using a specific construction which is different than Construction 1.

Claim 5. Let $n > 0$ be an integer, $m = \lceil \log_2(n+5)\rceil$ and $r = 2m+1$. Then, there exist a $[n \times 4, 4n-r, 5]_{\mathrm{TE}}$ code.

Proof: For this proof, a construction is given, similar to Construction 1, but with different ordering of the entries. Let $\mathcal{C}_B$ be an $[n+4, n+4-r, \ge 6]$ base code, which is a shortening (if needed) of the $[2^m-1, 2^m-2m-2, \ge 6]$ cyclic code in [27, Example 8.10], with a parity check matrix $H = [\boldsymbol{h}_1 \ \boldsymbol{h}_2 \cdots \boldsymbol{h}_{n+4}]$. Then,
$$\boldsymbol{\mathcal{H}} = \left[\begin{array}{cc|cc} \boldsymbol{h}_{n+4} & \boldsymbol{h}_{n+3} & \boldsymbol{h}_2+\boldsymbol{h}_3 & \boldsymbol{h}_1 \\ \boldsymbol{h}_{n+4} & \boldsymbol{h}_{n+3} & \boldsymbol{h}_3+\boldsymbol{h}_4 & \boldsymbol{h}_2 \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{h}_{n+4} & \boldsymbol{h}_{n+3} & \boldsymbol{h}_{n+1}+\boldsymbol{h}_{n+2} & \boldsymbol{h}_n \end{array}\right] ,$$
is a TE parity check matrix of $\mathcal{C}$, an $[n \times 4, 4n-r, 5]$ TE-correcting code. It is clear why the length is $n \times 4$ and the dimension is $4n-r$, since the redundancy stays the same as in the base code. What is left to prove is that $\rho_{\mathrm{TE}}(\mathcal{C}) = 5$, and the method will be to show that for any $\boldsymbol{p} \in P(4,4,n)$, the vectors in $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ are linearly independent. It can be verified that every vector $\boldsymbol{p}$ that does not contain exactly two entries of value 2 satisfies the condition of $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ being linearly independent, since it will be a sum of at most 5 distinct vectors of H, which is a parity check matrix of a code with minimum Hamming distance of at least 6.

Therefore, observe next on the case where there are two entries of $\boldsymbol{p}$, say $\boldsymbol{p}_i, \boldsymbol{p}_j$, where $i < j$, such that $\boldsymbol{p}_i = \boldsymbol{p}_j = 2$. If $j - i \le 2$, the sum of elements in $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ is again only at most 5 distinct vectors of H. So the last case one needs to check is where $j - i > 2$, therefore $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ can be written as $\{\{h_i, h_{i+1}+h_{i+2}, h_j, h_{j+1}+h_{j+2}\}\}$. To verify those vectors are independent, observe the equation, $\alpha_1 h_i + \alpha_2 h_{i+1} + \alpha_3 h_{i+2} + \alpha_4 h_j + \alpha_5 h_{j+1} + \alpha_6 h_{j+2} = 0$, where $\alpha_1, \ldots, \alpha_6 \in \mathbb{F}_2$, and all the six elements are distinct.

It is assumed that either $\alpha_1 = \alpha_2 = \cdots = \alpha_6 = 0$ or $\alpha_1 = \alpha_2 = \cdots = \alpha_6 = 1$, since there cannot be a linear dependent set of five distinct vectors. If those are all zero, the proof is done. So, assume, for sake of contradiction, that $h_i + h_{i+1} + h_{i+2} + h_j + h_{j+1} + h_{j+2} = 0$, which means there is a codeword in $\mathcal{C}_B$ of Hamming weight 6, with ones only in the entries $\{i, i+1, i+2, j, j+1, j+2\}$, denoted as $\boldsymbol{X}_1$. Since the code is cyclic, there exists a codeword with ones only in the entries $\{i+1, i+2, i+3, j+1, j+2, j+3\}$, denoted as $\boldsymbol{X}_2$. But then, also the sum $\boldsymbol{X} = \boldsymbol{X}_1 + \boldsymbol{X}_2$ is another codeword, but the Hamming weight of $\boldsymbol{X}$ is at most 4, which is a contradiction.

■

One can verify that $2\lceil \log_2(n+5)\rceil + 1 < 2\lceil \log_2(2n+1)\rceil$ for $n \ge 4$, if $n \notin \bigcup_{i=3}^{\infty}\{2^i-4, 2^i-3, 2^i-2, 2^i-1\}$. This implies that for most (but not all) values of n, the construction in Claim 5 gives a tighter upper bound for $d = 5$ than Construction 1 by one bit.

To conclude, the results are summarised in Table I, where the upper bound is the constructive one from the above discussion and the lower bounds from Section VI-A. Moreover, a comparison of the results presented in this section and some previous results under the m-metric is given next.

In [9], a family of MDS codes is proved to exist, since those achieve the Singleton bound in Claim 9. Similarly to the classical MDS codes for length-n vectors, for the family of array codes to fit large n (and arbitrary L), where n is the number of rows in the array, it is required to use a large field size, i.e., $q \ge n$, where q is the size of the field. Nevertheless, restricting to the binary case, one can construct a code for $n =$

Table I
Bounds on the Redundancy of TE-Correcting Codes.

Array Length	Minimum ρ_{TE}-distance	Constructive Upper Bound	Lower Bound	Gap
$n\times 1$	2	1	1	0
$n\times 2$	3	$\lceil\log_2(n+1)\rceil$	$\lceil\log_2(n+1)\rceil$	0
$n\times 3$	4	$\lceil\log_2(n+1)\rceil+1$	$\lceil\log_2(n+1)\rceil$	$\lesssim 1$
$n\times 4$	5	$\min\{2\lceil\log_2(n+5)\rceil+1, 2\lceil\log_2(2n+1)\rceil\}$	$2\lceil\log_2(n+1)\rceil-1$	$\lesssim 4$
$n\times 2t$	$2t+1$	$t\lceil\log_2(nt+1)\rceil$	$t\lceil\log_2(n)\rceil$	$\lesssim t\log_2(t)$
$n\times(2t+2)$	$2t+2$	$t\lceil\log_2(nt+2)\rceil+1$	$t\lceil\log_2(n)\rceil+1$	$\lesssim t\log_2(t)$

2, which is a $[2\times L, 2L-d+1, d]_{\text{TE}}$ code, where d is arbitrary, and which uses only $d-1$ bits of redundancy. The construction in [9] also solves the case where $e > L$, still only for $n = 2$ in the binary case, as will be discussed in Section III-D. Contrary to that, the first construction in [11], while also be an MDS code for $n = 2$, do come with a restriction of $e \leq L$. However, the authors give a more explicit construction of an MDS code for the $2 \times L$ binary array case. As mentioned in Remark 1, Construction 1 is degenerated in the $n = 2$ case and therefore one cannot compare between the two constructions.

Another important family of codes, introduced in [8], is the construction of BCH codes for the m-metric. In this paper, a construction of Reed-Solomon codes for the m-metric is given, using Galois-Fourier transform and Hasse derivatives, and then a BCH code is derived using alternant code [27, Section 5.5]. However, there are no specific bounds mentioned, and the authors only provide a specific construction example for the case where $n = 3$ and $L = 4$. Therefore, a comparison between their construction and this paper is parameter-dependent. As for their construction for 3×4 binary arrays [8, Table I], one can see that for $d \leq 5$ ($e \leq L$), a comparison can be done using Claims 4 and 5. In our construction, we use at most $1, 2, 3, 6$ bits of redundancy for $d = 2, 3, 4, 5$, respectively, while their optimal choice of parameters yields at most $1, 3, 4, 6$ bits of redundancy for $d = 2, 3, 4, 5$, respectively, therefore our construction improves some of their result, or achieves the same result for the other cases. Other types of basic general construction ideas in [8] are the vector construction, that we mentioned before, and the *Cartesian product construction*, which will be both discussed more in Section III-D, but in their paper the authors claim that these two constructions are inferior compared to their BCH codes. In [11], the authors gave mostly constructions for larger fields, but two constructions are relevant for the binary case. The first was mentioned before, as an MDS code for $n = 2$, and the second will be discussed next in Section III-D.

D. e-TE Codes Where $e > L$

Claim 3 holds only when $e \leq L$. As of the writing of this paper, it is still an open problem whether Construction 1 can be extended for the case where $e > L$, or not. The challenge becomes perceptible when one just examines the scenario in which $e = L + 1$. In this case, the multiset $\mathcal{J}(\boldsymbol{\mathcal{H}}, \boldsymbol{p})$ for $\boldsymbol{p}$ such that $\|\boldsymbol{p}\|_1 = L + 1$, can have an entire row of $\boldsymbol{\mathcal{H}}$ and also one other entry from the last column. Since we cannot know which entry it will be, every entry in the first column should differ from any other entry in $\boldsymbol{\mathcal{H}}$, which results in an immediate decrease in terms of the optimality of the code. However, there exist other constructions, where the assumption on e is not required, and those will be discussed next.

First, the vector construction, where the $n \times L$ array is treated as an length-nL vector. Therefore, the classical theory of erasure-correcting codes satisfies the model and no requirements such as $e \leq L$ exists, but the construction does not use the special structure of the position of the erasures. Next, we describe the Cartesian product construction as in [8]. The main idea here is to use independent erasure-correcting codes for every column. That is, the j-th column, $j \in [L]$, is a codeword in an $(n, M_j, \lceil d/(L-j+1)\rceil)$, where d represents the merged code being a $(d-1)$-TE code. Also in this code, as in the vector construction, the correctness of the code does not rely on the condition of $e \leq L$. Moreover, the entire BCH codes that were constructed in [8] can correct erasures without restrictions on the number of erasures. Also, the construction from [11, Section 3.5] can correct at most r erasures at the end of every row, which is nr erasures in total, but with a restriction on the number of erasures in each row. The other condition for this code to exist, is $L \geq n \geq r$.

Next, we propose a construction of an e-TE code, in which the restriction that $e \leq L$ is removed. The results we achieved for $L = 2, 3, 4$ and $e = 2, 3, 4, 5$, using this construction are summarized in Table II. The bold entries represent cases in which Construction 2 is at least as good as the known state of the art results, in terms of the number of redundancy bits.

This construction is inspired by ideas from [7], [8], and is based on the *Hasse derivative*, as defined in [28] and is presented next. For any non-negative integer i, the i-th Hasse derivative of a polynomial $f(x) = \sum_{k=0}^{d} a_k x^k \in \mathbb{F}_q[x]$ is $f^{(i)}(x) \triangleq \sum_{k=0}^{d} \binom{k}{i} a_k x^{k-i}$, where $\binom{k}{i} = 0$ for all $k < i$. Moreover, a fundamental property of the Hasse derivative [7, Lemma 6] is that for every non-zero polynomial $f(x) \in \mathbb{F}_q[x]$, an element $\beta \in \mathbb{F}$ (where $\mathbb{F}$ is $\mathbb{F}_q$ or any extension field of $\mathbb{F}_q$), is a root of multiplicity ℓ if and only if $f^{(s)}(x) = 0$ for $0 \leq s < \ell$ and $f^{(\ell)}(x) \neq 0$. The following is a construction which is based on the Hasse derivative.

Construction 2. Let n, L, e be arbitrary positive integers, and let m be the smallest integer such that $q = 2^m > n$, that is, $m = \lceil\log_2(n+1)\rceil$. Let $\alpha \in \mathbb{F}_q$ be a primitive element. Then, the TE parity check matrix of the code $\mathcal{C}$ follows,

$$\boldsymbol{\mathcal{H}} = \begin{bmatrix} \boldsymbol{h}_{1,1} & \cdots & \boldsymbol{h}_{1,j} & \cdots & \boldsymbol{h}_{1,L} \\ \boldsymbol{h}_{2,1} & \cdots & \boldsymbol{h}_{2,j} & \cdots & \boldsymbol{h}_{2,L} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \boldsymbol{h}_{n,1} & \cdots & \boldsymbol{h}_{n,j} & \cdots & \boldsymbol{h}_{n,L} \end{bmatrix},$$

where, for $\beta_i=\alpha^i$, each entry is the vector

$$\boldsymbol{h}_{i,j}=\begin{bmatrix}\binom{0}{L-j}\beta_i^{0-L+j}\\ \binom{1}{L-j}\beta_i^{1-L+j}\\ \vdots\\ \binom{e-1}{L-j}\beta_i^{e-1-L+j}\end{bmatrix}\in\mathbb{F}_q^e\ ,i\in[n],j\in[L]\ .$$

Remark 2. The key observation here is that for each vector $\boldsymbol{h}_{i,j}$, it holds that for any vector $f=(f_0,f_1,\dots,f_{e-1})\in\mathbb{F}_q^e$, which represents a polynomial $f(x)=\sum_{\ell=0}^{e-1}f_\ell x^\ell\in\mathbb{F}_q[x]$, the inner product $f\cdot\boldsymbol{h}_{i,j}$ is the $(L-j)$-th Hasse derivative of f, evaluated at β_i. Specifically, if there exists an index j, such that $f\cdot\boldsymbol{h}_{i,\ell}=0$ for $\ell\in\{L,L-1,\dots,j\}$ and $f\cdot\boldsymbol{h}_{i,j-1}\neq 0$, then β_i is a root of multiplicity $L-j+1$ of f.

The following claim proves that Construction 2 is an e-TE code. Although the proof follows from the same ideas as in [7] we include it for completeness.

Claim 6. Let $\mathcal{C}$ be the resulting code from Construction 2. Then, $\rho_{\text{TE}}(\mathcal{C})=e+1$.

Proof: By using Claim 2, it needs to be proved that any multiset $\mathcal{J}(\boldsymbol{\mathcal{H}},\boldsymbol{p})$ of cardinality e is a linearly independent multiset, and therefore let $\mathcal{J}$ be an arbitrary multiset as such. Next, let $A\in\mathbb{F}_q^{e\times e}$ be the matrix that is obtained from placing all the vectors in $\mathcal{J}$ as columns of A. We want to show that A has full rank. Therefore, assume, for the sake of contradiction, that there exists a nonzero vector $f\in\mathbb{F}_q^e$ such that $f\cdot A=0$. Then, using the observation in Remark 2, the polynomial $f(x)$ must have at least e roots (including multiplicities), but it is of degree $e-1$, which is a contradiction.

■

Remark 3. We note the similarity between our approach and the Universally Decodable Matrices (UDM) construction introduced by [7]. In both cases, the properties of the Hasse derivative are applied to ensure the existence of a full-rank matrix, as demonstrated in Claim 6. However, while [7] applied this technique for encoding purposes and with different parameters, we have adapted it for our specific context. To maintain clarity and consistency within our paper, we presented Construction 2 using our established notation. Furthermore, as will be discussed in the rest of this subsection, we derived interesting results on the redundancy of Construction 2 for certain cases, an aspect not addressed in [7].

In order to analyze the number of redundancy bits that Construction 2 requires, one needs to expand the entries in every $\boldsymbol{h}_{i,j}$ into binary columns of length m, and then to compute the rank of the binary expansion of $\boldsymbol{\mathcal{H}}$. This procedure is similar to the one done in analysis of alternant codes. Therefore, the actual number of redundancy bits depends on the specific parameters.

However, we provide next an example of a proof technique, for achieving a better upper bound for specific cases. The technique includes modifying the construction slightly by adding an additional row of parity. This will be demonstrated on a 5-TE code of length $n\times 2$ that uses only $2(\lceil\log_2(n)\rceil+1)$ redundancy bits. This is better than the vector construction (by 1 bit, as one can verify), which is the only construction that fits those parameters.

Claim 7. Let n be a positive integer and $q=2^m$, where m is the smallest integer such that $2^m>n$. Let $\alpha\in\mathbb{F}_q$ be a primitive element and $\beta_i=\alpha^i$.

$$\boldsymbol{\mathcal{H}}=\begin{bmatrix}\boldsymbol{h}_{1,1} & \boldsymbol{h}_{1,2}\\ \boldsymbol{h}_{2,1} & \boldsymbol{h}_{2,2}\\ \vdots & \vdots\\ \boldsymbol{h}_{n,1} & \boldsymbol{h}_{n,2}\end{bmatrix}\ ,$$

where

$$\boldsymbol{h}_{i,1}=\begin{bmatrix}1\\0\\1\\\beta_i^2\end{bmatrix}\ ,\quad \boldsymbol{h}_{i,2}=\begin{bmatrix}0\\1\\\beta_i\\\beta_i^3\end{bmatrix}\ .$$

Then, $\boldsymbol{\mathcal{H}}$ is a TE parity check matrix of a 5-TE code.

Proof: First, let

$$\boldsymbol{\mathcal{H}}'=\begin{bmatrix}\boldsymbol{h}'_{1,1} & \boldsymbol{h}'_{1,2}\\ \boldsymbol{h}'_{2,1} & \boldsymbol{h}'_{2,2}\\ \vdots & \vdots\\ \boldsymbol{h}'_{n,1} & \boldsymbol{h}'_{n,2}\end{bmatrix}\ ,$$

where

$$\boldsymbol{h}'_{i,1}=\begin{bmatrix}1\\0\\1\\0\\\beta_i^2\\0\end{bmatrix}\ ,\ \boldsymbol{h}'_{i,2}=\begin{bmatrix}0\\1\\\beta_i\\\beta_i^2\\\beta_i^3\\\beta_i^4\end{bmatrix}\ .$$

Notice that an entry is added in the beginning of each $\boldsymbol{h}'_{i,j}$, although from Claim 6 one can verify that $\boldsymbol{\mathcal{H}}'$ is a 5-TE code. The way to prove this, is by showing that the removed rows (4-th and 6-th) in every $\boldsymbol{h}'_{i,j}$ can be recovered from the retained rows. Let $\boldsymbol{p}$ be a pattern of erasures and $\mathcal{J}(\boldsymbol{\mathcal{H}}',\boldsymbol{p})$ be the respective multiset of columns from $\boldsymbol{\mathcal{H}}'$. Suppose that $\mathcal{I}=\{(i_1,j_1),(i_2,j_2),\dots,(i_5,j_5)\}$ denotes the indices of those columns in $\mathcal{J}(\boldsymbol{\mathcal{H}}',\boldsymbol{p})$. Also, let $\mathcal{I}_2=\{i\colon(i,2)\in\mathcal{I}\}$, that is all the indices that indicate erasures in the second (i.e. last) column, and $\mathcal{I}_1=\{i\colon(i,1)\in\mathcal{I}\}$, that is, the indices of erasures in the first (i.e., penultimate) column.

Next, denote

$$R_\ell=\sum_{(i,j)\in\mathcal{I}}c_{i,j}(\boldsymbol{h}'_{i,j})_\ell=\sum_{i\in\mathcal{I}_1}c_{i,1}(\boldsymbol{h}'_{i,1})_\ell+\sum_{i\in\mathcal{I}_2}c_{i,2}(\boldsymbol{h}'_{i,2})_\ell\ ,$$

where $c_{i,j}\in\mathbb{F}_2$, and notice that $\mathcal{I}_1$ and $\mathcal{I}_2$ are not disjoint, which is desired.

Next, we prove that $R_4 = (R_3)^2 + R_1$, so that R_4 can be removed from the parity check matrix, using the characteristic of $\mathbb{F}_q$, which is 2, as follows,

$$\begin{aligned} R_4 &= \sum_{i\in\mathcal{I}_1} c_{i,1}(\boldsymbol{h}'_{i,1})_4 + \sum_{i\in\mathcal{I}_2} c_{i,2}(\boldsymbol{h}'_{i,2})_4 \\ &= \sum_{i\in\mathcal{I}_2} c_{i,2}\beta_i^2 \\ &= \sum_{i\in\mathcal{I}_2} c_{i,2}\beta_i^2 + \sum_{i\in\mathcal{I}_1} c_{i,1}\cdot 1 + \sum_{i\in\mathcal{I}_1} c_{i,1}\cdot 1 \\ &= \left(\sum_{i\in\mathcal{I}_2} c_{i,2}\beta_i + \sum_{i\in\mathcal{I}_1} c_{i,1}\cdot 1\right)^2 + \sum_{i\in\mathcal{I}_1} c_{i,1}\cdot 1 \\ &= (R_3)^2 + R_1 \ . \end{aligned}$$

Noting that also $R_6 = (R_4)^2$ completes the proof, since by removing R_4, R_6, the matrix $\boldsymbol{\mathcal{H}}'$ becomes the desired $\boldsymbol{\mathcal{H}}$. ■

The same ideas from the proof of the previous claim can be applied to a few additional parameter regimes, which are highlighted in Table II, that achieve the best known results.

Table II
UPPER BOUNDS ON THE REDUNDANCY BITS OF CONSTRUCTION 2.

	$L = 2$	$L = 3, 4$
$e = 2$	$\mathbf{\log_2(n) + 1}$	$\mathbf{\log_2(n) + 1}$
$e = 3$	$\mathbf{\log_2(n) + 2}$	$\log_2(n) + 3$
$e = 4, 5$	$\mathbf{2\log_2(n) + 2}$	$\mathbf{2\log_2(n) + 3}$

IV. (t, s)-DELETION-CORRECTING CODES

In the process of constructing optimal (t, s)-DC codes, we first introduce a construction of a $(t, 1)$-DC code, and an explicit encoder for this construction will be presented afterwards. Then, a generalization for an encoder of a (t, s)-DC code will conclude this Section. In Section VI-B, it is shown that, in some cases, those constructions are optimal.

A. $(t, 1)$-*DC Codes*

In order to correct a single deletion in a length-n vector, the Varshamov-Tenengolts (VT) codes [29] are the common solution and will also form the basis of the following construction. For $0 \le a \le L$, denote $\mathrm{VT}_a(L)$, as defined in [29],

$$\mathrm{VT}_a(L) = \left\{ \boldsymbol{x} \in \{0,1\}^L \colon \sum_{j=1}^{L} j x_j \equiv a \pmod{L+1} \right\} \ .$$

Next, we construct a code based on the tensor product code concept introduced in [30]. To facilitate this construction, we will first define a variant of VT codes. Denote $h = \lceil \log_2(L+1) \rceil$, such that $\mathbb{F}_{2^h}$, the field of 2^h elements, is the smallest extension field of $\mathbb{F}_2$ that has at least $L+1$ elements. For each row $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,L})$, define the *q-syndrome* to be $s_q(\boldsymbol{x}_i) \equiv \sum_{j=1}^{L} j x_{i,j} \pmod{q}$. Let $\phi : \mathbb{Z}_{2^h} \to \mathbb{F}_{2^h}$ be a bijection and denote $\phi(s_{2^h}(\boldsymbol{x}_i))$ as $\sigma(\boldsymbol{x}_i)$, which is the syndrome of $\boldsymbol{x}_i$ as an element of the field $\mathbb{F}_{2^h}$.

A definition for a variation of the binary VT codes of length L follows,

$$\mathrm{VT}_a^{2^h}(L) = \left\{ \boldsymbol{x} \in \mathbb{F}_2^L \colon \sum_{j=1}^{L} j x_j \equiv a \pmod{2^h} \right\} \ ,$$

where now it holds that $0 \le a \le 2^h - 1$. One can verify that $\mathrm{VT}_a^{2^h}(L)$ can correct a single deletion using the same decoding algorithm as the original VT codes [31]. The following is an implicit construction of a $(t, 1)$-DC code.

Construction 3. Let $\mathcal{C}$ be a code of length n, that can correct t erasures over $\mathbb{F}_{2^h}$. Then,

$$\begin{aligned} \mathcal{C}_{\mathrm{DC}}(n, L, t, \mathcal{C}) \coloneqq \{ &\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{F}_2^{n\times L} \colon \\ &(\sigma(\boldsymbol{x}_1), \ldots, \sigma(\boldsymbol{x}_n)) \in \mathcal{C} \} \ . \end{aligned}$$

As discussed before, Construction 3 yields a tensor product code that relies on the syndromes of the variant of the VT code, such that the possible values of the syndrome can be treated as field elements, and therefore the n syndromes comprise the symbols from a codeword of the non-binary code $\mathcal{C}$ over the same field.

Remark 4. Note that increasing the parameter h results in codes with larger redundancy. One can choose to work with the original VT code, and then construct the non-binary code over the ring $\mathbb{Z}_{L+1}$, instead of a field, if there exists an optimal code over this ring with the required parameters.

Next, we prove that Construction 3 is a $(t, 1)$-DC code.

Lemma 1. The code $\mathcal{C}_{\mathrm{DC}}(n, L, t, \mathcal{C})$ is a $(t, 1)$-DC code.

Proof: The proof is based on the idea that due to the deletions, each erroneous row is shorter, and therefore the location (index) of these rows is known. Assume that $s \le t$ rows suffer from one deletion each and denote its indices as $i_1, \ldots, i_s$. Since $n - s$ rows are error-free, their syndromes can be computed and mapped into $\sigma(\boldsymbol{x}_j)$, where $j \in [n] \setminus \{i_1, \ldots, i_s\}$. Now $\mathcal{C}$ can recover $\sigma(\boldsymbol{x}_{i_1}), \ldots, \sigma(\boldsymbol{x}_{i_s})$ such that $(\sigma(\boldsymbol{x}_1), \ldots, \sigma(\boldsymbol{x}_n))$ can be used truthfully as syndromes for each of the $\mathrm{VT}^{2^h}_{\sigma(\boldsymbol{x}_i)}(L)$ codewords, which is the i-th row. Therefore, the rows $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_s}$ can be recovered. ■

Before we present an explicit encoder for Construction 3, we give an upper bound on the minimal number of redundancy bits that are required to construct a $(t, 1)$-DC code, based on Construction 3.

Theorem 3. Let R_{Opt} be the number of redundancy symbols of $\mathcal{C}_{\mathrm{Opt}}$, a linear code of length n, that can corrects t erasures over $\mathbb{F}_{2^h}$. Then, there exists a code $\mathcal{C}_{\mathrm{DC}}(n, L, t, \mathcal{C}_{\mathrm{Opt}})$ with a redundancy of at most $R_{\mathrm{Opt}} \cdot h = R_{\mathrm{Opt}} \cdot \lceil \log_2(L+1) \rceil$ bits.

Proof: Note that any coset of $\mathcal{C}_{\mathrm{Opt}}$ is a distinct t-erasures-correcting code, and there are $q^{R_{\mathrm{Opt}}}$ cosets, where $q = 2^h$. Thus, there are $q^{R_{\mathrm{Opt}}}$ distinct $\mathcal{C}_{\mathrm{DC}}(n, L, t, \cdot)$ codes, one for each coset, and all those distinct $\mathcal{C}_{\mathrm{DC}}(n, L, t, \cdot)$ codes create a partition of the space of all the 2^{nL} binary arrays of size $n \times L$. Therefore, using the pigeonhole principle, there exists at least one code with $2^{nL}/q^{R_{\mathrm{Opt}}} = 2^{nL - R_{\mathrm{Opt}}\cdot\lceil \log_2(L+1)\rceil}$ codewords. ■

Note that it implies the existence of a $(1,1)$-DC code with only $\log_2(L+1)$ bits of redundancy. Also, if $\mathcal{C}_{\text{Opt}}$ is an MDS code, the upper bound on the redundancy is exactly $t\lceil\log_2(L+1)\rceil$ bits, and by looking at the discussion after Theorem 6, one can verify optimality, assuming the VT codes are optimal 1-deletion-correcting codes.

Corollary 2. There exists a $\mathcal{C}_{\text{DC}}(n,L,t,\mathcal{C}_{\text{Opt}})$ code with a redundancy of at most $t(\log_2(n)+\lceil\log_2(L+1)\rceil)$ bits.

Proof: By using alternant code over $\mathbb{F}_{2^h}$ as in [27, Section 5.5], one has a code $\mathcal{C}$ with a redundancy of at most $t\cdot m$ symbols, where m is an integer such that $2^m \geq (2^h)^n$.

Therefore, an optimal choosing of m is $\lceil\log_{2^h}(n)\rceil$. Finally, using the same proof as in Theorem 3, where $R \leq t\cdot\lceil\log_{2^h}(n)\rceil$ and thus there exists a code with a redundancy of at most

$$R\cdot h \leq t\cdot\lceil\log_{2^h}(n)\rceil\cdot h = t\cdot\left\lceil\frac{\log_2(n)}{h}\right\rceil\cdot h \leq t(\log_2(n)+h) .$$

■

Remark 5. Note that if $n = c\cdot 2^h$ for some positive integer c, then the bound in Corollary 2 is $t(\log_2(n))$.

Finally, an explicit encoder of Construction 3 is presented. In [32], a systematic encoder of $\text{VT}_a(L)$, where the redundancy bits are in the $\{2^i\}_{i=0}^{h-1}$ coordinates was introduced. Then, an encoder for $\text{VT}_a^{2^h}(L)$, denoted as $\mathcal{E}_{\text{VT}}^{(L,a)}$, can be obtained using the same idea, since one can verify that it requires exactly the same entries, $\{2^i\}_{i=0}^{h-1}$, for the encoding process. $\mathcal{E}_{\text{VT}}^{(L,a)}$ is defined as follows. First, it receives a binary input vector $\boldsymbol{d}$ of length $L-h$, then outputs a codeword $\boldsymbol{x}$ of length L, such that $\mathcal{E}_{\text{VT}}^{(L,a)}(\boldsymbol{d}) = \boldsymbol{x} \in \text{VT}_a^{2^h}(L)$.

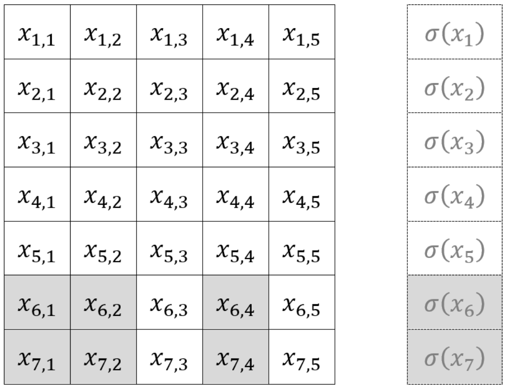
Encoder 1. Let $\text{E}\colon (\mathbb{F}_{2^h})^{n-R} \to (\mathbb{F}_{2^h})^{n}$ be a systematic encoder of a linear code $\mathcal{C}$ of length n, that can correct t erasures over $\mathbb{F}_{2^h}$. Lastly, denote $K = nL - Rh$.

Input: $\boldsymbol{D} = (d_1,\ldots,d_K) \in \mathbb{F}_2^K$.

Output: $\boldsymbol{X} = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n) \in \mathbb{F}_2^{n\times L}$.

1) For $i\in[n-R], j\in[L]\colon x_{i,j} \leftarrow d_{(i-1)L+j}$.
2) Compute $\{\sigma(\boldsymbol{x}_1),\ldots,\sigma(\boldsymbol{x}_{n-R})\}$. Then compute $\text{E}(\sigma(\boldsymbol{x}_1),\ldots,\sigma(\boldsymbol{x}_{n-R}))$ and denote the output as $(\sigma(\boldsymbol{x}_1),\ldots,\sigma(\boldsymbol{x}_n))$.
3) Take the remaining $R(L-h)$ bits of $\boldsymbol{D}$ and split them into R vectors of $L-h$ bits, denoted by $\{\boldsymbol{d}_1,\ldots,\boldsymbol{d}_R\}$.
4) For $i\in[R]\colon a_{n-R+i} \leftarrow \phi^{-1}(\sigma(\boldsymbol{x}_{n-R+i}))$, that is recovering from $\sigma(\cdot)$ the expected syndrome value of the $(n-R+i)$-th row, which is denoted as a_{n-R+i}.
5) For $i\in[R]\colon \boldsymbol{x}_{n-R+i} \leftarrow \mathcal{E}_{\text{VT}}^{(L,a_{n-R+i})}(\boldsymbol{d}_i)$, i.e., encode the VT codeword $\boldsymbol{x}_{n-R+i}$ from $\boldsymbol{d}_i$ as defined in step 3, based on the given syndrome value a_{n-R+i}.
6) Return $\boldsymbol{X}$.

The decoding steps are straightforward, using the decoders of $\mathcal{C}$ and $\text{VT}_a^{2^h}(L)$ in the same way as in the proof of Lemma 1, which proves the correctness, since one can readily verify that the output of Encoder 1 is a codeword of $\mathcal{C}_{\text{DC}}(n,L,t)$ from Construction 3. Moreover, as one can verify, Encoder 1 uses exactly $R_{\text{Opt}}\cdot\lceil\log_2(L+1)\rceil$ bits of redundancy, which is an explicit construction for the existance proof in Theorem 3.



$x_{1,1}$	$x_{1,2}$	$x_{1,3}$	$x_{1,4}$	$x_{1,5}$	$\sigma(x_1)$
$x_{2,1}$	$x_{2,2}$	$x_{2,3}$	$x_{2,4}$	$x_{2,5}$	$\sigma(x_2)$
$x_{3,1}$	$x_{3,2}$	$x_{3,3}$	$x_{3,4}$	$x_{3,5}$	$\sigma(x_3)$
$x_{4,1}$	$x_{4,2}$	$x_{4,3}$	$x_{4,4}$	$x_{4,5}$	$\sigma(x_4)$
$x_{5,1}$	$x_{5,2}$	$x_{5,3}$	$x_{5,4}$	$x_{5,5}$	$\sigma(x_5)$
$x_{6,1}$	$x_{6,2}$	$x_{6,3}$	$x_{6,4}$	$x_{6,5}$	$\sigma(x_6)$
$x_{7,1}$	$x_{7,2}$	$x_{7,3}$	$x_{7,4}$	$x_{7,5}$	$\sigma(x_7)$

Figure 2. Example of an output of Encoder 1 for a $(2,1)$-DC code over binary arrays of length 7×5. Illustration of Example 4.

Finally, we give an example of Encoder 1, for constructing a $(2,1)$-DC code over the space of 7×5 binary arrays.

Example 4. In Figure 2, observe an example of an output of Encoder 1. It receives 29 bits as an input, denoted as $\boldsymbol{D} = (x_{1,1},\ldots,x_{5,5},x_{6,3},x_{6,5},x_{7,3},x_{7,5}) \in \mathbb{F}_2^{29}$ and are shown with the white background in the 7×5 array codeword. The encoder computes the syndromes $(\sigma(\boldsymbol{x}_1),\ldots,\sigma(\boldsymbol{x}_5))$ and then, using an encoder of a $[7,5,3]$ code over $\mathbb{F}_{2^3}$, computes also $\sigma(\boldsymbol{x}_6)$ and $\sigma(\boldsymbol{x}_7)$. The syndromes are presented with dashed lines on the right side of the codeword. Lastly, using the systematic encoder for the VT codes, it computes the redundancy values in the last 2 rows, marked with gray background. Therefore, it is an example of size (7×5) binary array code with 6 bits of redundancy.

B. Construction of a (t,s)-DC Code

This section is based on [33]. First, denote by $\mathcal{C}_s$ the authors' s-deletion-correcting code over the vector space $\mathbb{F}_2^{L+r}$, which is composed from all the codewords $(\boldsymbol{c}, g(\boldsymbol{c}))$, where $\boldsymbol{c}\in\mathbb{F}_2^L$, and $g(\boldsymbol{c})$[1] is used by the authors as an s-deletion-correcting hash for $\boldsymbol{c}$ of size $r = 4s\log_2(L)+o(\log_2(L))$. The existence of such a code is proved in [33]. Finally, denote by $\Gamma(\boldsymbol{c})$ the mapping of $g(\boldsymbol{c})$ to an element of $\mathbb{F}_{2^r}$.

Construction 4. Let n, L, t and s be positive integers, and let r be the constant described previously. Next, let $\mathcal{C}$ be a code of length n, that can correct t erasures over $\mathbb{F}_{2^r}$. Then,

$$\mathcal{C}_{s\text{-DC}}(n,L,t,\mathcal{C}) \coloneqq \{\boldsymbol{X} = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)\in\mathbb{F}_2^{n\times L}\colon \\ (\Gamma(\boldsymbol{x}_1),\ldots,\Gamma(\boldsymbol{x}_n))\in\mathcal{C}\} .$$

The next lemma follows a proof outline similar to that of Lemma 1.

Lemma 2. The code $\mathcal{C}_{s\text{-DC}}(n,L,t,\mathcal{C})$ is a (t,s)-DC code.

Proof: As in Lemma 1, due to the deletions, each erroneous row is shorter, and therefore the index of these rows is known. Assume that $w\leq t$ rows suffer from one deletion each and denote its indices as $i_1,\ldots,i_w$. Since $n-w$ rows

[1] For shortening, we use g instead of the modular version in [33], which is denoted by g_c

Table III
BOUNDS ON THE REDUNDANCY OF (t, s)-DC CODES, WHERE $h = \lceil \log_2(L+1) \rceil$ AND $c > 0$ IS AN INTEGER.

Model	Restriction	Constructive Upper Bound	Asymptotic Lower Bound
$(t,1)$	$n \leq 2^h + 1$	th	$t\lceil \log_2(L) \rceil$
$(t,1)$	$n = c \cdot 2^h$	$t \log_2(n)$	$\max\{th, \lfloor t/2 \rfloor \log_2(n)\}$
$(t,1)$	-	$t(\log_2(n) + h)$	$\max\{th, \lfloor t/2 \rfloor \log_2(n)\}$
(t,s)	-	$t(\log_2(n) + 4s \log_2(L) + o(\log_2(L))$	$\lfloor t/2 \rfloor (\log_2(n) + \lfloor s/2 \rfloor \log_2(L))$

are error-free, the value of their g function can be computed and mapped into $\Gamma(\boldsymbol{x}_j)$, where $j \in [n] \setminus \{i_1, \ldots, i_w\}$. Now $\mathcal{C}$ can recover $\Gamma(\boldsymbol{x}_{i_1}), \ldots, \Gamma(\boldsymbol{x}_{i_s})$ such that $(\Gamma(\boldsymbol{x}_1), \ldots, \Gamma(\boldsymbol{x}_n))$ can be mapped back to valid values of $(g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n))$, and then the rows $\boldsymbol{x}_{i_1}, \ldots, \boldsymbol{x}_{i_w}$ can be recovered using the decoder from [33], since each of these rows suffers at most s deletions, and therefore the set of words $\{(\boldsymbol{x}_{i_j}, g(\boldsymbol{x}_{i_j}))\}_{j=1}^{w}$ also, which is the input to the decoder in [33].

■

Moreover, the next theorem is also using similar techniques as in Theorem 3.

Theorem 4. Let R_{Opt} be the number of redundancy symbols of $\mathcal{C}_{\text{Opt}}$, a linear code of length n, that can corrects t erasures over $\mathbb{F}_{2^r}$. Then, there exists a code $\mathcal{C}_{s\text{-DC}}(n, L, t, \mathcal{C})$ with a redundancy of at most $R_{\text{Opt}} \cdot r \approx R_{\text{Opt}} \cdot 4s \log_2(L)$ bits.

Proof: Note that any coset of $\mathcal{C}_{\text{Opt}}$ is a distinct t-erasures-correcting code, and there are $q^{R_{\text{Opt}}}$ cosets, where $q = 2^r$. Thus, there are $q^{R_{\text{Opt}}}$ distinct $\mathcal{C}_{s\text{-DC}}(n, L, t, \cdot)$ codes, one for each coset, and all those distinct $\mathcal{C}_{s\text{-DC}}(n, L, t, \cdot)$ codes create a partition of the space of all the 2^{nL} binary arrays of size $n \times L$. Therefore, using the pigeonhole principle, there exists at least one code with $2^{nL}/q^{R_{\text{Opt}}} = 2^{nL - R_{\text{Opt}} \cdot r}$ codewords.

■

Remark 6. We note that Construction 4 and Theorem 4 provide only an existential proof of (t, s)-DC codes. Since one cannot ensure that Γ is surjective, an explicit encoder of Construction 4 remains a more complex task, which is left future research.

V. $(t, 1, e)$-TED-CORRECTING CODE

In this section we present a method of constructing $(t, 1, e)$-TED-correcting code. This construction and its encoder use a non-binary code that is based upon the following two pieces of information: (i) The VT syndrome for each row, (ii) The value of the last e bits of each row.

As in Section IV-A, let $h = \lceil \log_2(L+1) \rceil$, and for any array $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{F}_2^{n \times L}$ denote by s_i the 2^h-syndrome of the i-th row of $\boldsymbol{X}$ as in Section IV-A, which is to say that $s_i = \sum_{j=1}^{L} j x_{i,j} \pmod{2^h}$. Moreover, denote the last e bits of the i-th row of $\boldsymbol{X}$ by $\boldsymbol{x}_i^{(:e)} \triangleq (\boldsymbol{x}_i)_{[L-e+1:L]}$. Next, for each row of $\boldsymbol{X}$, define the tuple $\left(s_i, \boldsymbol{x}_i^{(:e)}\right)$. Finally, let $q = 2^h \cdot 2^e = 2^{h+e}$ and denote by $\theta_{\boldsymbol{x}_i}$ the mapping of the tuple $\left(s_i, \boldsymbol{x}_i^{(:e)}\right)$ to an element of $\mathbb{F}_q$.

Construction 5. Let $\mathcal{C}$ be an $[n, k, t+e+1]$ code over $\mathbb{F}_q$. Then,

$$\mathcal{C}_{\text{TED}}(n, L, t, e) := \{\boldsymbol{X} \in \mathbb{F}_2^{n \times L} \colon (\theta_{\boldsymbol{x}_1}, \ldots, \theta_{\boldsymbol{x}_n}) \in \mathcal{C}\} \ .$$

Next, is a proof for the correctness of Construction 5.

Lemma 3. The code $\mathcal{C}_{\text{TED}}(n, L, t, e)$ is a $(t, 1, e)$-TED code.

Proof: A codeword $\boldsymbol{X} \in \mathcal{C}_{\text{TED}}(n, L, t, e)$ is received with at most $t + e$ deletions, some of which are arbitrary deletions and some are tail-erasures. Therefore, there are at most $t + e$ rows for which one cannot compute the tuple $\left(s_i, \boldsymbol{x}_i^{(:e)}\right)$, and since $\mathcal{C}$ can correct $t + e$ erasures, the entire codeword $(\theta_{\boldsymbol{x}_1}, \ldots, \theta_{\boldsymbol{x}_n}) \in \mathcal{C}$ can be recovered. Next, one can recover the codeword $\boldsymbol{X}$ row by row. For each row, if it is of length L, it does not suffer from any deletion. If it is of length $L-1$, it can be recovered using the VT syndrome, regardless of the source of the deletion (arbitrary or TE). Finally, if it is of length $L - k$ for some $1 < k \leq e + 1$, then it suffered from at most 1 arbitrary deletion, and the rest are $k - 1$ TEs, or just k TEs. Thus, one can recover the last $k - 1$ bits, as it is saved in the tuple $\left(s_i, \boldsymbol{x}_i^{(:e)}\right)$, and treat the last deleted bit as an arbitrary deletion, regardless of the source.

■

Similarly to Remark 4, one can choose to work with the original VT codes, and then $q = 2^e \cdot (L + 1)$ is not a prime power, and afterwards construct the non-binary code over the ring $\mathbb{Z}_q$, instead of a field. Moreover, as a proof of existence of codes that are built according to Construction 5, an explicit encoder is provided next.

Encoder 2. Let E: $(\mathbb{F}_q)^{n-R} \to (\mathbb{F}_q)^n$ be a systematic encoder of an optimal linear code $\mathcal{C}$ of length n, that can correct $t + e$ erasures over $\mathbb{F}_q$. Also, let $\mathcal{E}_{\text{VT}}^{(L,a)}$ be a systematic VT code encoder, as defined in Section IV. Lastly, denote $K_{\text{TED}} = nL - R(e + h)$.
Input: $\boldsymbol{D}_{\text{TED}} = (d_1, \ldots, d_{K_{\text{TED}}}) \in \mathbb{F}_2^{K_{\text{TED}}}$.
Output: $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{F}_2^{n \times L}$.

1) For $i \in [n - R], j \in [L] \colon x_{i,j} \leftarrow d_{(i-1)L+j}$, and denote $\boldsymbol{x}_i^{(:e)} = (x_{i,L-e+1}, \ldots, x_{i,L})$.
2) Compute $\{s_1, \ldots, s_{n-R}\}$, and then $\{\theta_{\boldsymbol{x}_1}, \ldots, \theta_{\boldsymbol{x}_{n-R}}\}$.
3) Compute $\text{E}\left(\theta_{\boldsymbol{x}_1}, \ldots, \theta_{\boldsymbol{x}_{n-R}}\right)$ and denote the output as $(\theta_{\boldsymbol{x}_1}, \ldots, \theta_{\boldsymbol{x}_n})$.
4) Take the remaining $R(L - e - h)$ bits of $\boldsymbol{D}$ and split them into R vectors of length $L - e - h$, denoted by $\{\boldsymbol{d}_1, \ldots, \boldsymbol{d}_R\}$.
5) For $i \in \{n - R + 1, \ldots, n\}$, extract the tuple $\left(s_i, \boldsymbol{x}_i^{(:e)}\right)$, computed as $\theta_{\boldsymbol{x}_i}$ in Step 3.

6) Let $\widetilde{\boldsymbol{d}}_i$ be the appending of $\boldsymbol{x}_i^{(:e)}$ to the end of $\boldsymbol{d}_i$.

7) Compute $\boldsymbol{x}_i = \mathcal{E}_{\text{VT}}^{(L,s_i)}(\widetilde{\boldsymbol{d}}_i)$, i.e. encode the VT codeword $\boldsymbol{x}_i$ from $\widetilde{\boldsymbol{d}}_i$, such that the last e bits of $\boldsymbol{x}_i$ are $\boldsymbol{x}_i^{(:e)}$ and the syndrome value of $\boldsymbol{x}_i$ is s_i.

8) Return $\boldsymbol{X}$.

Note that Encoder 2 requires that $e < (L+1) - 2^{h-1}$. Also, it can be verified that Encoder 2 outputs a codeword of $\mathcal{C}_{\text{TED}}(n, L, t, e)$ from Construction 5.

Corollary 3. The following are results regarding the redundancy of $\mathcal{C}_{\text{TED}}(n, L, t, e)$, denoted as R_{TED}.

1) The redundancy of the code $\mathcal{C}_{\text{TED}}(n, L, t, e)$, using Encoder 2 is $nL - K_{\text{TED}} = R(e+h)$.

2) Using an alternant code as $\mathcal{C}$ in Encoder 2 implies

$$R_{\text{TED}} \leq (e+h)\left(\frac{t+e}{2}\right)\log_2(n) \ .$$

3) If $n \leq 2^{h+e}$, using an MDS code code as $\mathcal{C}$ in Encoder 2 implies

$$R_{\text{TED}} \leq (e + \lceil \log_2(L+1) \rceil)\,(t+e) \ .$$

4) If $n \leq 2^{h+1}$, using an MDS code code as $\mathcal{C}$ in Encoder 2 implies a $(1,1,1)$-TED code with $2 + \lceil 2\log_2(L+1) \rceil$ bits of redundancy.

Note that the forth result in Corollary 3 is asymptotically optimal, when compared to the result in Theorem 10. Finally, note that one can use a $(t+e)$-deletion-correcting code over vectors of length nL as in [33], with a redundancy of

$$4(t+e)(\log_2(nL)) + o(\log(nL)) \ ,$$

which in cases where $n > 2^{h+e}$, can achieve better results in terms of redundancy than Encoder 2.

VI. Upper Bounds on the Code Cardinalities

In this section we consider upper bounds on the maximal cardinality of TE codes, (t,s)-DC codes, and $(1,1,1)$-TED codes. These bounds are then compared against the constructions presented in previous sections.

A. Upper Bounds on the Size of Tail-Erasure Codes

Let $\mathcal{A}_{\text{TE}}(n,d)$ denote the maximal cardinality of a $(d-1)$-TE code over the set of $n \times (d-1)$ binary arrays. Let $A(n,d)$ denote the maximal cardinality of a binary vector code of length n with minimum Hamming distance d. The next observation follows, by noting that if the first $n-1$ columns of two codewords in a $(d-1)$-TE code are correspondingly equal, their last column must have Hamming distance at least d.

Claim 8. $\mathcal{A}_{\text{TE}}(n,d) \leq A(n,d) \cdot 2^{n(d-2)}$.

Proof: To prove the result, suppose $\mathcal{C}_{TE} \subseteq \mathbb{F}_2^{n\times(d-1)}$ is an $(d-1)$-TE code of maximal size. Next, decompose the codewords in $\mathcal{C}_{TE}$ into two parts. Let $\mathcal{X} \subseteq \mathbb{F}_2^{n\times(d-2)}$ be equal to the set of arrays that result by removing the last column from each of the codeword arrays from $\mathcal{C}_{TE}$. For any $\mathbf{Y} \in \mathcal{X}$, let $S_{\mathbf{Y}} = \left\{ \boldsymbol{v} \in \mathbb{F}_2^n : \begin{bmatrix} \mathbf{Y} & \boldsymbol{v}^T \end{bmatrix} \in \mathcal{C}_{TE} \right\}$. In other words, $S_{\mathbf{Y}}$ is the set of vectors that can be appended to $\mathbf{Y}$ as the last column to generate a codeword. Since $\mathcal{C}_{TE}$ can correct $d-1$ TEs, that can all occur in the last column, it follows that the minimum Hamming distance of the set of vectors in $S_{\mathbf{Y}}$ is at least d, which implies that for any $\mathbf{Y} \in \mathcal{X}$, we have $|S_{\mathbf{Y}}| \leq A(n,d)$. One can verify that $|\mathcal{X}| \leq 2^{n(d-2)}$, and it follows that,

$$|\mathcal{C}_{TE}| = \sum_{\mathbf{Y}\in\mathcal{X}} |S_{\mathbf{Y}}| \leq \sum_{\mathbf{Y}\in\mathcal{X}} A(n,d) \leq A(n,d) \cdot 2^{n(d-2)},$$

which completes the proof. ■

Using the notations of Section III-C, the above claim provides a simple lower bound on $R_{\text{TE}}(n,d)$, which is

$$R_{\text{TE}}(n,d) \geq R(n,d) \geq \left\lceil \frac{d-1}{2} \right\rceil \log_2(n),$$

where the second inequality is derived from the sphere packing bound for vector codes of length n.

Next, a sphere packing bound is presented, which is tighter, but it is also harder to compute for larger values of minimum ρ_{TE}-distance of a TE code.

Lemma 4. The volume of a ball of radius r in $\mathbb{F}_2^{n\times L}$, under the ρ_{TE}-metric, denoted as $V_{\text{TE}}(r)$, is given by

$$V_{\text{TE}}(r) = \sum_{k=0}^{r} \sum_{\substack{t_1+2t_2+\ldots \\ +k't_{k'}=k}} \left(\binom{n}{t_0, t_1, t_2, \ldots, t_{k'}} \cdot \prod_{i=1}^{k} 2^{(i-1)t_i} \right),$$

where $k' = \min\{k, L\}$.

Lemma 4 is the binary version of the one in [9, Proposition 1] and in [12, Lemma 3], and since it was proved there, we omit the proof. A simplified expression for the case where $e \leq L$ is given next.

Lemma 5. The volume of a ball of radius r in $\mathbb{F}_2^{n\times L}$, where $r \leq L$ is,

$$V_{\text{TE}}(r) = 1 + \sum_{k=1}^{r}\sum_{i=1}^{k} \binom{n}{i}\binom{k-1}{i-1} 2^{k-i} \ .$$

And $V_{\text{TE}}(0)$ is defined to be equal to 1.

Proof: Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$ and $k \leq n$ be the number of TEs that occurred in $\boldsymbol{X}$. Moreover, let $i \leq k$ be the number of rows that suffer from TEs in $\boldsymbol{X}$. Thus, there are $\binom{n}{i}$ possible combinations of rows that suffer from those k TEs, and for each such possible combination of rows, there are $\binom{k-1}{i-1}$ ways to distribute k erasures into i rows. Also, for each erroneous row, the first (leftmost) erased bit is known (must differ from $\boldsymbol{X}$), but the other bits afterwards and until the end of the row, can be either 1 or 0. To conclude, for each k and i, there are $\binom{n}{i}\binom{k-1}{i-1}2^{k-i}$ possible distinct $\boldsymbol{Y} \in \mathbb{F}_2^{n\times L}$ such that $\rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) = k$ and there are exactly i distinct rows between $\boldsymbol{X}$ and $\boldsymbol{Y}$. Then, we complete the proof, using the fact that the number of arrays $\boldsymbol{Y} \in \mathbb{F}_2^{n\times L}$ such that $\rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) = k$ is

$$\sum_{i=1}^{k} \binom{n}{i}\binom{k-1}{i-1} 2^{k-i} \ .$$

■

Equipped with the previous lemma, we can obtain the following sphere packing bound.

Theorem 5. Let $\mathcal{C}$ be a $(n \times L, M, d)_{\mathrm{TE}}$ code. Then,

$$M \cdot V_{\mathrm{TE}}\left(\lfloor (d-1)/2 \rfloor\right) \leq 2^{n \cdot L} \ .$$

Next are some examples for the size of the above volume.

Example 5. The size of $V_{\mathrm{TE}}(r)$, for $r = 1, 2, 3$:

$$V_{\mathrm{TE}}(1) = 1 + n,$$
$$V_{\mathrm{TE}}(2) = 1 + \frac{n^2 + 5n}{2},$$
$$V_{\mathrm{TE}}(3) = 1 + \frac{n^3 + 12n^2 + 29n}{6}.$$

The results above are used to derive the bound from Theorem 5 for $d = 3, 5, 7$, and the authors achieve a tighter bound than the one in Claim 8, as can be verified easily.

In Table I, one can find a summary of Corollary 1, Claims 4 and 5, Claim 8 and Theorem 5.

Finally, we note for completeness that in [9], a version of the Singleton bound was derived for the m-metric, although it is mostly relevant to cases of larger alphabets. The result is as follows.

Claim 9. Let $\mathcal{C}$ be a $(n \times L, M, d)_{\mathrm{TE}}$ code. Then,

$$d \leq nL - \lceil \log_2(M) \rceil + 1 \ .$$

B. Upper Bounds on the Cardinality of (t, s)-DC Codes

In the following subsection, three upper bounds on the cardinality of (t, s)-DC codes are derived. First, we give an overview of the results, in the order it will be proved afterwards through this section. Moreover, a summary of the bounds for (t, s)-DC codes is in Table III. This includes both the constructive upper bounds from Section IV and the lower bounds that are derived next.

Let $\mathcal{A}_{\mathrm{DC}}(t, s)$ be the largest cardinality of a (t, s)-DC code of dimension $n \times L$. Furthermore, denote the largest cardinality of an s-deletion-correcting code of vectors of length L over $\mathbb{F}_2$ by $M_s(L)$.

Theorem 6. The following are bounds on $\mathcal{A}_{\mathrm{DC}}(t, s)$.

1) For any positive integers s, t,
$$\mathcal{A}_{\mathrm{DC}}(t, s) \leq (M_s(L))^t \cdot 2^{L(n-t)} \ .$$
2) For integers $t, s \geq 2$,
$$\mathcal{A}_{\mathrm{DC}}(t, s) \leq \frac{2^{nL}}{\left(\left(\frac{n}{\lfloor t/2 \rfloor}\right) \cdot \left(\frac{L}{\lfloor s/2 \rfloor}\right)^{\lfloor s/2 \rfloor}\right)^{\lfloor t/2 \rfloor}} \ .$$
3) For an integer $t \geq 2$,
$$\mathcal{A}_{\mathrm{DC}}(t, 1) \leq \frac{2^{nL} + 1}{\left(\frac{n}{\lfloor t/2 \rfloor}\right)^{\lfloor t/2 \rfloor}} \ .$$

The bounds in Theorem 6 imply lower bounds on the number of redundancy bits, as summarized next.

Corollary 4. The asymptotic lower bounds on the redundancy of a (t, s)-DC code are as follows.

1) $t(L - \log_2(M_s(L))$.
2) $\lfloor t/2 \rfloor (\log_2(n) + \lfloor s/2 \rfloor \log_2(L)) + O_{s,t}(1)$.
3) $\lfloor t/2 \rfloor \log_2(n) + O_{s,t}(1)$, for $s = 1$.

Note that by observing the case of $s = 1$, and comparing to the results in Section IV-A, one can verify the following. First, in [34] a bound is presented, such that $M_1(L) \leq \frac{2^L - 2}{L - 1}$, and therefore a lower bound on the redundancy of a $(t, 1)$-DC code is at least $t \log_2(L - 1)$ bits, which is attained (up to one bit) in Construction 3, using an MDS code, when $n \leq 2^h + 1$. Secondly, when $n = c \cdot 2^h$ for some positive integer c, the constructive bound from Construction 3 is $t \log_2(n)$, which, compared to the third lower bound in Corollary 4, differ only by a factor of 2.

Next, proofs for all the three bounds will be given, starting with the first one. However, before the proofs, we give the definition of the *Fixed Length Levenshtein* (FLL) distance, as defined in [35], and repeated next. The FLL distance between two binary words of length n, denoted as d_ℓ, is s, if a word $\boldsymbol{x} \in \mathbb{F}_2^n$ can be obtained from a word $\boldsymbol{y} \in \mathbb{F}_2^n$ using s deletions and s insertions (where s is minimal).

Proof of Theorem 6, Part 1: We want to show that $\mathcal{A}_{\mathrm{DC}}(t, s) \leq (M_s(L))^t \cdot 2^{L(n-t)}$. Let $\mathcal{A}'_{\mathrm{DC}}(t, s)$ denote the largest cardinality of a (t, s)-DC code, where the deletions can only occur in the first t rows. Since every (t, s)-DC code can also, in particular, correct (t, s)-DC errors when they occur specifically in the first t rows, $\mathcal{A}_{\mathrm{DC}}(t, s) \leq \mathcal{A}'_{\mathrm{DC}}(t, s)$, and thus providing an upper bound for $\mathcal{A}'_{\mathrm{DC}}(t, s)$ is also an upper bound for $\mathcal{A}_{\mathrm{DC}}(t, s)$.

Next, let $\mathcal{C}_{\mathrm{DC}}$ be a (t, s)-DC code, where the deletions can only occur in the first t rows, and of cardinality $\mathcal{A}'_{\mathrm{DC}}(t, s)$. We partition $\mathcal{C}_{\mathrm{DC}}$ into at most $2^{L(n-t)}$ subcodes, such that any two codewords arrays, $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{C}_{\mathrm{DC}}$, that belong to the same subcode, have their last $n - t$ rows identical. But, since $\boldsymbol{X}, \boldsymbol{Y}$ belong to a code that corrects (t, s)-DC pattern in the first t rows, $\boldsymbol{X}_{[1:t]}$ and $\boldsymbol{Y}_{[1:t]}$, the subarrays of $\boldsymbol{X}$ and $\boldsymbol{Y}$ that are obtained only from their first t rows, belong to a (t, s)-DC code over $\mathbb{F}_2^{t \times L}$. Thus, the cardinality of each subcode is at most $D(t, s, L)$, where $D(t, s, L)$ is the largest cardinality of a (t, s)-DC code over $\mathbb{F}_2^{t \times L}$, and there are $2^{L(n-t)}$ subcodes, which imply $\mathcal{A}_{\mathrm{DC}}(t, s) \leq \mathcal{A}'_{\mathrm{DC}}(t, s) \leq D(t, s, L) \cdot 2^{L(n-t)}$, and we wish to prove that $D(t, s, L) \leq (M_s(L))^t$.

We provide a proof by induction on t. First, for $t = 1$, the code is over vectors of length L, and therefore the inequality $D(1, s, L) \leq M_s(L)$ is given by definition. Next, we prove that if $D(i, s, L) \leq (M_s(L))^i$, then $D(i+1, s, L) \leq (M_s(L))^{i+1}$. Let $\mathcal{C}$ be an $(i+1, s)$-DC code over $\mathbb{F}_2^{(i+1) \times L}$ of cardinality $D(i+1, s, L)$, and let $\boldsymbol{X}^1$ denote the set of all possible rows of the first row of any codeword array in $\mathcal{C}$. If $|\boldsymbol{X}^1| \leq M_s(L)$, then for every $\boldsymbol{x}^1 \in \boldsymbol{X}^1$, denote by $\mathcal{C}(\boldsymbol{x}^1) \subset \mathcal{C}$, the subcode of $\mathcal{C}$, in which the first row of every codeword array in $\mathcal{C}(\boldsymbol{x}^1)$ is $\boldsymbol{x}^1$. Note that $\mathcal{C}(\boldsymbol{x}^1)$ can correct s deletions in each of its $i + 1$ rows, as a subcode of $\mathcal{C}$, and since for every codeword in $\mathcal{C}(\boldsymbol{x}^1)$, its first row remains the same, it is necessary that the last i rows of every codeword in

$\mathcal{C}(\boldsymbol{x}^1)$ be an (i,s)-DC correcting code, or else there exists a pattern of deletions that one cannot recover from.

Therefore, $|\mathcal{C}(\boldsymbol{x}^1)|\leq D(i,s,L)$, which by the induction step, yields $|\mathcal{C}(\boldsymbol{x}^1)|\leq (M_s(L))^i$, and since $|\boldsymbol{X}^1|\leq M_s(L)$, we get that $D(i+1,s,L)=|\mathcal{C}|\leq (M_s(L))^{i+1}$.

Next, if $|\boldsymbol{X}^1|> M_s(L)$, we will construct, in a greedy manner, another $(i+1,s)$-DC code, denoted as $\overline{\mathcal{C}}$, where $|\overline{\mathcal{C}}|=|\mathcal{C}|$, and its set of all possible rows of the first row of any codeword array in the code $\overline{\mathcal{C}}$, denoted $\overline{\boldsymbol{X}}^1$, is of cardinality at most $M_s(L)$, which then will finish the proof. First, arbitrarily choose any $\boldsymbol{y}_1\in\overline{\boldsymbol{X}}^1$, and define the set $Y_{\boldsymbol{y}_1}=\left\{\boldsymbol{y}\in\overline{\boldsymbol{X}}^1\colon d_\ell(\boldsymbol{y},\boldsymbol{y}_1)\leq s\right\}$, i.e., all the row vectors in $\overline{\boldsymbol{X}}^1$, that can be reached from $\boldsymbol{y}_1$ after at most s deletions and insertions. Note that the set $Y_{\boldsymbol{y}_1}$ contains $\boldsymbol{y}_1$. Next, we select any vector $\boldsymbol{y}_2\in\overline{\boldsymbol{X}}^1\setminus Y_{\boldsymbol{y}_1}$, and we define the set $Y_{\boldsymbol{y}_2}=\left\{\boldsymbol{y}\in\overline{\boldsymbol{X}}^1\setminus Y_{\boldsymbol{y}_1}\colon d_\ell(\boldsymbol{y},\boldsymbol{y}_2)\leq s\right\}$.

We keep repeating this procedure until we have ν sets $Y_{\boldsymbol{y}_1}$, $Y_{\boldsymbol{y}_2},\ldots,Y_{\boldsymbol{y}_\nu}$ (which are disjoint by design) and where $Y_{\boldsymbol{y}_1}\cup\cdots\cup Y_{\boldsymbol{y}_\nu}=\overline{\boldsymbol{X}}^1$. By design, $\{\boldsymbol{y}_i\}_{i=1}^{\nu}$ is an s-deletion-correcting code and so $\nu\leq M_s(L)$.

Let $\mathcal{C}(\{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_k\})$ be the subcode of $\mathcal{C}$ where the first row of every codeword array in the subcode $\mathcal{C}(\{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_k\})$ belongs to the set $\{\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_k\}$. Then, let $\overline{\mathcal{C}}$ be defined in the following way. For every $i\in[\nu]$, we take all the codewords in $\mathcal{C}(Y_{\boldsymbol{y}_i})$, add those to $\overline{\mathcal{C}}$, and set their first row as $\boldsymbol{y}_i$.

We claim that $|\overline{\mathcal{C}}|=|\mathcal{C}|$ and that $\overline{\mathcal{C}}$ is an $(i+1,s)$-DC code.

1) Assume, for the sake of contradiction, that $\overline{\mathcal{C}}$ is not an $(i+1,s)$-DC code. Then, there exists $\overline{\boldsymbol{X}},\overline{\boldsymbol{Y}}\in\overline{\mathcal{C}}$ such that these arrays are confusable after a pattern of $(i+1,s)$ deletions. Specifically, it means that their first rows, $\overline{\boldsymbol{x}}_1,\overline{\boldsymbol{y}}_1$ are confusable after s deletions, and thus $\overline{\boldsymbol{x}}_1=\overline{\boldsymbol{y}}_1$ as we chose the first row vectors from an s-deletion-correcting code. Hence, in $\mathcal{C}$, there exist $\boldsymbol{X},\boldsymbol{Y}$ such that $\boldsymbol{X}$ and $\overline{\boldsymbol{X}}$ coincide on the last i rows, and the same is true for $\boldsymbol{Y}$ and $\overline{\boldsymbol{Y}}$.
   However, since $\overline{\boldsymbol{x}}_1=\overline{\boldsymbol{y}}_1$, the first row of $\boldsymbol{X}$ and the first row of $\boldsymbol{Y}$ belong to the same set Y_i, which implies that $\boldsymbol{x}_1,\boldsymbol{y}_1$ are confusable after s deletions, and since $\mathcal{C}$ is an $(i+1,s)$-DC code, the last i rows of $\boldsymbol{X}$ and $\boldsymbol{Y}$ cannot be confusable after a pattern of (i,s) deletions, which also implies that $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$ cannot be confusable, as they have the same last i rows. In turn, this means that $\overline{\boldsymbol{X}}$ and $\overline{\boldsymbol{Y}}$ are not confusable after $(i+1,s)$ deletions, and therefore a contradiction.
2) We show that $|\overline{\mathcal{C}}|=|\mathcal{C}|$. Since every codeword of $\overline{\mathcal{C}}$ was constructed from a unique codeword in $\mathcal{C}$, it implies a surjective function from $\mathcal{C}$ to $\overline{\mathcal{C}}$ and thus $|\overline{\mathcal{C}}|\leq|\mathcal{C}|$. Next, suppose, for the sake of contradiction, that $|\overline{\mathcal{C}}|<|\mathcal{C}|$, i.e., the function is not injective.
   Then, we will have two codewords in $\mathcal{C}$ that differ only in their first row (which was the only row that was changed), where one is $\boldsymbol{u}\in Y_i$ and the other is $\boldsymbol{v}\in Y_i$. In this case, a pattern of s-deletions in the first row results two confusable arrays, a contradiction to $\mathcal{C}$ being an $(i+1,s)$-DC code.

To conclude, we constructed $\overline{\mathcal{C}}$, an $(i+1,s)$-DC code, where $|\overline{\mathcal{C}}|=|\mathcal{C}|$, and its set of possible first rows is of cardinality at most $M_s(L)$. Thus, $|\overline{\boldsymbol{X}}^1|\leq M_s(L)$ and $D(i+1,s,L)=|\mathcal{C}|\leq (M_s(L))^{i+1}$.

■

Next, a proof for the second bound is given. Since the second bound uses a sphere packing argument, a definition of a distance function is defined first. To do that, we will use the FLL distance.

Definition 7. Let $\boldsymbol{X},\boldsymbol{Y}\in\mathbb{F}_2^{n\times L}$ and $s\geq 0$ is an integer. Then,

$$d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})=\begin{cases}\infty & \text{if } \exists i\in[n]\colon d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i)>s,\\ |\{i\colon \boldsymbol{x}_i\neq\boldsymbol{y}_i\}| & \text{otherwise.}\end{cases}$$

Remark 7. If $s=0$, then for any $\boldsymbol{X},\boldsymbol{Y}\in\mathbb{F}_2^{n\times L}$,

$$d_{0\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})=\begin{cases}\infty & \text{if } \boldsymbol{X}\neq\boldsymbol{Y}\ ,\\ 0 & \text{otherwise.}\end{cases}$$

The following theorem states the important connection between Definition 7 and (t,s)-DC codes.

Theorem 7. A code $\mathcal{C}_{s\text{-DC}}\subseteq\mathbb{F}_2^{n\times L}$ is a (t,s)-DC code if and only if every $\boldsymbol{X},\boldsymbol{Y}\in\mathcal{C}_{s\text{-DC}}$ satisfy $d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})>t$.

Proof: First, let $\boldsymbol{X},\boldsymbol{Y}\in\mathcal{C}_{s\text{-DC}}$ and assume for the sake of contradiction that $d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})\leq t$. This implies that there are only $w\leq t$ indices of rows, denoted as $\mathcal{I}=\{i_1,\ldots,i_w\}$, such that $\boldsymbol{x}_{i_j}\neq\boldsymbol{y}_{i_j}$ for every $i_j\in\mathcal{I}$, and $\boldsymbol{x}_k=\boldsymbol{y}_k$ for every $k\notin\mathcal{I}$. Moreover, it also implies that $1\leq d_\ell(\boldsymbol{x}_{i_j},\boldsymbol{y}_{i_j})\leq s$ for every $i_j\in\mathcal{I}$. Thus, for every row index $i_j\in\mathcal{I}$, one can delete s bits in $\boldsymbol{x}_{i_j}$ and s bits in $\boldsymbol{y}_{i_j}$ such that the resulted $(L-s)$-length rows are indistinguishable, as was stated in [35]. Since $|\mathcal{I}|\leq t$, there exists a pattern of (t,s) deletions (i.e., choosing t rows and then choose s bits to delete in each of them) on $\boldsymbol{X}$ and another pattern of (t,s) deletions (over the same row indices) on $\boldsymbol{Y}$, such that the resulted erroneous (punctured) arrays are indistinguishable, which is a contradiction for $\boldsymbol{X},\boldsymbol{Y}$ being both codewords of $\mathcal{C}_{s\text{-DC}}$.

Conversely, let $\mathcal{C}_{s\text{-DC}}$ be a code where every $\boldsymbol{X},\boldsymbol{Y}\in\mathcal{C}_{s\text{-DC}}$ satisfy $d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})>t$, and we will show that it is a (t,s)-DC code. Assume, for the sake of contradiction, that there exist two patterns of (t,s) deletions, P_1 and P_2, such that $\boldsymbol{X}$ suffers from deletions according to P_1 and $\boldsymbol{Y}$ suffers from deletions according to P_2, but the resulted erroneous arrays are indistinguishable. Then, both P_1 and P_2 caused deletions in the same $w\leq t$ rows, so we denote the indices of those rows as $\mathcal{I}=\{i_1,\ldots,i_w\}$. First, note that $\boldsymbol{x}_{i_j}=\boldsymbol{y}_{i_j}$ for every $i_j\notin\mathcal{I}$. Also, since every i_j-th row, where $i_j\in\mathcal{I}$, suffer at most s deletions, it implies $d_\ell(\boldsymbol{x}_{i_j},\boldsymbol{y}_{i_j})\leq s$, otherwise the resulted erroneous arrays cannot be indistinguishable. But, this means that $\forall i\in[n]\colon d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i)\leq s$ (either the rows are the same, or they differ by at most s deletions), and there are at most t rows such that $d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i)\geq s$, which is a contradiction, since $d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y})>t$.

■

Following is the definition for the volume of a ball, using the distance function $d_{s\text{-DC}}$.

Definition 8. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$. Then, the volume of an (τ,σ)-ball, centered at $\boldsymbol{X}$, using the distance function $d_{s\text{-DC}}$, is as follows,

$$\mathrm{V}_{\mathrm{DC}}(\tau,\sigma,\boldsymbol{X}) = \{\boldsymbol{Y} \in \mathbb{F}_2^{n\times L} \colon d_{\sigma\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) \leq \tau\} \ .$$

A sphere packing argument follows immediately.

Theorem 8. Let $\mathcal{C}_{s\text{-DC}} = \{\boldsymbol{X}_i\}_{i=1}^{M}$ be a (t,s)-DC code of cardinality M. Then,

$$\sum_{i=1}^{M} |\mathrm{V}_{\mathrm{DC}}(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor, \boldsymbol{X}_i)| \leq 2^{nL} \ .$$

Proof: It is sufficient to prove that the $(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor)$-balls around each codeword, do not intersect. Assume, for the sake of contradiction, that there exists $\boldsymbol{Y} \in \mathbb{F}_2^{n\times L}$, such that $\boldsymbol{Y} \in \mathrm{V}_{\mathrm{DC}}(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor, \boldsymbol{X}_i) \cap \mathrm{V}_{\mathrm{DC}}(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor, \boldsymbol{X}_j)$. Since $d_{\lfloor s/2 \rfloor\text{-DC}}(\boldsymbol{Y},\boldsymbol{X}_i) < \infty$, there are at most $\lfloor t/2 \rfloor$ distinct rows between $\boldsymbol{Y}$ and $\boldsymbol{X}_i$, and their FLL distance is at most $\lfloor s/2 \rfloor$. Also, since $d_{s\text{-DC}}(\boldsymbol{Y},\boldsymbol{X}_j) < \infty$, there are at most $\lfloor t/2 \rfloor$ distinct rows between $\boldsymbol{Y}$ and $\boldsymbol{X}_j$, and their FLL distance is at most $\lfloor s/2 \rfloor$.

Combining those two observations, there are at most $2\lfloor t/2 \rfloor \leq t$ distinct rows between $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, and their FLL distance is at most $2\lfloor s/2 \rfloor \leq s$. But this means that $d_{s\text{-DC}}(\boldsymbol{X}_i,\boldsymbol{X}_j) \leq t$, which is a contradiction to Theorem 7.

■

Since $\mathrm{V}_{\mathrm{DC}}(r_1,r_2,\boldsymbol{X}) = \dot{\bigcup}_{i=0}^{r_1}\{\boldsymbol{Y} \colon d_{r_2\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) = i\}$, the next claim will be a step towards estimating the volume of a ball.

Claim 10. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$. Then, for any positive integer s, $|\{\boldsymbol{Y} \colon d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) = t\}| \geq \binom{n}{t}\binom{L}{s}^t$.

Proof: First, each $\boldsymbol{Y}$ such that $d_{s\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) = t$ is composed of t row indices $\mathcal{I} = \{i_1,\ldots,i_t\}$ where $d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i) \leq s$ and all the other row indices, $j \in [n] \setminus \mathcal{I}$, satisfy $\boldsymbol{x}_i = \boldsymbol{y}_i$. Therefore, one can divide those binary arrays $\boldsymbol{Y}$ into $\binom{n}{t}$ distinct sets, that differ by their indices set $\mathcal{I}$. Next, let $\mathcal{I}$ be one of these sets of indices. Then, the cardinality of $\mathcal{B}_{\mathcal{I}}(\boldsymbol{X}) = \{\boldsymbol{Y} \mid \forall i \in \mathcal{I} \colon d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i) \leq s, \text{ and } \forall i \notin \mathcal{I} \colon \boldsymbol{x}_i = \boldsymbol{y}_i\}$ is $\prod_{i\in\mathcal{I}} |\mathcal{L}_s(\boldsymbol{x}_i)|$, where $\mathcal{L}_s(\boldsymbol{x}_i) = \{\boldsymbol{y}_i \colon d_\ell(\boldsymbol{x}_i,\boldsymbol{y}_i) \leq s\}$. Using [35, Corollary 3], $|\mathcal{L}_s(\boldsymbol{x}_i)| \geq \binom{L}{s}$ and thus $\prod_{i\in\mathcal{I}} |\mathcal{L}_s(\boldsymbol{x}_i)| \geq \binom{L}{s}^t$. Since there are $\binom{n}{t}$ distinct sets, we conclude the proof.

■

Remark 8. It is clear from Remark 7 that for any positive integer t, $|\{\boldsymbol{Y} \colon d_{0\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) = t\}| = 0$, and thus $|\mathrm{V}_{\mathrm{DC}}(r,0,\boldsymbol{X})| = 1$ for any integer r.

Next, a lemma is given, to conclude a lower bound for the volume of a ball.

Lemma 6. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$. Then, for any integer $s \geq 2$.

$$|\mathrm{V}_{\mathrm{DC}}(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor, \boldsymbol{X})| \geq \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{n}{i}\binom{L}{\lfloor s/2 \rfloor}^i \ .$$

Proof: Using Definition 8 and Claim 10,

$$\begin{aligned} |\mathrm{V}_{\mathrm{DC}}(\lfloor t/2 \rfloor, \lfloor s/2 \rfloor, \boldsymbol{X})| &= \sum_{i=0}^{\lfloor t/2 \rfloor} |\{\boldsymbol{Y} \colon d_{(\lfloor s/2 \rfloor)\text{-DC}}(\boldsymbol{X},\boldsymbol{Y}) = i\}| \\ &\geq \sum_{i=0}^{r} \binom{n}{i}\binom{L}{\lfloor s/2 \rfloor}^i \ . \end{aligned}$$

■

Next, follows the proof of the second bound.

Proof of Theorem 6, Part 2: We show that if $\mathcal{C}_{s\text{-DC}}$ is a (t,s)-DC code, $|\mathcal{C}_{s\text{-DC}}| = M$, $s \geq 2$, then,

$$M \cdot \left(\frac{n}{\lfloor t/2 \rfloor} \cdot \left(\frac{L}{\lfloor s/2 \rfloor} \right)^{\lfloor s/2 \rfloor} \right)^{\lfloor t/2 \rfloor} \leq 2^{nL} \ .$$

Using Theorem 8,

$$2^{nL} \geq \sum_{i=1}^{M} \left| \mathrm{V}_{s\text{-DC}}\left(\left\lfloor \frac{t}{2} \right\rfloor, \boldsymbol{X}_i \right) \right| \ .$$

Then, from Lemma 6,

$$\sum_{i=1}^{M} \left| \mathrm{V}_{s\text{-DC}}\left(\left\lfloor \frac{t}{2} \right\rfloor, \boldsymbol{X}_i \right) \right| \geq M \cdot \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{n}{i}\binom{L}{\lfloor s/2 \rfloor}^i \ .$$

Then,

$$M \cdot \sum_{i=0}^{\lfloor t/2 \rfloor} \binom{n}{i}\binom{L}{\lfloor \frac{s}{2} \rfloor}^i \geq M \binom{n}{\lfloor \frac{t}{2} \rfloor}\binom{L}{\lfloor \frac{s}{2} \rfloor}^{\lfloor \frac{t}{2} \rfloor} \ .$$

And to conclude,

$$M \binom{n}{\lfloor \frac{t}{2} \rfloor}\binom{L}{\lfloor \frac{s}{2} \rfloor}^{\lfloor \frac{t}{2} \rfloor} \geq M \left(\frac{n}{\lfloor \frac{t}{2} \rfloor} \cdot \left(\frac{L}{\lfloor \frac{s}{2} \rfloor} \right)^{\lfloor \frac{s}{2} \rfloor} \right)^{\lfloor \frac{t}{2} \rfloor} \ .$$

■

Finally, the third bound is derived. It is needed, in addition to the previous bound, since, as one can observe, the previous bound is not useful for the $(t,1)$-DC case. Firstly, a definition of a simple distance function between vectors is given.

Definition 9. Let $\boldsymbol{x},\boldsymbol{y} \in \mathbb{F}_2^L$. Then,

$$d^1(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} \infty & \text{if } \exists j \in \{2,\ldots,L\} \colon x_j \neq y_j \ , \\ 1 & \text{if } x_1 \neq y_1 \text{ and } \forall j \in \{2,\ldots,L\} \colon x_j = y_j \ , \\ 0 & \text{otherwise} \ . \end{cases}$$

Then, based on the above distance function, a definition of a distance function for arrays is given.

Definition 10. Let $\boldsymbol{X},\boldsymbol{Y} \in \mathbb{F}_2^{n\times L}$. Then,

$$d_{\mathrm{DC}}^1(\boldsymbol{X},\boldsymbol{Y}) = \begin{cases} \infty & \text{if } \exists i \in [n] \colon d^1(\boldsymbol{x}_i,\boldsymbol{y}_i) = \infty \ , \\ |\{i \colon d^1(\boldsymbol{x}_i,\boldsymbol{y}_i) = 1\}| & \text{otherwise} \ . \end{cases}$$

The next theorem demonstrates the relationship between the distance function d_{DC}^1 and $(t,1)$-DC codes.

Theorem 9. Let $\mathcal{C}_{\mathrm{DC}} \subseteq \mathbb{F}_2^{n\times L}$ be a $(t,1)$-DC code. Then, every $\boldsymbol{X},\boldsymbol{Y} \in \mathcal{C}_{\mathrm{DC}}$ satisfy $d_{\mathrm{DC}}^1(\boldsymbol{X},\boldsymbol{Y}) > t$.

Proof: Assume, for the sake of contradiction, that there exists $\boldsymbol{X},\boldsymbol{Y} \in \mathcal{C}_{\mathrm{DC}}$ such that $d_{\mathrm{DC}}^1(\boldsymbol{X},\boldsymbol{Y}) \leq t$. Then, since the distance is finite, $\forall i \in [n]\colon d^1(\boldsymbol{x}_i, \boldsymbol{y}_i) \leq 1$. But since $d^1_{\text{DC}}(\boldsymbol{X}, \boldsymbol{Y}) \leq t$, there are $w \leq t$ row indices, denoted as $\mathcal{I}$, in which $d^1(\boldsymbol{x}_i, \boldsymbol{y}_i) = 1$ for all $i \in \mathcal{I}$. Moreover, all $\forall j \notin \mathcal{I}\colon \boldsymbol{x}_j = \boldsymbol{y}_j$. Thus, a deletion in the first bit of all the w rows, whose indices are in $\mathcal{I}$, which is a valid pattern of a $(t,1)$-DC error, will yield the same punctured arrays from $\boldsymbol{X}$ and $\boldsymbol{Y}$, which is a contradiction for $\mathcal{C}_{\text{DC}}$ being a $(t,1)$-DC code.

■

Following, as in the previous bound, is a sphere packing argument, starting with a definition of a ball.

Definition 11. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$. Then, the volume of a ball of radius r, centered at $\boldsymbol{X}$, using the distance function d^1_{DC}, is $\text{V}^1_{\text{DC}}(r, \boldsymbol{X}) = \{\boldsymbol{Y}\colon d^1_{\text{DC}}(\boldsymbol{X}, \boldsymbol{Y}) \leq r\}$.

Next, we prove the connection between balls in the Hamming distance and balls in the d^1_{DC} distance.

Claim 11. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$. Then, using $\text{V}_2(r,n)$ as defined in [27, Section 4.2]

$$|\text{V}^1_{\text{DC}}(r, \boldsymbol{X})| = \text{V}_2(r,n) \triangleq \sum_{i=0}^{r} \binom{n}{i} \ .$$

Proof: Since any two arrays $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{F}_2^{n\times L}$ for which their d^1_{DC} distance is finite, can differ only by bits in the first column, and then their d^1_{DC} is equal to the Hamming distance of their first column, there is a clear bijection between the volume of balls in the Hamming distance, $\text{V}_2(r,n)$, and the volume of balls in the d^1_{DC} distance function, $|\text{V}^1_{\text{DC}}(r, \boldsymbol{X})|$. ■

Finally, the proof for the third bound follows.

Proof of Theorem 6, Part 3: We show that if $\mathcal{C}_{\text{DC}}$ is a $(t,1)$-DC code of cardinality M, then,

$$M \cdot \text{V}_2(\lfloor t/2 \rfloor, n) \leq 2^{nL} \ .$$

Again, it is sufficient to prove that the balls $\text{V}^1_{\text{DC}}(\lfloor t/2 \rfloor, \cdot)$ around codewords of $\mathcal{C}_{\text{DC}}$ do not intersect. Assume, for the sake of contradiction, that there exist an array $\boldsymbol{Y} \in \text{V}^1_{\text{DC}}(\lfloor t/2 \rfloor, \boldsymbol{X}_1) \cap \text{V}^1_{\text{DC}}(\lfloor t/2 \rfloor, \boldsymbol{X}_2)$, where $\boldsymbol{X}_1, \boldsymbol{X}_2 \in \mathcal{C}_{\text{DC}}$. Thus, $d^1_{\text{DC}}(\boldsymbol{X}_1, \boldsymbol{Y}) < \infty$ and $d^1_{\text{DC}}(\boldsymbol{X}_1, \boldsymbol{Y}) < \infty$, which implies that there are at most $\lfloor t/2 \rfloor$ indices of rows where $\boldsymbol{X}_1$ and $\boldsymbol{Y}$ differ only in the first bit of the row, and the same is true for $\boldsymbol{X}_1$ and $\boldsymbol{Y}$. Therefore, there are at most t rows in which $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ differ in their first bit, and all the other bits in those arrays are the same. So $d^1_{\text{DC}}(\boldsymbol{X}_1, \boldsymbol{X}_1) < t$, thus a contradiction to Theorem 9.

■

C. Upper Bounds on the Size of $(1,1,1)$-TED Codes

In order to investigate whether a code is optimal in terms of redundancy, a few notations are introduced.

Definition 12. Let $\boldsymbol{X} \in \mathbb{F}_2^{n\times L}$ and denote $D_i^{\downarrow\uparrow}(\boldsymbol{X}) \subset \mathbb{F}_2^{n\times L}$ as the set of all possible arrays that can be received after deleting any bit from the i-th row of $\boldsymbol{X}$, and then inserting an arbitrary bit at the end of the same row of the resulting array.

Let $B_{\text{Del}}(\boldsymbol{x}_i) \subset \mathbb{F}_2^{L-1}$ be the 1-deletion error ball of the i-th row of $\boldsymbol{X}$, i.e., the set of vectors that are obtained by removing one bit from $\boldsymbol{x}_i$. Moreover, let $(\boldsymbol{x}_i)' \in B_{\text{Del}}(\boldsymbol{x}_i)$ be the vector that is obtained by one TE, i.e., $(\boldsymbol{x}_i)' = (x_{i,1}, \ldots, x_{i,L-1})$. Therefore, for every $\boldsymbol{Z} \in D_i^{\downarrow\uparrow}(\boldsymbol{X})$, from the way $\boldsymbol{Z}$ is defined, $(\boldsymbol{z}_i)' \in B_{\text{Del}}(\boldsymbol{x}_i)$. That is, $\boldsymbol{Z} \in D_i^{\downarrow\uparrow}(\boldsymbol{X})$ with the last element in the i-th row deleted, is a possible outcome of $\boldsymbol{X}$ after being transmitted through the $(1,1,1)$-TED channel.

Example 6. Let

$$\boldsymbol{X} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ \boldsymbol{Y} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \ .$$

Then,

$$D_1^{\uparrow\downarrow}(\boldsymbol{X}) = \left\{ \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right\}$$

$$D_2^{\uparrow\downarrow}(\boldsymbol{Y}) = \left\{ \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} \right\} .$$

We want to show that $\boldsymbol{X}$ and $\boldsymbol{Y}$ cannot belong to the same $(1,1,1)$-TED code. Indeed, assume, for sake of contradiction, that they do. Let

$$\boldsymbol{Z} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \in D_1^{\uparrow\downarrow}(\boldsymbol{X}) \cap D_2^{\uparrow\downarrow}(\boldsymbol{Y}) \ .$$

Therefore, let the array

$$\widetilde{\boldsymbol{Z}} = \begin{bmatrix} 1 & 1 & \bullet \\ 1 & 1 & \bullet \\ 0 & 0 & 1 \end{bmatrix} ,$$

be an output of the $(1,1,1)$-TED channel, i.e., $\boldsymbol{Z}$ with the first two rows suffer from a TE. Then, one cannot distinguish between the cases of a deletion in the first row of $\boldsymbol{X}$ and a TE in the second, and a TE in the first row of $\boldsymbol{Y}$ and a deletion in the second. Thus, $\boldsymbol{X}$ and $\boldsymbol{Y}$ cannot belong to the same $(1,1,1)$-TED code.

Next, let $\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X}) = \bigcup_{i=1}^{n} D_i^{\uparrow\downarrow}(\boldsymbol{X})$ and the next claim follows, which generalize the previous example.

Claim 12. Let $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$ be a $(1,1,1)$-TED code. Then, for any distinct $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{C}$,

$$\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X}) \cap \mathcal{D}^{\uparrow\downarrow}(\boldsymbol{Y}) = \emptyset \ .$$

Proof: We need to show that for any distinct $\boldsymbol{X}, \boldsymbol{Y} \in \mathcal{C}$, we have $D_{i_1}^{\uparrow\downarrow}(\boldsymbol{X}) \cap D_{i_2}^{\uparrow\downarrow}(\boldsymbol{Y}) = \emptyset$, for any arbitrary row indices i_1, i_2. Assume, for sake of contradiction, that there exists an array $\boldsymbol{Z} \in D_{i_1}^{\uparrow\downarrow}(\boldsymbol{X}) \cap D_{i_2}^{\uparrow\downarrow}(\boldsymbol{Y})$. If $i_1 = i_2$, then it implies that for any other row than the i_1-th row, $\boldsymbol{X}$ and $\boldsymbol{Y}$ coincide, and that $B_{\text{Del}}(\boldsymbol{x}_i) \cap B_{\text{Del}}(\boldsymbol{y}_i) \neq \emptyset$, and therefore there is a deletion in the i_1-th row that cannot be recovered, while any deletion in the other rows cannot be recovered, and therefore the contradiction. Next, assume $i_1 \neq i_2$. This implies again that $\boldsymbol{X}$ and $\boldsymbol{Y}$ coincide on any row other than i_1, i_2.

Moreover, $\boldsymbol{z}_{i_1} = \boldsymbol{y}_{i_1}$ and $\boldsymbol{z}_{i_2} = \boldsymbol{x}_{i_2}$. Also, $(\boldsymbol{z}_{i_1})' \in B_{\text{Del}}(\boldsymbol{x}_{i_1})$ and $(\boldsymbol{z}_{i_2})' \in B_{\text{Del}}(\boldsymbol{y}_{i_2})$. Thus, the array

$$\widetilde{\boldsymbol{Z}} = \begin{bmatrix} z_{1,1} & z_{1,2} & \cdots & z_{1,L-1} & z_{1,L} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{i_1,1} & z_{i_1,2} & \cdots & z_{i_1,L-1} & \bullet \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{i_2,1} & z_{i_2,2} & \cdots & z_{i_2,L-1} & \bullet \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ z_{n,1} & z_{n,2} & \cdots & z_{n,L-1} & z_{n,L} \end{bmatrix}$$

is indeed a valid output of the transmission of either $\boldsymbol{X}$ or $\boldsymbol{Y}$ through the $(1,1,1)$-TED channel, while $\boldsymbol{X}$ suffers from an arbitrary deletion in the i_1-th row and a TE in the i_2-th row, or $\boldsymbol{Y}$ suffers from a TE in the i_1-th row and an arbitrary deletion in the i_2-th row. Thus, a contradiction to the fact that both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are codewords in $\mathcal{C}$.

■

Next, we use Claim 12, to prove the following upper bound on the cardinality of $(1,1,1)$-TED codes.

Theorem 10. Let $\mathcal{A}_{\text{TED}}(n \times L)$ be the cardinality of the maximal size $(1,1,1)$-TED code of $n \times L$ binary arrays. Then,

$$\mathcal{A}_{\text{TED}}(n \times L) \lesssim \frac{2^{nL}}{nL} \ , \quad \text{i.e.,} \quad \lim_{n\to\infty} \frac{\mathcal{A}_{\text{TED}}(n \times L)}{\frac{2^{nL}}{nL}} \leq 1 \ .$$

Proof: First, following the definition in [35] of a *run* in a vector, denoted as $r(\boldsymbol{x}_i),\ \boldsymbol{x}_i \in \mathbb{F}_2^L$, a generalization is presented next. Let $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n) \in \mathbb{F}_2^{n\times L}$, then *the sum of runs* is defined as $R(\boldsymbol{X}) = \sum_{i=1}^n r(\boldsymbol{x}_i)$. It is possible to show that $|D_i^{\uparrow\downarrow}(\boldsymbol{X})| = 2r(\boldsymbol{x}_i)$, and then $\left|\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X})\right| = 2R(\boldsymbol{X})$. Next, assume that $\mathcal{C}$ is a $(1,1,1)$-TED code of size $\mathcal{A}_{\text{TED}}(n \times L)$, and then define the following,

$$\mathcal{C}_0 = \left\{ \boldsymbol{X} \in \mathcal{C} \colon R(\boldsymbol{X}) \geq \frac{nL}{2} - \sqrt{nL\ln(nL)} \right\}$$

And $\mathcal{C}_1 = \mathcal{C} \setminus \mathcal{C}_0$. From Claim 12, every distinct codeword array $\boldsymbol{X} \in \mathcal{C}$ results a distinct $\mathcal{D}(\boldsymbol{X})$, thus

$$\begin{aligned} 2^{nL} &\geq \sum_{\boldsymbol{X}\in\mathcal{C}} |\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X})| \\ &= \sum_{\boldsymbol{X}\in\mathcal{C}_0} |\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X})| + \sum_{\boldsymbol{X}\in\mathcal{C}_1} |\mathcal{D}^{\uparrow\downarrow}(\boldsymbol{X})| \\ &\geq \sum_{\boldsymbol{X}\in\mathcal{C}_0} 2R(\boldsymbol{X}) \\ &\geq |\mathcal{C}_0| \cdot \left(nL - 2\sqrt{nL\ln(nL)} \right) \ , \end{aligned}$$

and therefore,

$$|\mathcal{C}_0| \leq \frac{2^{nL}}{nL - 2\sqrt{nL\ln(nL)}} \approx \frac{2^{nL}}{nL} \ .$$

Where the last step above is due to the fact that

$$\frac{2^{nL}}{nL - 2\sqrt{nL\ln(nL)}} \Big/ \frac{2^{nL}}{nL} \xrightarrow{n\to\infty} = 1 \ .$$

Denoting $N = nL$ and $g(N) = \frac{N}{2} - \sqrt{N\ln(N)}$, we say

$$\begin{aligned} |\mathcal{C}_1| &\leq \sum_{r=n}^{g(N)-1} |\{\boldsymbol{X} \in \{0,1\}^{n\times L} \colon R(\boldsymbol{X}) = r\}| \\ &\leq \sum_{r=1}^{g(N)-1} |\{\boldsymbol{X} \in \{0,1\}^{N} \colon r(\boldsymbol{X}) = r\}| \\ &= \sum_{r=1}^{g(N)-1} 2\binom{N-1}{r-1} \ . \end{aligned}$$

Where $r(\boldsymbol{X})$ is the number of runs in the vectorized array $\boldsymbol{X}$, that is appending every row to the end of its previous row. Moreover, the second inequality is due to the fact that any vectorized array can either have the same number of runs (if every row ends with the same bit as the next row starts with), or have an larger number of runs. Therefore, any array that is in one of the sets on the l.h.s (of the second inequality), has to be in one of the sets on the r.h.s. Now, using the lemma in [31], $|\mathcal{C}_1| = o\left(\frac{2^N}{N^2}\right)$. To conclude, $\frac{|\mathcal{C}_1|}{\frac{2^{nL}}{nL}} \xrightarrow{n\to\infty} 0$ and therefore,

$$|\mathcal{C}_0| + |\mathcal{C}_1| \lesssim \frac{2^{nL}}{nL} \ .$$

■

That means that the redundancy of an $n \times L$ binary array code capable of correcting at most a single deletion and a single TE is asymptotically at least $\log_2(n) + \log_2(L)$ bits.

VII. Conclusions and Future Work

Our research has shown promising results. For TE and DC codes we provided some optimal constructions, using some interesting techniques. The new coding schemes developed in this work show that it is possible to correct many frequently occurring error patterns by manipulating the parity check matrices of existing linear codes, and using ideas based on tensor product codes. With the ever-increasing demand for data storage, our findings contribute to the growing body of research into alternative data storage solutions, offering a new and innovative approach. Further research is necessary to fully assess the potential of this technique, but our results are a step towards a future where DNA storage is a reality.

Acknowledgments

The authors would like to acknowledge Prof. Ron M. Roth and Prof. Yuval Cassuto for their helpful reviews and comments on the paper. We extend special thanks to Prof. Roth for suggesting the expression in Lemma 5 and its proof. We are also grateful to the reviewers and editor for their insightful comments and suggestions, which have improved the quality of this work.

References

[1] "DNA data storage market size is expected to reach usd 1926.7 million by 2028," https://www.prnewswire.com/news-releases/, [Online; accessed January-2023].

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

[3] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized dna," *Nature*, vol. 494, no. 1435, pp. 77–80, 2013.

[4] P. Predki and M. Cassidy, "Systems and methods for writing and reading data stored in a polymer," U.S. Patent 2020/0224264 A1, Jul. 2020.

[5] B. Vasic and E. Kurtas, *Coding and signal processing for magnetic recording systems*. CRC press, 2004.

[6] K. Chen, J. Kong, J. Zhu, N. Ermann, P. Predki, and U. F. Keyser, "Digital data storage using dna nanostructures and solid-state nanopores," *Nano Letters*, vol. 19, no. 2, pp. 1210–1215, 2019. [Online]. Available: https://doi.org/10.1021/acs.nanolett.8b04715

[7] A. Ganesan and P. O. Vontobel, "On the existence of universally decodable matrices," *IEEE Transactions on Information Theory*, vol. 53, no. 7, pp. 2572–2575, 2007.

[8] W. Zhou, S. Lin, and K. A. Abdel-Ghaffar, "BCH codes for the rosenbloom–tsfasman metric," *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 6757–6767, 2016.

[9] M. Y. Rosenbloom and M. A. Tsfasman, "Codes for the m-metric," *Problemy Peredachi Informatsii*, vol. 33, no. 1, pp. 55–63, 1997.

[10] S. Dougherty and M. Skriganov, "Maximum distance separable codes in the ρ metric over arbitrary alphabets," *Journal of Algebraic Combinatorics*, vol. 16, pp. 71–81, 07 2002.

[11] N. Raviv, M. Schwartz, R. Cohen, and Y. Cassuto, "Hierarchical erasure correction of linear codes," *Finite Fields and Their Applications*, vol. 68, p. 101743, 2020.

[12] S. Jain, "Array codes in m-metric correcting independent and clustered errors simultaneously," *World Applied Sciences Journal*, vol. 22, pp. 36–40, 01 2013.

[13] A. Lenz, P. H. Siegel, A. Wachter-Zeh, and E. Yaakobi, "Anchor-based correction of substitutions in indexed sets," in *IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 757–761.

[14] K. Immink and K. Cai, "Design of capacity-approaching constrained codes for DNA-based storage systems," *IEEE Communication Letters*, vol. 22, no. 2, pp. 224–227, 2018.

[15] S. Yazdi, H. Kiah, R. Gabrys, and O. Milenkovic, "Mutually uncorrelated primers for DNA-based data storage," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6283–6296, 2018.

[16] W. Press, J. Jones, J. Schaub, and I. Finkelstein, "Hedges error-correcting code for DNA storage corrects indels and allows sequence constraints," *Proceedings of the National Academy of Sciences*, vol. 117, no. 31, pp. 18 489–18 496, 2020.

[17] A. Lenz, P. Siegel, and E. Yaakobi, "Coding over sets for DNA storage," *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2331–2351, 2019.

[18] H. Wei and M. Schwartz, "Improved coding over sets for DNA-based data storage," *IEEE Transactions on Information Theory*, vol. 68, no. 1, pp. 118–129, 2022.

[19] A. Boruchovsky, D. Bar-Lev, and E. Yaakobi, "DNA-correcting codes: End-to-end correction in DNA storage systems," in *2023 IEEE International Symposium on Information Theory (ISIT)*, 2023, pp. 579–584.

[20] J. Sima, N. Raviv, and J. Bruck, "On coding over sliced information," *IEEE Transactions on Information Theory*, vol. 67, no. 5, pp. 2793–2807, 2021.

[21] Z. Liu and M. Mitzenmacher, "Codes for deletion and insertion channels with segmented errors," *IEEE Transactions on Information Theory*, vol. 56, no. 1, pp. 224–232, 2009.

[22] M. Abroshan, R. Venkataramanan, and A. Fàbregas, "Coding for segmented edit channels," *IEEE Transactions on Information Theory*, vol. 64, no. 4, pp. 3086–30 982, 2017.

[23] B. Masnick and J. Wolf, "On linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 13, no. 4, p. 600–607, 1967.

[24] Z. Zhou and C. Xu, "An improved unequal error protection turbo codes," in *Proceedings of 2005 International Conference on Wireless Communications, Networking and Mobile Computing*, Piscataway, 2005, pp. 284–287.

[25] I. Boyafunov and G. Katsman, "Linear unequal error protection codes," *IEEE Transactions on Information Theory*, vol. 27, no. 2, pp. 168–175, 1981.

[26] S. Lin and D. J. Costello, *Error Control Coding*, 2nd ed. Pearson, 2004.

[27] R. Roth, *Introduction to Coding Theory*. USA: Cambridge University Press, 2006.

[28] H. Hasse, "Theorie der höheren differentiale in einem algebraischen funktionenkörper mit vollkommenem konstantenkörper bei beliebiger charakteristik." *Journal für die reine und angewandte Mathematik*, vol. 175, pp. 50–54, 1936. [Online]. Available: http://eudml.org/doc/149955

[29] R. Varshamov and G. Tenengolts, "Codes which correct single asymmetric errors (in russian)," *Automatika i Telemkhanika*, vol. 161, no. 3, pp. 288–292, 1965.

[30] J. K. Wolf, "An introduction to tensor product codes and applications to digital storage systems," in *2006 IEEE Information Theory Workshop - ITW '06 Chengdu*, 2006, pp. 6–10.

[31] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals." *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, feb 1966, doklady Akademii Nauk SSSR, V163 No4 845-848 1965.

[32] K. Abdel-Ghaffar and H. Ferreira, "Systematic encoding of the varshamov-tenengol'ts codes and the constantin-rao codes," *IEEE Transactions on Information Theory*, vol. 44, no. 1, pp. 340–345, 1998.

[33] J. Sima, R. Gabrys, and J. Bruck, "Optimal systematic t-deletion correcting codes," in *2020 IEEE International Symposium on Information Theory (ISIT)*, 2020, pp. 769–774.

[34] A. Fazeli, A. Vardy, and E. Yaakobi, "Generalized sphere packing bound," *IEEE Transactions on Information Theory*, vol. 61, no. 5, pp. 2313–2334, 2015.

[35] D. Bar-Lev, T. Etzion, and E. Yaakobi, "On levenshtein balls with radius one," in *2021 IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 1979–1984.

Appendix

Theorem 1. A code $\mathcal{C} \subseteq \mathbb{F}_2^{n\times L}$ is an e-TE-correcting code if and only if $\rho_{\text{TE}}(\mathcal{C}) \geq e+1$.

Proof: Let $\boldsymbol{X} \in \mathcal{C}$. If $\mathcal{C}$ is an e-TE code, it means that there is no other $\boldsymbol{Y} \in \mathcal{C}$ such that $\boldsymbol{X}^{(\boldsymbol{p})} = \boldsymbol{Y}^{(\boldsymbol{p})}$ for any $\boldsymbol{p} \in P(e, L, n)$, where $\|\boldsymbol{p}\|_1 = t$, and $t \leq e$. Assume, for the sake of contradiction, that such $\boldsymbol{p}$ exists. It means that if $\boldsymbol{X}$ suffers from a TE pattern $\boldsymbol{p}$ where $\|\boldsymbol{p}\|_1 = t$, and $t \leq e$, it cannot be distinguished from $\boldsymbol{Y}$ that suffers from the same TE pattern $\boldsymbol{p}$. Therefore, this TE cannot be corrected, which yields a contradiction to $\mathcal{C}$ being a e-TE code. So, conclude that $\rho_{\text{TE}}(\mathcal{C}) > e$. Conversely, if $\rho_{\text{TE}}(\mathcal{C}) \geq e+1$, let $\boldsymbol{X} \in \mathcal{C}$ and assume an e-TE occurred in $\boldsymbol{X}$, and we want to show that the code can correct it. Assume, for the sake of contradiction, that there exists a $\boldsymbol{Y} \in \mathcal{C}$ and $\boldsymbol{p} \in P(e, n, L)$ such that $\boldsymbol{X}^{(\boldsymbol{p})} = \boldsymbol{Y}^{(\boldsymbol{p})}$ and $\|\boldsymbol{p}\|_1 \leq e$, and therefore the code cannot correct this e-TE, since it cannot distinguish between $\boldsymbol{X}$ and $\boldsymbol{Y}$. But then $\rho_{\text{TE}}(\boldsymbol{X}, \boldsymbol{Y}) \leq e$ and this contradicts the minimum ρ_{TE}-distance of the code. We conclude that after at most e-TE occurred, $\boldsymbol{X}$ is the only codeword that could be the result of those e-TEs, and conclude that $\mathcal{C}$ can correct up to e-TEs. ■

Boaz Moav received the B.Sc. degree in computer science and mathematics and the M.Sc. degree from the Computer Science Department, Technion-Israel Institute of Technology, in 2021 and 2024, respectively, where he is currently pursuing the Ph.D. degree with the Henry and Marilyn Taub Faculty of Computer Science. His advisor is Prof. E. Yaakobi. His research interests include coding theory and algorithms for DNA storage systems.

Ryan Gabrys is a scientist jointly affiliated with the Qualcomm Institute and Naval Information Warfare Center San Diego. His research interests broadly lie in the areas of theoretical computer science and electrical engineering, including bioinformatics, combinatorics, coding theory, and signal processing. He is particularly interested in inter-disciplinary problems that span multiple areas such as biology, mathematics, and systems design. Gabrys received a Ph.D. from the University of California, Los Angeles in 2014; a Master of Engineering from the University of California, San Diego in 2010; and a B.S. in Computer Science and Mathematics from the University of Illinois, Urbana-Champaign in 2005.

Eitan Yaakobi (S'07–M'12–SM'17) is an Associate Professor at the Computer Science Department at the Technion — Israel Institute of Technology. He also holds a courtesy appointment in the Technion's Electrical and Computer Engineering (ECE) Department. He received the B.A. degrees in computer science and mathematics, and the M.Sc. degree in computer science from the Technion — Israel Institute of Technology, Haifa, Israel, in 2005 and 2007, respectively, and the Ph.D. degree in electrical engineering from the University of California, San Diego, in 2011. Between 2011-2013, he was a postdoctoral researcher in the Department of Electrical Engineering at the California Institute of Technology and at the Center for Memory and Recording Research at the University of California, San Diego. His research interests include information and coding theory with applications to non-volatile memories, associative memories, DNA storage, data storage and retrieval, and private information retrieval. He received the Marconi Society Young Scholar in 2009 and the Intel Ph.D. Fellowship in 2010-2011. Between 2020 and 2023, he served as an Associate Editor for Coding and Decoding for the IEEE TRANSACTIONS ON INFORMATION THEORY. Since 2016, he has been affiliated with the Center for Memory and Recording Research at the University of California, San Diego, and between 2018–2022, he was affiliated with the Institute of Advanced Studies, Technical University of Munich, where he held a four-year Hans Fischer Fellowship, funded by the German Excellence Initiative and the EU 7th Framework Program. Between August 2023 and January 2024, he was a Visiting Associate Professor at the School of Physical and Mathematical Sciences at Nanyang Technological University. He is a recipient of several grants, including the ERC Consolidator Grant and the EIC Pathfinder Challenge.