

# Non-asymptotic Upper Bounds for Deletion Correcting Codes

Ankur A. Kulkarni      Negar Kiyavash

**Abstract**—Explicit non-asymptotic upper bounds on the sizes of multiple-deletion correcting codes are presented. In particular, the largest single-deletion correcting code for  $q$ -ary alphabet and string length  $n$  is shown to be of size at most  $\frac{q^n - q}{(q-1)(n-1)}$ . An improved bound on the asymptotic rate function is obtained as a corollary. Upper bounds are also derived on sizes of codes for a constrained source that does not necessarily comprise of all strings of a particular length, and this idea is demonstrated by application to sets of run-length limited strings.

The problem of finding the largest deletion correcting code is modeled as a matching problem on a hypergraph. This problem is formulated as an integer linear program. The upper bound is obtained by the construction of a feasible point for the dual of the linear programming relaxation of this integer linear program.

The non-asymptotic bounds derived imply the known asymptotic bounds of Levenshtein and Tenengolts and improve on known non-asymptotic bounds. Numerical results support the conjecture that in the binary case, the Varshamov-Tenengolts codes are the largest single-deletion correcting codes.

**Index Terms**—Deletion channel, multiple-deletion correcting codes, single-deletion correcting codes, non-asymptotic bounds, hypergraphs, integer linear programming, linear programming relaxation, Varshamov-Tenengolts codes.

## I. INTRODUCTION

A *deletion channel* is a communication channel that takes a string of symbols as its input and transmits only a subset of the input symbols leaving the order of the symbols unchanged. Symbols that are not transmitted constitute the errors in the channel and are called *deletions*. A deletion channel is distinct from the widely studied erasure channel wherein the positions of the errors are known. This paper mainly concerns deletion channels where the maximum number of deletions, denoted  $s$ , is fixed.

A *codebook* or a *deletion correcting code* for the deletion channel is a set  $C$  of input strings, no two of which on transmission through the channel can result in the same output. For a string  $x$ , call the set of strings obtained by

deletion of  $s$  symbols from  $x$ , the *s-deletion set* of  $x$ . An *s-deletion correcting code* is thus a set of input strings with pairwise disjoint *s-deletions* sets.

To explain our contribution, consider the case where  $s = 1$  (the *single-deletion* channel). An open problem pertaining to this channel is the determination of the size of the largest or *optimal* codebook  $C = C_n^*$ , for input strings comprising of all strings of length  $n$  [1]. The classical bound of Levenshtein [2] provides one benchmark for optimality. For the case of binary strings, Levenshtein [2] showed that the size  $|C_n^*|$  of an optimal codebook for the single-deletion channel is *asymptotically* at most  $\frac{2^n}{n}$ . It is important to note here the sense in which this asymptoticity is being defined. A function  $f : \mathbb{N} \rightarrow \mathbb{R}$  is said to be asymptotically less than or equal to another function  $g : \mathbb{N} \rightarrow \mathbb{R}$ , written  $f \lesssim g$ , if  $\lim_{n \rightarrow \infty} \frac{f(n)}{g(n)} \leq 1$ .  $f$  is said to be asymptotically equal to  $g$ , written  $f \sim g$ , if  $f \lesssim g$  and  $g \lesssim f$ . Thus Levenshtein's result says that  $\lim_{n \rightarrow \infty} \frac{|C_n^*|}{2^n/n} \leq 1$ . Levenshtein then constructs a codebook of size at least  $\frac{2^n}{n+1}$ , thereby proving  $\frac{2^n}{n} \lesssim |C_n^*|$ , and hence concludes that the optimal codebook  $C_n^*$  has size asymptotically equal to  $\frac{2^n}{n}$ , i.e.  $C_n^*$  satisfies  $\lim_{n \rightarrow \infty} \frac{|C_n^*|}{2^n/n} = 1$ .

If the function  $g$  is bounded, the asymptotic equality  $f \sim g$  implies equality of the limiting values of  $f(n)$  and  $g(n)$  or their near-equality for sufficiently large  $n$ . However since  $g(n) = 2^n/n$  is unbounded, Levenshtein's asymptotic results do not allow one to obtain a fine approximation to  $|C_n^*|$ , or conclude if for a particular  $n$ ,  $|C_n^*|$  is greater or less than  $\frac{2^n}{n}$ , or even conclude the boundedness or unboundedness of the difference  $||C_n^*| - \frac{2^n}{n}|$ . Indeed, the best known codes for the binary version of this channel, the Varshamov-Tenengolts (VT) codes [3], are of size at least  $\frac{2^n}{n+1}$  for input length  $n$ . Although this sequence is asymptotically equal to  $\frac{2^n}{n}$  (and recently verified by exact search to be optimal for string lengths  $n \leq 10$  [4]), the difference  $\frac{2^n}{n} - \frac{2^n}{n+1}$  grows to infinity.

In other words, for this problem, asymptotic optimality of a codebook does not say much about its optimality per se. The challenges noted above continue to hold (and are perhaps more severe) for larger alphabet and larger number of deletions. For the case of multiple deletions, asymptotic bounds exist, thanks to Levenshtein [2] for binary alphabet, but little is known about the quality of these bounds, since

Both authors are at the Coordinated Science Laboratory at the University of Illinois at Urbana-Champaign, Urbana, Illinois, U.S.A., 61801. They can be reached at [akulkar3@illinois.edu](mailto:akulkar3@illinois.edu) and [kiyavash@illinois.edu](mailto:kiyavash@illinois.edu), respectively. This work was supported in part by AFOSR under Grants FA9550-11-1-0016, FA9550-10-1-0573, and NSF grants CCF 10-54937 CAR and CCF 10-65022 Kiyavash.

no matching lower bounds exist. A more useful bound for any such channel would be a *non-asymptotic upper bound* that also implies known asymptotic bounds. Such a bound can serve as a hard bound on the size of a codebook for any string length and help in assessing the quality of specific code constructions. Such non-asymptotic upper bounds are the subject of this paper.

We derive explicit non-asymptotic upper bounds on the sizes of codebooks for any number of deletions  $s$  and any alphabet size  $q$ . These bounds imply the known asymptotic bounds of Levenshtein [2] and generalize them to larger alphabet. For the case of a single deletion we obtain this bound in closed form. We show that for string length  $n$ , an optimal  $q$ -ary single-deletion codebook has size at most  $\frac{q^n - q}{(q-1)(n-1)}$ . This implies the asymptotic upper bound of  $\frac{q^n}{(q-1)n}$  shown by Tenengolts [5]. In the binary case, together with the size of the VT codes (which effectively provide non-asymptotic *lower* bounds), our upper bound  $\frac{2^n - 2}{n-1}$  implies Levenshtein's asymptotic results.

From these bounds we derive an upper bound on the *asymptotic rate function*. For a channel where the number of deletions is a constant fraction of string length, this function gives the asymptotic value of the rate of the largest deletion correcting code, as a function of the fraction of symbols that are deleted. This bound on the rate function improves on the previous bound shown by Levenshtein [6].

We then extend this methodology to derive bounds on deletion correcting codes for *constrained sources*. These are codebooks for a specific set of strings, i.e., not necessarily the set of *all* strings of a particular length. Recording systems such as magnetic tapes impose physical constraints on the patterns that symbols can take in codewords [7]. If such a code is subsequently transmitted through a deletion channel, the codewords can be thought of as a constrained source. As a specific demonstration of this idea, we derive non-asymptotic upper bounds on sizes of codebooks for run-length limited sources for the single-deletion channel.

The bounds are obtained as follows. We characterize the largest codebook for the deletion channel as a maximum *matching* on a suitably defined hypergraph. The problem of finding a maximum matching is written as a 0-1 integer linear program. The *fractional matching* on this hypergraph is the solution of the linear programming relaxation of this integer linear program, and its value is an upper bound on the size of the maximum matching. Our upper bound is obtained by constructing a feasible solution for the *dual* of this linear program. For the single-deletion channel the construction is such that it allows for the calculation of the dual objective in closed form as  $\frac{q^n - q}{(q-1)(n-1)}$ . Unfortunately, for larger number of deletions, due to the complicated nature of the resulting expressions, we are unable to produce closed form expressions.

Computations on a computer reveal that for the binary single-deletion channel the optimal fractional matching size is quite close to the size of the VT codes. For strings of length up to 14, the difference between the size of the VT codes and the optimal fractional matching is at most 8; this indicates that the VT codes are either optimal or very close to being optimal (at least up to string length 14). On a side note, the hypergraph approach also appears to be more amenable to algorithmic approaches due to its compact representation; this aspect of this paper may be of independent interest.

### A. Related work

A wide-ranging survey on various results and challenges associated with deletion correction and its variants was recently presented by Mercier et al. [8]. Sloane's survey [1] deals specifically with the binary single-deletion channel and illuminates several deep open questions pertaining to the VT codes. Here we recall some highlights from this area of work.

The study of the deletion channel has a long history going back at least to the seminal work of Levenshtein [2] wherein asymptotic bounds on the sizes of optimal binary codebooks were derived. For  $s$  deletions and binary input strings, Levenshtein [2] showed that the largest codebook  $\mathcal{C}_{2,s,n}^*$  for string length  $n$  satisfies the asymptotic relations

$$\frac{2^s (s!)^2 2^n}{n^{2s}} \lesssim |\mathcal{C}_{2,s,n}^*| \lesssim \frac{s! 2^n}{n^s}. \quad (1)$$

Levenshtein [2] also noticed that the Varshamov-Tenengolts codes [3], which were proposed for asymmetric error correction, served as asymptotically optimal codes for the binary single-deletion channel; these remain to date the best known codes and have recently been confirmed to be optimal for string length up to 10. An independent line of study on this topic appears to have been contemporaneously pursued by Ullman [9], [10].

Thereafter there have been many efforts at code construction. An attempt at generalizing the VT codes for the binary multiple-deletion channel was made by Helberg and Ferreira [11]; that this generalization indeed corrects deletion errors was recently shown by Abdel-Ghaffar et al. [12]. For non-binary alphabet this problem was first studied by Calabi and Harnett [13] and Tanaka and Kasai [14]. Later Tenengolts proposed a construction similar to the VT codes for the  $q$ -ary single-deletion channel and showed that the optimal codebook for string length  $n$ ,  $\mathcal{C}_{q,1,n}^*$ , is of size at least  $\frac{q^n}{qn}$  and satisfies the asymptotic upper bound  $|\mathcal{C}_{q,1,n}^*| \lesssim \frac{q^n}{(q-1)n}$  [5]. Interestingly, no asymptotic bounds for  $q$ -ary  $s$ -deletion correcting codes appear to have been explicitly articulated, though Levenshtein's original proof from [2] seems extendable to  $q$ -ary strings. The VT

codes are number-theoretic and the underlying number-theoretic logic was generalized to correct larger number of asymmetric errors by Varshamov [15].

Butenko et al. attempted to find codes algorithmically by casting this problem as a maximum independent set problem on a class of graphs [16]. Schulman and Zuckerman considered a construction that is in part algorithmic and showed the existence of ‘asymptotically good’ codes for deletions whose number increases proportionally to the length of the string [17]. More recently, the algorithmic approach has been pursued by Khajouei et al. [18] and a graph coloring based approach was studied by Cullina et al. [19]. Finding codes for the deletion channel, either algorithmically or through a number-theoretic construction, is a considerable challenge, as evidenced by the attempts at achieving the records for largest codebooks on the webpage maintained by Sloane [4].

Deletion errors have also been studied for run-length limited sources – which we consider in this paper as an example of a constrained source – by Roth and Siegel [20], Hilden et al. [21] and Bours [22], amongst others. However in these works, the deletion errors considered have a specific pattern and do not exactly correspond to the deletion channel we consider. Exceptions to this are the recent works of Cheng et al. [23] and Palunčić et al. [24] which consider codes for run-length limited sources for the deletion channel in its full generality.

The topic of deletion errors has spawned research on related questions, such as the existence of ‘perfect codes’ (Levenshtein [25]), and the combinatorial problems of counting subsequences (e.g., Hirschberg and Regnier [26], Swart and Ferreira [27], Mercier et al. [28] and more recently, Liron and Langberg [29]) and the reconstruction of sequences (Levenshtein [30], [31]). Another body of active ongoing research studies the capacity of the deletion channel (e.g., Mitzenmacher [32], Kanoria and Montanari [33], and Diggavi et al. [34]).

The question of non-asymptotic upper bounds, which is our interest, is comparatively less studied. One may scan Levenshtein’s proof of the asymptotic bound from [2] to see if a non-asymptotic bound has been found in it as an intermediate step. For the single deletion channel, the bound so discovered (see Sloane’s proof [1, Theorem 2.5]) is greater than  $\frac{2^n}{n-2\sqrt{n \log n}}$  (for binary alphabet) which is clearly weaker than our bound. In fact, Levenshtein [6] has presented a somewhat more general bound on the size of a  $q$ -ary  $s$ -deletion correcting code:

$$|\mathcal{C}_{q,s,n}^*| \leq \frac{q^{n-s}}{\sum_{i=0}^s \binom{r-s+1}{i}} + q \sum_{i=0}^{r-1} \binom{n-1}{i} (q-1)^i, \quad (2)$$

where  $r$  is any integer satisfying  $1 \leq s \leq r+1 \leq n$ . It is not clear which value of  $r$  provides the strongest bound of these (although a heuristic argument using Stirling’s

approximation suggests that  $r \approx \frac{n}{2}$  should be optimal in the binary single-deletion case; this is essentially Levenshtein’s original argument [2]). We have found via numerical calculation that the strongest of the bounds in (2) is weaker than our bound. Additionally, our bound in the single-deletion case also has the attractiveness of being in closed form. Levenshtein in another paper derives another non-asymptotic bound for the size of a  $q$ -ary single-deletion codebook [25, Theorem 5.1],

$$|\mathcal{C}_{q,1,n}^*| \leq \frac{q^{n-1} + (n-2)q^{n-2} + q}{n}, \quad (3)$$

but this bound is asymptotically much weaker than Tenengolts’ asymptotic bound of  $\frac{q^n}{(q-1)^n}$  (their ratio grows to infinity; our bound implies Tenengolts’ asymptotic bound). Sloane’s website [4] contains several numerical bounds found by calculating the Lovász  $\vartheta$  [35] on certain graphs. But unlike our bounds, there are no expressions (closed form or otherwise) for these bounds.

The scarcity of non-asymptotic upper bounds is perhaps due to the property that deletion sets of distinct strings can have distinct sizes. This point has also been stressed by Sloane [1, Section “Optimality”]: “*It is more difficult to obtain upper bounds for deletion-correcting codes than for conventional error-correcting codes, since the disjoint balls  $D_e(u)$  (deletion sets) associated with the codewords ... do not all have the same size. Furthermore the metric space  $(\mathbb{F}_2^n, d)^1$  is not an association scheme and so there is no obvious linear programming bound.*” In the light of this comment it is interesting that our non-asymptotic bound is obtained from a linear programming argument, and it relies critically on the sizes of the deletion sets.

## B. Organization

This paper is organized as follows. Section II comprises of preliminaries including, notation, problem definition, background on hypergraphs and the derivation of lemmas that are of use in our analysis. Section III contains the hypergraph characterization of the optimal codebook and the derivation of the upper bounds for single-deletion correcting codes. In Section IV we extend the analysis to obtain bounds on codes for larger number of deletions and derive a bound on the asymptotic rate function. In Section V, we derive bounds on codebooks for constrained sources, in particular, for run-length limited sources. Numerical simulations comparing the values of Levenshtein’s bound from (2), our bound, the tightest bound obtainable by our logic, and the best known codes are presented in Section VI. In Section VII we discuss our results and possible avenues for tightening our bound and conclude the paper.

<sup>1</sup> $d$  is the Levenshtein or edit distance, cf. Definition 2.4.

## II. PRELIMINARIES

Let  $\mathbb{F}_q = \{0, 1, \dots, q-1\}$  be a  $q$ -ary alphabet and let  $\mathbb{F}_q^n$  denote the set of all  $q$ -ary sequences of length  $n$ . Any such  $q$ -ary sequence is called a *string*. We let  $\mathbb{F}_q^* = \bigcup_{n=0}^{\infty} \mathbb{F}_q^n$  denote set of all strings; here  $\mathbb{F}_q^0$  denotes the empty string. Let  $x = x_1 \dots x_n$  be a string. A *subsequence* of  $x$  is formed by taking a subset of the symbols of  $x$  and aligning them without altering their order. In other words, a subsequence of  $x$  is a sequence  $y = x_{i_1} \dots x_{i_k}$ , where  $1 \leq k \leq n$  and the indices satisfy  $1 \leq i_1 < \dots < i_k \leq n$ ;  $x$  is called a *supersequence* of  $y$ . We say that  $y$  is obtained from  $x$  by the *deletion* of  $n - k$  symbols and  $x$  is obtained from  $y$  by the *insertion* of  $n - k$  symbols.

A specific type of subsequence that is important for our results is a *run*, defined below.

**Definition 2.1:** Let  $x = x_1 \dots x_n \in \mathbb{F}_q^n$  be a string. A *run* of  $x$  is a maximal contiguous subsequence with identical symbols, i.e. a run of  $x$  is a sequence  $x_i x_{i+1} \dots x_{i+j}$ ,  $1 \leq i \leq i+j \leq n$  with the property that  $x_i = x_{i+1} = \dots = x_{i+j}$  and the properties that, a) if  $1 < i$  then  $x_{i-1} \neq x_i$ , and b) if  $i+j < n$ , then  $x_{i+j} \neq x_{i+j+1}$ . For any  $x \in \mathbb{F}_q^n$ ,  $r(x)$  denotes the number of runs of  $x$ .

For example if  $q = 3$  and  $x = 120010$ , the runs of  $x$  are 1, 2, 00, 1, 0 and  $r(x) = 5$ . Clearly for any  $x \in \mathbb{F}_q^n$ ,  $1 \leq r(x) \leq n$ .

**Definition 2.2:** For any string  $x \in \mathbb{F}_q^*$ , the set of subsequences of  $x$  obtained by deletion of  $s$  symbols is denoted by  $D_s(x)$  and set of supersequences obtained by insertion of  $s$  symbols into  $x$  is denoted by  $I_s(x)$ . We call  $D_s(x)$  and  $I_s(x)$  the *s-deletion set* of  $x$  and *s-insertion set* of  $x$ , respectively.

For example if  $q = 3, s = 1$  and  $x = 120010$ , then  $D_1(x) = \{20010, 10010, 12010, 12000, 12001\}$ . Notice that subsequences obtained by the deletion of a symbol from the same run of  $x$  are all identical. For example, in the run 00, deletion of either 0 results in the same subsequence 12010. Consequently we have the following relation [25],

$$|D_1(x)| = r(x), \quad \forall x \in \mathbb{F}_q^*. \quad (4)$$

For  $s > 1$ , expressions for  $|D_s(x)|$  get increasingly complicated, and depend on statistics of  $x$  other than the number of runs (see, e.g., [28] for one set of expressions). We discuss bounds on  $|D_s(\cdot)|$  later in Section IV.

Surprisingly, the size of  $I_s(x)$  is independent of  $x$ , but is a function only of the length of  $x$  and the size of the alphabet [36, Lemma 1, p. 354]. Specifically, we have

$$|I_s(x)| = \sum_{j=0}^s \binom{n}{j} (q-1)^j \quad \forall x \in \mathbb{F}_q^{n-s}. \quad (5)$$

We denote this quantity by  $\iota_{q,s,n}$ ,

$$\iota_{q,s,n} \triangleq \sum_{j=0}^s \binom{n}{j} (q-1)^j. \quad (6)$$

As a general rule, instead of using ‘1-deletion’ or ‘1-insertion’ (correcting code, set, . . .), we use the more elegant ‘single-deletion’ (correcting code, set, . . .) etc.

The central object of our interest, namely, a deletion correcting code is defined below.

**Definition 2.3:** A *s-deletion correcting code* (or “*s-deletion codebook*”) for string length  $n$  and alphabet  $\mathbb{F}_q$  is a set  $C \subseteq \mathbb{F}_q^n$  with the property that the sets  $D_s(x), x \in C$ , are pairwise disjoint. The largest such code is denoted by  $C_{q,s,n}^*$  and called an *optimal s-deletion correcting code* or *optimal s-deletion codebook*.

A code capable of correcting  $s$  deletions is also capable of correcting a total of  $s$  insertions and deletions [2], whereby an *s-deletion correcting code* is also a *s-insertion correcting code* (i.e., a set  $C \subseteq \mathbb{F}_q^n$  such that the sets  $I_s(x), x \in C$ , are pairwise disjoint) [2]. Another characterization of single-deletion correcting codes is through the Levenshtein distance.

**Definition 2.4:** For any  $x, y \in \mathbb{F}_q^*$  define the Levenshtein distance or edit distance  $d(x, y)$  as minimum number of insertions or deletions required to obtain  $x$  from  $y$ .

A set  $C \subseteq \mathbb{F}_q^n$  is a *s-deletion correcting code* if and only if  $d(x, y) > 2s$  for any two distinct strings  $x, y \in C$ . In summary, we have the following equivalence [2].

**Lemma 2.1:** For any  $x, y \in \mathbb{F}_q^n$ , the following three statements are equivalent.

- 1)  $d(x, y) \leq 2s$ ,
- 2)  $D_s(x) \cap D_s(y) \neq \emptyset$ ,
- 3)  $I_s(x) \cap I_s(y) \neq \emptyset$ .

The following lemma, although not directly related to deletion correction, will be required for our analysis.

**Lemma 2.2:** Let  $n, k, d \in \mathbb{N}, k \leq n, dk \leq n$  and let  $t_1, \dots, t_k$  be variables taking values in  $\mathbb{N}$ . The number of solutions  $(t_1, \dots, t_k)$  to the set of equations

$$\sum_{i=1}^k t_i = n, \quad t_i \geq d, t_i \in \mathbb{N}, \forall 1 \leq i \leq k, \quad (7)$$

is  $\binom{n-k(d-1)-1}{k-1}$ .

**Proof:** First suppose  $d = 1$ . Consider an array of  $n$  1’s and insert  $k - 1$  0’s between the 1’s, so that no two 0’s are inserted next to each other and no 0’s are inserted at the beginning or the end of the array. There is a one-to-one correspondence between an arrangement of this kind and a solution of (7):  $t_i$ , for  $1 < i < k$ , corresponds to the number of 1’s between the  $(i-1)^{\text{th}}$  0 and  $i^{\text{th}}$  0 and  $t_1, t_k$  are the number of 1’s at the beginning and the end of the array. The number of such arrangements is easily seen to be  $\binom{n-1}{k-1}$ .

Now suppose  $d > 1$ . Notice that the system (7) is

equivalent to the system

$$\sum_{i=1}^k (t_i - (d-1)) = n - k(d-1),$$

$$(t_i - (d-1)) \geq 1, t_i - (d-1) \in \mathbb{N}, \forall 1 \leq i \leq n.$$

This system reduces to the earlier case with  $d = 1$ , but with variables  $t'_i = t_i - (d-1)$ , for  $i = 1, \dots, k$ . The number of solutions in this case is  $\binom{n-k(d-1)-1}{k-1}$ . ■

#### A. Background on hypergraphs

The contents of this section are sourced from Berge [37].

A hypergraph is a generalization of the concept of a graph. In a graph edges are pairs of vertices. In a hypergraph, one allows arbitrary nonempty sets of vertices, including those with exactly one element, to be the so-called *hyperedges*. Formally,

*Definition 2.5:* A hypergraph  $\mathcal{H}$  is a tuple  $(X, \mathcal{E})$ , where  $X$  is a finite set and  $\mathcal{E}$  is a collection of nonempty subsets of  $X$  such that  $\bigcup_{E \in \mathcal{E}} E = X$ .  $X$  is called the *vertex set*, its elements are called *vertices* and the elements of  $\mathcal{E}$  are called *hyperedges*.

When a vertex belongs to a hyperedge, we say it is *covered* by the hyperedge. The above definition assumes that the hypergraph contains no exposed vertex, i.e., a vertex that is covered by no hyperedge. This is a matter of convention; other definitions, e.g. [38], do not impose this requirement.

Let  $\mathcal{E} = \{E_1, \dots, E_m\}$  be the set of hyperedges of the hypergraph  $\mathcal{H} = (X, \mathcal{E})$ . For a set of indices  $J \subseteq \{1, \dots, m\}$ , the *partial hypergraph generated by  $J$*  is  $\mathcal{H}_J = (X_J, \{E_j | j \in J\})$ , where  $X_J = \bigcup_{j \in J} E_j$ .

Hyperedges are defined as sets and as such one can talk of intersection of hyperedges. Specifically, two hyperedges are disjoint if there is no vertex that is covered by both hyperedges. The idea of packing neighborhoods or spheres used in coding theory sits naturally in the theory of hypergraphs. A packing of hyperedges is called a *matching*.

*Definition 2.6:* A matching of a hypergraph  $\mathcal{H} = (X, \mathcal{E})$  is a collection of pairwise disjoint hyperedges  $E_1, \dots, E_j \in \mathcal{E}$ . The matching number of  $\mathcal{H}$ , denoted  $\nu(\mathcal{H})$ , is the largest  $j$  for which such a matching exists.

A dual concept (in a sense we make precise below) of a matching is a transversal.

*Definition 2.7:* A transversal of a hypergraph  $\mathcal{H} = (X, \mathcal{E})$  is a subset  $T \subset X$  that intersects every hyperedge in  $\mathcal{E}$ . The transversal number of  $\mathcal{H}$ , denoted  $\tau(\mathcal{H})$ , is the smallest size of a transversal.

Suppose  $\mathcal{H} = (X, \mathcal{E})$  is a hypergraph with  $n$  vertices  $x_1, \dots, x_n$  and  $m$  hyperedges  $E_1, \dots, E_m$ . Consider a matrix  $A \in \{0, 1\}^{n \times m}$ , where the element in the  $i^{\text{th}}$  row

and  $j^{\text{th}}$  column is

$$A[i, j] = \begin{cases} 1 & \text{if } x_i \in E_j, \\ 0 & \text{otherwise.} \end{cases}$$

$A$  is called the incidence matrix of  $\mathcal{H}$ . The matching number and the transversal number are both solutions of integer linear programs. In the rest of this paper, we refer to problem (8) below as the *matching problem* and (9) as the *transversal problem* on hypergraph  $\mathcal{H}$ .

*Lemma 2.3:* The matching number and transversal number are solutions of integer linear programs:

$$\nu(\mathcal{H}) = \max\{\mathbf{1}^\top z \mid Az \leq \mathbf{1}, z_j \in \{0, 1\}, 1 \leq j \leq m\}, \quad (8)$$

$$\tau(\mathcal{H}) = \min\{\mathbf{1}^\top w \mid A^\top w \geq \mathbf{1}, w_i \in \{0, 1\}, 1 \leq i \leq n\}, \quad (9)$$

where  $\mathbf{1}$  denotes a column vector of all 1's of appropriate dimension.

*Proof:* In the integer linear programming formulation of the matching problem, each hyperedge  $E_j \in \mathcal{E}$  corresponds to a variable  $z_j \in \{0, 1\}$  and  $z$  is the vector  $(z_1, \dots, z_m)$ . The variable  $z_j$  is interpreted as the indicator function that identifies if hyperedge  $E_j$  is a part of the matching represented by  $z$ . Thus  $z_j = 1$  if  $E_j$  is selected, and  $z_j = 0$  otherwise. The matching problem has one constraint for each vertex: for a vertex  $x_i$ , the sum of  $z_j$  over those hyperedges  $j$  that cover vertex  $x_i$  is at most 1; hence, at most one of these  $z_j$  takes value 1. Consequently, a vector  $z$  is feasible for the matching problem if and only if the collection  $\{E_j : z_j = 1\}$  is a matching of  $\mathcal{H}$ . It follows that the matching number of  $\mathcal{H}$  is the optimal value of (8).

By a similar construction, in the integer linear programming formulation of the transversal problem, let each vertex  $x_i \in X$  correspond to a variable  $w_i \in \{0, 1\}$  and let  $w = (w_1, \dots, w_n)$ . The variable  $w_i = 1$  if and only if vertex  $x_i$  is included in the transversal represented by  $w$ . The transversal problem has one constraint for each hyperedge which says that for a hyperedge  $E_j$ , the sum of  $w_i$  over those vertices  $i$  that are covered by  $E_j$  is at least 1, whereby at least one of these  $w_i$  takes value 1. There is thus a one-to-one correspondence between a transversal of  $\mathcal{H}$  and a feasible vector  $w$  for (9). The transversal number is thus characterized by (9). ■

Notice that the mathematical programs in (8) and (9) are duals of each other. A fundamental theorem of integer linear programming states that a pair of dual programs satisfy *weak duality*. Weak duality means that of the pair of dual problems, the value of the maximization problem is no greater than the value of the minimization problem [39]. Applied to (8)-(9), this implies, for any hypergraph  $\mathcal{H}$ ,

$$\nu(\mathcal{H}) \leq \tau(\mathcal{H}). \quad (10)$$

We note a technical point about problems (8)-(9) that helps in simplifying our analysis. Notice that the constraint  $z_j \in \{0, 1\}$  in (8) and the constraint  $w_i \in \{0, 1\}$  in (9) may as well be replaced with the constraints  $z_j \in \mathbb{Z}_+$  and  $w_i \in \mathbb{Z}_+$ , respectively, where  $\mathbb{Z}_+$  is the set of nonnegative integers, to give the following equivalent characterizations for  $\nu(\mathcal{H})$  and  $\tau(\mathcal{H})$

$$\nu(\mathcal{H}) = \max\{\mathbf{1}^\top z \mid Az \leq \mathbf{1}, z_j \in \mathbb{Z}_+, 1 \leq j \leq m\}, \quad (11)$$

$$\tau(\mathcal{H}) = \min\{\mathbf{1}^\top w \mid A^\top w \geq \mathbf{1}, w_i \in \mathbb{Z}_+, 1 \leq i \leq n\}. \quad (12)$$

To see the equivalence between (8) and (11), notice that no vector  $z \in \mathbb{Z}_+^m$  satisfying  $Az \leq \mathbf{1}$  can have a component greater than 1. And in (9), observe that no minimizing  $w \in \mathbb{Z}_+^n$  of (12) can have a component greater than 1. From now on, we consider only the formulations (11)-(12). Note that sources such as Berge [37] omit the above analysis and directly employ (11)-(12) to define  $\nu(\mathcal{H})$  and  $\tau(\mathcal{H})$ .

The linear programming relaxation of an integer program is constructed by replacing the requirement that a variable takes only integral values by a requirement that allows the variable to also take any real value between the integral values (i.e., in the convex hull of the integral values) [39]. By  $\nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H})$  we denote the values of the linear programming relaxations of (11) and (12), respectively. i.e.,

$$\nu^*(\mathcal{H}) = \max\{\mathbf{1}^\top z \mid Az \leq \mathbf{1}, z \geq 0\}, \quad (13)$$

$$\tau^*(\mathcal{H}) = \min\{\mathbf{1}^\top w \mid A^\top w \geq \mathbf{1}, w \geq 0\}, \quad (14)$$

where for simplicity, we denote a vector of zeros of appropriate size also by '0'.  $\nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H})$  are called the *fractional matching number* and *fractional transversal number* of  $\mathcal{H}$ . A vector  $z$  feasible for (13) is called a *fractional matching* and the set  $\{z : Az \leq \mathbf{1}, z \geq 0\}$  is called the *fractional matching polytope* of  $\mathcal{H}$ . A vector  $w$  feasible for (14) is called a *fractional transversal* and the set  $\{w : A^\top w \geq \mathbf{1}, w \geq 0\}$  is called the *fractional transversal polytope*.  $\mathbf{1}^\top z$  and  $\mathbf{1}^\top w$  are called the *weights* of  $z$  and  $w$ .  $\nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H})$  being linear programs satisfy the fundamental property of *strong duality* [39], i.e.,

$$\nu^*(\mathcal{H}) = \tau^*(\mathcal{H}).$$

Thus for any hypergraph the fractional matching number and the fractional transversal number are equal. In general, integer programs do not satisfy strong duality and thereby equality may not hold in (10). Equality or lack thereof in (10) depends on the shape of the fractional matching and fractional transversal polytopes. On a side note, we recall that linear programming relaxations have been employed in the decoding of binary linear codes by Feldman et al. [40].

Fractional matchings and transversals do not have as direct a counting interpretation as the vectors feasible for (8)-(9). However they are extremely useful for obtaining

bounds. Since the feasible regions of the integer programs are strictly contained in the feasible regions of their of the linear programming relaxations, we immediately have  $\nu(\mathcal{H}) \leq \nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H}) \leq \tau(\mathcal{H})$ . Furthermore, we have the following lemma.

*Lemma 2.4:* For any hypergraph  $\mathcal{H}$ , we have

$$\nu(\mathcal{H}) \leq \nu^*(\mathcal{H}) = \tau^*(\mathcal{H}) \leq \tau(\mathcal{H}).$$

In particular,

$$\nu(\mathcal{H}) \leq \tau^*(\mathcal{H}) \leq \mathbf{1}^\top w,$$

for any fractional transversal  $w$ .

*Proof:* Since fractional matchings and transversal problems are relaxations of the matching and transversal problem,  $\nu(\mathcal{H}) \leq \nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H}) \leq \tau(\mathcal{H})$ . By the duality theorem of linear programming  $\nu^*(\mathcal{H})$  and  $\tau^*(\mathcal{H})$  are equal. By definition, any fractional transversal  $w$  must have weight no less than the fractional transversal number, by which the last claim follows. ■

We end this survey with one final concept, that of a line graph.

*Definition 2.8:* A line graph of a hypergraph  $\mathcal{H} = (X, \mathcal{E})$  is a graph  $L(\mathcal{H})$  with vertices given by the hyperedges of  $\mathcal{H}$  and two vertices in  $L(\mathcal{H})$  are joined by an edge if they intersect as hyperedges in  $\mathcal{H}$ .

An independent set of a graph is a set of vertices, no two of which share an edge. For a graph  $G$  we denote the size of its largest independent set, or its *independence number*, by  $\alpha(G)$ . Now consider a hypergraph  $\mathcal{H}$ . An independent set of its line graph  $L(\mathcal{H})$  corresponds to a collection of hyperedges of  $\mathcal{H}$  that are pairwise disjoint. Consequently,

$$\nu(\mathcal{H}) = \alpha(L(\mathcal{H})), \quad (15)$$

i.e., the matching number of a hypergraph equals the independence number of its line graph.

### III. NON-ASYMPTOTIC UPPER BOUNDS FOR SINGLE-DELETION CORRECTING CODES

#### A. Hypergraph characterization

The contents of this subsection apply to any  $s$  number of deletions. We will specialize to single-deletions and present our bounds in the following subsection.

Consider the following hypergraphs.

$$\begin{aligned} \mathcal{H}_{q,s,n}^D &= (\mathbb{F}_q^{n-s}, \{D_s(x) \mid x \in \mathbb{F}_q^n\}), \\ \mathcal{H}_{q,s,n}^I &= (\mathbb{F}_q^{n+s}, \{I_s(x) \mid x \in \mathbb{F}_q^n\}). \end{aligned}$$

In each of these hypergraphs, hyperedges correspond to strings in  $\mathbb{F}_q^n$  and the vertices are strings in  $\mathbb{F}_q^{n-s}$  and  $\mathbb{F}_q^{n+s}$  for  $\mathcal{H}_{q,s,n}^D$  and  $\mathcal{H}_{q,s,n}^I$ , respectively. By Definition 2.3, an  $s$ -deletion correcting code in  $\mathbb{F}_q^n$  corresponds to disjoint hyperedges in  $\mathcal{H}_{q,s,n}^D$  and therefore corresponds to

a *matching* in  $\mathcal{H}_{q,s,n}^D$ . The size of the largest codebook for string length  $n$ ,  $|\mathcal{C}_{q,s,n}^*|$  is thus equal to  $\nu(\mathcal{H}_{q,s,n}^D)$ , the matching number of  $\mathcal{H}_{q,s,n}^D$ . The matching problem for  $\mathcal{H}_{q,s,n}^D$  when written explicitly, is as follows,

$$\begin{aligned} |\mathcal{C}_{q,s,n}^*| = & \underset{z}{\text{maximize}} \sum_{y \in \mathbb{F}_q^n} z(y) \\ \text{subject to} \quad & \sum_{y \in I_s(x)} z(y) \leq 1, \quad \forall x \in \mathbb{F}_q^{n-s}, \\ & z(y) \in \mathbb{Z}_+, \quad \forall y \in \mathbb{F}_q^n. \end{aligned}$$

Here the integer variables are denoted  $z(y), y \in \mathbb{F}_q^n$ . The constraints are that for each vertex  $x \in \mathbb{F}_q^{n-s}$ , the sum of  $z(y)$  over those  $y$  for which the hyperedge corresponding to  $y$  covers  $x$  (i.e.,  $y \in I_s(x)$ ) is at most unity. Since a code is an  $s$ -deletion correcting code if and only if it is an  $s$ -insertion correcting code, a matching of  $\mathcal{H}_{q,s,n}^D$  also corresponds to a  $s$ -deletion correcting code and thereby,  $\nu(\mathcal{H}_{q,s,n}^D) = |\mathcal{C}_{q,s,n}^*|$ .

Another characterization of the optimal codebook adopted in [19], [18], [1] employs the following graph.

**Definition 3.1:** Let  $L_{q,s,n}$  be the graph with vertex set  $\mathbb{F}_q^n$  wherein two vertices are adjacent if their Levenshtein distance is at most  $2s$ .

The optimal  $s$ -deletion codebook corresponds to the maximum independent set in this graph. The Levenshtein distance (restricted to  $\mathbb{F}_q^n \times \mathbb{F}_q^n$ ) is the shortest path metric on the graph  $L_{q,1,n}$ . The hypergraph characterization relates to this characterization through the concept of a line graph. Specifically,

**Lemma 3.1:** For any  $q, s, n \in \mathbb{N}$ , the graph  $L_{q,s,n}$  is the line graph of hypergraph  $\mathcal{H}_{q,s,n}^D$  and of hypergraph  $\mathcal{H}_{q,s,n}^I$ . Consequently,

$$\begin{aligned} \nu(\mathcal{H}_{q,s,n}^D) &= \alpha(L_{q,s,n}) = |\mathcal{C}_{q,s,n}^*|, \\ \nu(\mathcal{H}_{q,s,n}^I) &= \alpha(L_{q,s,n}) = |\mathcal{C}_{q,s,n}^*|. \end{aligned}$$

*Proof:* By the Definition 2.4 of Levenshtein distance and by Lemma 2.1, two vertices in  $L_{q,s,n}$  share an edge if and only if their  $s$ -deletion (and  $s$ -insertion) sets intersect. Consequently,  $L_{q,s,n} = L(\mathcal{H}_{q,s,n}^D) = L(\mathcal{H}_{q,s,n}^I)$ . By (15), the matching numbers of  $\mathcal{H}_{q,s,n}^D$  and  $\mathcal{H}_{q,s,n}^I$  are both equal to the independence number of  $L_{q,s,n}$ . ■

If one attempts to upper bound the size of a code by packing graph  $L_{q,s,n}$  with non-overlapping neighborhoods centered around strings in  $\mathbb{F}_q^n$ , the main difficulty encountered is that the resulting neighborhoods are not of the same size. This property of the Levenshtein distance is a fundamental departure from, say, the Hamming distance under which the sizes of the neighborhoods are same for every string.

Alternatively, one may pack  $\mathbb{F}_q^{n-s}$  with deletion sets of strings in  $\mathbb{F}_q^n$ . This approach too encounters the difficulty that deletion sets are of different sizes. For example for

$s = 1$ , if one argues that

$$|\mathcal{C}_{q,1,n}^*| \min_{x \in \mathbb{F}_q^n} |D_1(x)| \leq \sum_{x \in \mathcal{C}_{q,1,n}^*} |D_1(x)| \leq q^{n-1},$$

since  $\min_{x \in \mathbb{F}_q^n} |D_1(x)| = 1$ , one gets the bound  $|\mathcal{C}_{q,1,n}^*| \leq q^{n-1}$  which is far weaker than the asymptotic bound (the ratio  $\frac{q^{n-1}}{q^n/n(q-1)}$  approaches infinity for large  $n$ ). A similar situation results for  $s > 1$ . Levenshtein's bound (2) is obtained by a refinement of this approach in which strings are classified in two categories based on their number of runs.

Since insertion-correction and deletion-correction are equivalent, and since insertion sets are of the same size for each string of a given length (cf., (5)), one may exploit this to pack  $\mathbb{F}_q^{n+s}$  with insertion sets. Unfortunately, this leads to a weak upper bound. For example, for  $s = 1$  we get the bound  $\frac{q^{n+1}}{n(q-1)+q}$ , which is asymptotically  $q$  times larger than the known upper bound (this bound is  $\frac{2^{n+1}}{n+1}$  for binary alphabet and the asymptotic size is  $\frac{2^n}{n}$ ).

The approaches of packing deletion sets or insertion sets can be conceptually unified by casting them as matching problems on hypergraphs  $\mathcal{H}_{q,s,n}^D$  and  $\mathcal{H}_{q,s,n}^I$ , respectively. Since insertion sets are of the same size, hypergraph  $\mathcal{H}_{q,s,n}^I$  is *uniform* [37]; indeed the matching problem is well studied on uniform hypergraphs (see e.g., [37, Chapter 3],[41] and [42]). It is a quirk of the problem of deletion-correcting codes that although the characterization of  $\mathcal{C}_{q,s,n}^*$  via  $\mathcal{H}_{q,s,n}^I$  is analytically convenient and well studied, it leads to a weak bound.

The other hypergraph  $\mathcal{H}_{q,s,n}^D$  is *regular*, since all vertices in  $\mathcal{H}_{q,s,n}^D$  have the same number of hyperedges covering them [37]. Although this hypergraph does not belong to a category where the matching problem appears to be well studied, we show in the following sections that, if appropriately tackled, it does lead to a better bound. The crux of the proof of our bound lies in tackling this hypergraph.

### B. The non-asymptotic upper bounds for single-deletion correcting codes

In this section we present bounds on single-deletion correcting codes. The bounds we obtain are based on two concepts. The first is a monotonicity relationship between the number of runs of a string (recall Definition 2.1) under the operation of insertion. The second is the property that the size of the deletion set is also equal to the number of runs (cf. (4)). We first note the monotonicity.

**Lemma 3.2:** Let  $q, n \in \mathbb{N}$  and let  $x \in \mathbb{F}_q^n$  be a string. Then for any supersequence  $y \in I_1(x)$ , the number of runs of  $x$  and  $y$  satisfy  $r(x) \leq r(y)$ .

This lemma is quite obvious; we omit the proof for brevity.

Our proof utilizes Lemma 2.4; for easy reference the fractional transversal problem of  $\mathcal{H}_{q,1,n}^D$  is written below explicitly.

$$\begin{array}{ll} \tau^*(\mathcal{H}_{q,1,n}^D) = \underset{w}{\text{minimize}} & \sum_{x \in \mathbb{F}_q^{n-1}} w(x) \\ \text{subject to} & \sum_{x \in D_1(y)} w(x) \geq 1, \quad \forall y \in \mathbb{F}_q^n, \\ & w(x) \geq 0, \quad \forall x \in \mathbb{F}_q^{n-1}. \end{array}$$

Notice that the variables are  $w(x), x \in \mathbb{F}_q^{n-1}$  and the constraint is that for any  $y \in \mathbb{F}_q^n$ , the sum of  $w(x)$  over those  $x$  that are covered by the hyperedge corresponding to  $y$  (i.e.,  $x \in D_1(y)$ ), is at least unity.

*Theorem 3.1:* Let  $q, n \in \mathbb{N}, q \geq 2, n \geq 2$ . The optimal  $q$ -ary single-deletion correction code  $\mathcal{C}_{q,1,n}^*$  satisfies

$$|\mathcal{C}_{q,1,n}^*| \leq \frac{q^n - q}{(q-1)(n-1)}.$$

*Proof:* By Lemma 3.1, the size of the largest single-deletion correcting code equals the matching number of hypergraph  $\mathcal{H}_{q,1,n}^D$ , i.e.,  $\nu(\mathcal{H}_{q,1,n}^D) = |\mathcal{C}_{q,1,n}^*|$ . By Lemma 2.4, to show the required upper bound on  $\nu(\mathcal{H}_{q,1,n}^D)$  it suffices to construct a fractional transversal of  $\mathcal{H}_{q,1,n}^D$  with weight equal to  $\frac{q^n - q}{(q-1)(n-1)}$ . To this end, consider the fractional transversal  $w$ , where the component of  $w$  corresponding to string  $x \in \mathbb{F}_q^{n-1}$ , denoted  $w(x)$ , is given by

$$w(x) = \frac{1}{r(x)}, \quad \forall x \in \mathbb{F}_q^{n-1},$$

where  $r(x)$  is the number of runs of  $x$ . Clearly,  $w \geq 0$ . To show that  $w$  is indeed a fractional transversal, observe that for any  $y \in \mathbb{F}_q^n$ ,

$$\sum_{x \in D_1(y)} w(x) = \sum_{x \in D_1(y)} \frac{1}{r(x)} \stackrel{(a)}{\geq} \frac{|D_1(y)|}{r(y)} \stackrel{(b)}{=} 1.$$

The inequality in (a) follows from monotonicity relationship claimed in Lemma 3.2 and the equality in (b) follows from the size of the deletion set, given in (4). It only remains to calculate the weight of this transversal. For this, note that the number of strings of length  $n-1$  with exactly  $r$  runs is  $q(q-1)^{r-1} \times \binom{n-2}{r-1}$ . This is because, we have  $q$  choices for the symbol of the first run and for every subsequent run we have  $q-1$  choices for its symbol. The number of choices for the lengths of the runs equals the number of integral solutions  $(t_1, \dots, t_r)$  to

$$\sum_{i=1}^r t_i = n-1, \quad t_i \geq 1, 1 \leq i \leq r,$$

which, by Lemma 2.2, is  $\binom{n-2}{r-1}$ . Consequently, the weight of  $w$  is

$$\begin{aligned} \sum_{x \in \mathbb{F}_q^{n-1}} w(x) &= \sum_{r=1}^{n-1} q(q-1)^{r-1} \binom{n-2}{r-1} \cdot \frac{1}{r} \\ &= q \sum_{r=1}^{n-1} \frac{(n-2)!}{(n-r-1)!(r-1)!} \cdot \frac{1}{r} \cdot (q-1)^{r-1} \\ &\stackrel{(c)}{=} \frac{q}{(q-1)(n-1)} \sum_{r=1}^{n-1} \binom{n-1}{r} (q-1)^r \\ &= \frac{q((1+(q-1))^{n-1} - \binom{n-1}{0})}{(q-1)(n-1)} \\ &= \frac{q^n - q}{(q-1)(n-1)}. \end{aligned}$$

In (c), we have simplified  $\frac{(n-2)!}{(n-r-1)!(r-1)!} \cdot \frac{1}{r} = \frac{1}{n-1} \frac{(n-1)!}{(n-r-1)!r!}$ . By Lemma 2.4,  $\frac{q^n - q}{(q-1)(n-1)}$  is an upper bound on  $|\mathcal{C}_{q,1,n}^*|$ . ■

Although this bound is non-asymptotic, as a corollary we get the asymptotic results of Levenshtein [2] and Tenengolts [5].

*Corollary 3.2:* The optimal single-deletion correcting code for binary alphabet has size that asymptotically satisfies

$$|\mathcal{C}_{2,1,n}^*| \sim \frac{2^n}{n}.$$

The optimal single-deletion correcting code for  $q$ -ary alphabet satisfies

$$|\mathcal{C}_{q,1,n}^*| \lesssim \frac{q^n}{(q-1)n}.$$

*Proof:* For binary alphabet, Levenshtein [2] shows that the VT codes correct single deletions. These codes are of size at least  $\frac{2^n}{n+1}$ , whereby  $|\mathcal{C}_{2,1,n}^*| \geq \frac{2^n}{n+1}$ . Combining this with Theorem 3.1 shows that

$$\frac{2^n}{n+1} \leq |\mathcal{C}_{2,1,n}^*| \leq \frac{2^n - 2}{n-1}.$$

Thus  $\frac{|\mathcal{C}_{2,1,n}^*|}{2^n/n} \xrightarrow{n} 1$ . For the  $q$ -ary case, since by Theorem 3.1,  $|\mathcal{C}_{q,1,n}^*| \leq \frac{q^n - q}{(q-1)(n-1)}$ ,  $\lim_{n \rightarrow \infty} \frac{|\mathcal{C}_{q,1,n}^*|}{q^n/n(q-1)} \leq 1$ . ■

#### IV. NON-ASYMPTOTIC UPPER BOUNDS FOR MULTIPLE-DELETION CORRECTING CODES AND THE ASYMPTOTIC RATE FUNCTION

We now extend the logic used in the bound above to channels with multiple deletions.

And as we did in the single-deletion case, we will use the hypergraph  $\mathcal{H}_{q,s,n}^D$  to obtain our bound. The key property employed in the proof of Theorem 3.1 was that the number of runs of a string increases under the insertion of a symbol. This is in fact a specific consequence of a more general

property shown by Hirschberg and Regnier [26, Lemma 3.1]: for any  $s$ , the size of the  $s$ -deletion set of a string increases under the insertion of a symbol. This result is articulated in the following lemma. Here if  $x = x_1x_2 \dots x_n$  and  $y = y_1y_2 \dots y_m$  are  $q$ -ary strings, ' $xy$ ' denotes the string  $x_1x_2 \dots x_ny_1y_2 \dots y_m$ .

**Lemma 4.1:** Let  $s \in \mathbb{N}$ . For any strings  $x, y \in \mathbb{F}_q^*$  and any symbol  $\sigma \in \mathbb{F}_q$ ,  $|D_s(xy)| \leq |D_s(x\sigma y)|$ .

The original result from [26, Lemma 3.1] seems to pertain to nonempty strings  $x, y$ ; this is apparent from their proof. However the extension to the case where one of  $x, y$  is empty is trivial and we have included it in the above statement. The consequence is that, in this lemma,  $\sigma$  can be thought of as a symbol inserted into an existing string  $xy$ . A recursive application of Lemma 4.1 then immediately yields that for any  $s$  and any string  $x \in \mathbb{F}_q^n$ ,

$$|D_s(x)| \leq |D_s(y)|, \quad \forall y \in I_s(x). \quad (16)$$

Looking back at the size of the single-deletion set from (4), one sees that the monotonicity relationship of Lemma 3.2 is a special case of (16).

We now exploit (16) to give an upper bound on the size of an  $s$ -deletion correcting code for arbitrary  $s$ . The proof utilizes, as before, the fractional transversal problem of  $\mathcal{H}_{q,s,n}^D$ .

$$\begin{aligned} \tau^*(\mathcal{H}_{q,s,n}^D) = & \text{minimize } \sum_{w \in \mathbb{F}_q^{n-s}} w(x) \\ \text{subject to } & \sum_{x \in D_s(y)} w(x) \geq 1, \quad \forall y \in \mathbb{F}_q^n, \\ & w(x) \geq 0, \quad \forall x \in \mathbb{F}_q^{n-s}. \end{aligned}$$

**Theorem 4.1:** Let  $s, q, n \in \mathbb{N}$  such that  $n > s, q \geq 2$ . The optimal  $s$ -deletion correcting code  $\mathcal{C}_{q,s,n}^*$  satisfies

$$|\mathcal{C}_{q,s,n}^*| \leq \sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|}. \quad (17)$$

*Proof:* We construct a fractional transversal for  $\mathcal{H}_{q,s,n}^D$ . Consider the candidate fractional transversal  $w$ , such that for any  $x \in \mathbb{F}_q^{n-s}$ ,  $w(x) = \frac{1}{|D_s(x)|}$ . Obviously,  $w \geq 0$ . Furthermore, for any  $y \in \mathbb{F}_q^n$ ,

$$\sum_{x \in D_s(y)} w(x) = \sum_{x \in D_s(y)} \frac{1}{|D_s(x)|} \stackrel{(a)}{\geq} 1,$$

where (a) follows from the monotonicity relation (16). Thus  $w$  is indeed a fractional transversal of  $\mathcal{H}_{q,s,n}^D$ . Now by Lemma 2.4, the weight of  $w$  is an upper bound on  $\nu(\mathcal{H}_{q,s,n}^D) = |\mathcal{C}_{q,s,n}^*|$ , whereby the result follows. ■

In order to derive explicit bounds, we now discuss the sizes of  $s$ -deletion sets. For  $s \leq 5$ , Mercier et al. [28, Section III.D] give closed form formulae for the size of  $s$ -deletion sets, which unlike in the single-deletion case, have quite a complicated form. Closed form expressions for 2-deletion sets for binary alphabet are also given by Swart and

Ferreira [27] and Sloane [1]. The only results on deletion sets valid for arbitrary  $s$  are bounds. For all  $x \in \mathbb{F}_q^n$ , the  $s$ -deletion set of  $x$  admits the following lower bound, shown recently by Liron and Langberg [29, Theorem VI.2]. For any  $s < n$  and any string  $x \in \mathbb{F}_q^n$  with  $2 < r(x) \leq n$ ,

$$|D_s(x)| \geq \delta(r(x), s) + \sum_{i=s+r(x)-n-1}^{\min(s-2, r(x)-3)} \delta(r(x) - 2, i) \quad (18)$$

$$\text{where } \delta(r, s) \triangleq \begin{cases} \sum_{i=0}^s \binom{r-s}{i}, & r > s \geq 0, \\ 1, & s = r \geq 0, \\ 0, & s < 0 \text{ or } s > r. \end{cases} \quad (19)$$

Notice that this bound on  $|D_s(\cdot)|$  is always positive. Additionally it is an improvement on previous bounds of Levenshtein [25] and Hirschberg and Regnier [26].

By using the explicit formulae (e.g., [28], [27], [1]) for the sizes of  $s$ -deletion sets in (17), one may obtain explicit upper bounds on  $|\mathcal{C}_{q,s,n}^*|$ , for  $s \leq 5$ . For general  $s$ , we derive an upper bound on the right hand side of (17) by combining Theorem 4.1 with the lower bound in (18). Note that the explicit formulae will yield tighter bounds than the one below.

**Corollary 4.2:** Let  $s, q, n \in \mathbb{N}, q \geq 2, n > 2s$ . The optimal  $s$ -deletion correcting code  $\mathcal{C}_{q,s,n}^*$  satisfies

$$|\mathcal{C}_{q,s,n}^*| \leq U_{q,s,n},$$

where

$$\begin{aligned} U_{q,s,n} \triangleq & \sum_{r=3}^{n-s} \frac{q(q-1)^{r-1} \binom{n-s-1}{r-1}}{\delta(r, s) + \sum_{i=s+r-(n-s)-1}^{\min(s-2, r-3)} \delta(r-2, i)} \\ & + \sum_{r=1}^2 q(q-1)^{r-1} \binom{n-s-1}{r-1}, \end{aligned} \quad (20)$$

and  $\delta(\cdot, \cdot)$  is as defined in (19).

*Proof:* By Theorem 4.1, we have

$$|\mathcal{C}_{q,s,n}^*| \leq \sum_{x \in \mathbb{F}_q^{n-s}: r(x) \geq 3} \frac{1}{|D_s(x)|} + \sum_{x \in \mathbb{F}_q^{n-s}: r(x) < 3} \frac{1}{|D_s(x)|}.$$

For  $n-s > s$  and strings  $x \in \mathbb{F}_q^{n-s}$  such that  $r(x) \geq 3$ , the bound in (18) applies; furthermore, notice that for such  $x$ , the bound in (18) is strictly positive. So using (18) in the equation above, the first sum can be upper-bounded and the resulting bound is the first term in (20). The second sum in the equation above admits the trivial upper bound  $|\{x \in \mathbb{F}_q^{n-s} | r(x) \leq 2\}|$ , which is the second term in (20). Hence the bound. ■

One of the aims of this paper was to produce non-asymptotic upper bounds that imply known asymptotic bounds. We now show that the bound  $U_{q,s,n}$  meets this purpose. Our main result is that  $U_{q,s,n}$  (and the expression

$\sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|}$ ) implies the previous results of Levenshtein [2] stated in (1) for  $q = 2$ , and generalizes these results to  $q$ -ary alphabet.

In order to do this, we first show a *lower* bound on (the upper bound)  $U_{q,s,n}$ . For this we recall an upper bound on sizes of deletion sets due to Levenshtein [25]: for any  $n, q \in \mathbb{N}$ ,

$$|D_s(x)| \leq \binom{r(x) + s - 1}{s}, \quad \forall x \in \mathbb{F}_q^n. \quad (21)$$

**Lemma 4.2:** Let  $q, s, n \in \mathbb{N}$ ,  $n > 2s, q \geq 2$ . The upper bound  $U_{q,s,n}$  satisfies the lower bound

$$U_{q,s,n} \geq \sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|} \geq \frac{q^n - q \sum_{r=0}^{s-1} (q-1)^r \binom{n-1}{r}}{(q-1)^s \binom{n-1}{s}}.$$

*Proof:* The first inequality on the left follows from the proof of Corollary 4.2. To show the second inequality, use the upper bound on  $|D_s(\cdot)|$  from (21), to get that the sum  $\sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|}$  is no less than

$$\begin{aligned} \sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{\binom{r(x)+s-1}{s}} &= \sum_{r=1}^{n-s} \frac{q(q-1)^{r-1} \binom{n-s-1}{r-1}}{\binom{r+s-1}{s}} \\ &\stackrel{(a)}{=} \frac{q}{(q-1)^s \binom{n-1}{s}} \sum_{r=1}^{n-s} (q-1)^{r+s-1} \binom{n-1}{r+s-1}, \\ &= \frac{q^n - q \sum_{r=0}^{s-1} (q-1)^r \binom{n-1}{r}}{(q-1)^s \binom{n-1}{s}}. \end{aligned}$$

In (a) we have used that  $\binom{n-s-1}{r-1} = \binom{n-1}{r+s-1}$ . This proves the claim. ■

Notice that the above calculations are a generalization of our proof of the bound on single-deletion correcting codes in Theorem 3.1.

We now prove the asymptotics of  $U_{q,s,n}$  by deriving a matching asymptotic upper bound.

**Theorem 4.3:** Let  $q, s \in \mathbb{N}, q \geq 2$ . The upper bound on  $s$ -deletion correcting codes  $U_{q,s,n}$  satisfies

$$U_{q,s,n} \sim \sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|} \sim \frac{s!q^n}{(q-1)^s n^s},$$

as  $n \rightarrow \infty$ . Consequently, as  $n \rightarrow \infty$ ,

$$|\mathcal{C}_{q,s,n}^*| \lesssim \frac{s!q^n}{(q-1)^s n^s}.$$

*Proof:* Thanks to Lemma 4.2, to prove the first set of asymptotics, it suffices to show that  $U_{q,s,n} \lesssim \frac{s!q^n}{(q-1)^s n^s}$  as  $n \rightarrow \infty$ .

Fix  $r' \in \mathbb{N}$ ,  $1 \leq s \leq r' \leq n-s$ . We first claim that  $U_{q,s,n}$  satisfies

$$U_{q,s,n} \leq \sum_{r=r'}^{n-s} \frac{q(q-1)^{r-1} \binom{n-s-1}{r-1}}{\delta(r', s)} + \sum_{r=1}^{r'-1} q(q-1)^{r-1} \binom{n-s-1}{r-1}. \quad (22)$$

To see this, use (19) to conclude

$$\delta(r, s) + \sum_{i=s+r-(n-s)-1}^{\min(s-2, r-3)} \delta(r-2, i) \geq \delta(r, s) \geq \delta(r', s),$$

for any  $r \geq r'$ , and thus bound the terms in (20) corresponding to  $r \geq r'$ . For terms corresponding to  $r < r'$ , employ the trivial bound  $\delta(\cdot, \cdot) \geq 1$ . Eq (22) further implies

$$U_{q,s,n} \leq \frac{q^{n-s}}{\delta(r', s)} + \sum_{r=1}^{r'-1} q(q-1)^{r-1} \binom{n-s-1}{r-1}. \quad (23)$$

Consider a binomial distribution with parameters  $(n-s-1)$  and  $\frac{q-1}{q}$ . The Chernoff bound on the cumulative binomial distribution implies that for  $r' - 1 < \frac{q-1}{q}(n-s-1)$ , the sum  $\sum_{r=1}^{r'-1} q(q-1)^{r-1} \binom{n-s-1}{r-1}$  is no more than

$$q^{n-s} \exp \left( - \frac{((n-s-1)\frac{q-1}{q} - r' - 2)^2}{2\frac{q-1}{q}(n-s-1)} \right).$$

Setting  $r' = \mathbf{r} = \frac{q-1}{q}(n-s-1) - \sqrt{(n-s-1)\log(n-s-1)}$  in (23), using the Chernoff bound and the fact that  $\delta(\mathbf{r}, s) \sim s! \left(\frac{q-1}{q}\right)^s n^s$ , as  $n \rightarrow \infty$ , we get

$$U_{q,s,n} \lesssim \frac{s!q^n}{(q-1)^s n^s},$$

as  $n \rightarrow \infty$ . Combining this bound with Lemma 4.2, we get  $U_{q,s,n} \sim \sum_{x \in \mathbb{F}_q^{n-s}} \frac{1}{|D_s(x)|} \sim \frac{s!q^n}{(q-1)^s n^s}$ . Finally, by Corollary 4.2, we get  $|\mathcal{C}_{q,s,n}^*| \lesssim \frac{s!q^n}{(q-1)^s n^s}$ . ■

Note that in addition to clarifying the asymptotics of  $U_{q,s,n}$  the above theorem shows that using explicit formulae for  $|D_s(\cdot)|$  in (17) does not lead to any improvement over  $U_{q,s,n}$  in an asymptotic sense.

Notice that the right hand side in (23) closely resembles the expression in Levenshtein's bound from (2). In fact Levenshtein's expression in (2) contains the term  $\binom{n-1}{\cdot}$  in place of  $\binom{n-s-1}{\cdot}$ , and therefore appears to be weaker than (23). However this observation does not directly translate to a proof that our bound  $U_{q,s,n}$  is stronger than Levenshtein's bound. This is because the parameter  $r$  in (2) is allowed to vary between  $s-1$  and  $n-1$ , whereas in (23),  $r'$  is allowed to vary between  $s-1$  and  $n-s$ . If one could make the left argument that for any  $n, s$ , values of  $r$  in (2) beyond  $n-s$  are inconsequential to the comparison of (2) with  $U_{q,s,n}$ , one could establish that  $U_{q,s,n}$  is indeed a better

bound than Levenshtein's. We have empirically found that this is true; we discuss this in Section VI.

Finally, it is evident that the bound  $U_{q,s,n}$ , while explicit, is hard to reduce to a closed form for any  $s \neq 1$ . It appears that the single-deletion case is a unique one which allows for a neat calculation of a closed form expression.

#### A. The asymptotic rate function

Consider the case of a deletion channel where a fraction  $\tau \in [0, 1]$  of the symbols in a  $q$ -ary string are deleted. Denote by  $R_q(\tau)$  the asymptotic value of the rate of the largest code for this channel,

$$R_q(\tau) \triangleq \lim_{n \rightarrow \infty} \frac{1}{n} \log_q |\mathcal{C}_{q,\tau n,n}^*|. \quad (24)$$

We call  $R_q(\tau)$  the asymptotic rate function for the deletion channel. Very little seems to be known about this function. Levenshtein's non-asymptotic bounds from (2) only lead to the conclusion  $R_2(\tau) \leq 0.7729$  for  $\tau \geq 0.0757$  [6]. In this section we show that our non-asymptotic bound  $U_{q,s,n}$  from Corollary 4.2 allows for a calculation of a finer bound on  $R_q(\cdot)$ .

In order to perform this calculation, we need to address some technicalities. Notice that Corollary 4.2 assumes  $n > 2s$  to obtain the bound  $U_{q,s,n}$ . When  $s$  was fixed, this restriction was immaterial. But for  $s = \tau n$ , this restriction means that Corollary 4.2 can be used only for  $\tau < \frac{1}{2}$ . For  $\tau \geq \frac{1}{2}$ , we will use the trivial bound

$$|\mathcal{C}_{q,\tau n,n}^*| \leq \sum_{x \in \mathbb{F}_q^{n-\tau n}} \frac{1}{|D_s(x)|} \leq q^{(1-\tau)n}. \quad (25)$$

Denote by  $h_q(x)$ ,  $x \in [0, 1]$  the following function

$$h_q(x) = -x \log_q(x) - (1-x) \log_q(1-x) + x \log_q(q-1),$$

and let  $h(\cdot) \equiv h_2(\cdot)$ , denote the binary entropy function.

**Theorem 4.4:** Consider the asymptotic rate function  $R_q(\cdot)$  defined in (24). For  $\tau \in [0, \frac{1}{2})$ , the asymptotic rate function satisfies

$$R_q(\tau) \leq \max_{\rho \in [0, 1-\tau]} N(\rho; \tau) - D(\rho; \tau),$$

where

$$N(\rho; \tau) = (1-\tau)h_q\left(\frac{\rho}{1-\tau}\right),$$

$$D(\rho; \tau) = \max_{m_{\tau,\rho} \leq \mu \leq \min(\tau, \rho)} \frac{(\rho - \mu)h\left(\min\left(\frac{\mu}{\rho - \mu}, \frac{1}{2}\right)\right)}{\log_2 q},$$

and  $m_{\tau,\rho} = \max(2\tau + \rho - 1, 0)$ . For  $\tau \in [\frac{1}{2}, 1]$ , the asymptotic rate function satisfies

$$R_q(\tau) \leq (1-\tau).$$

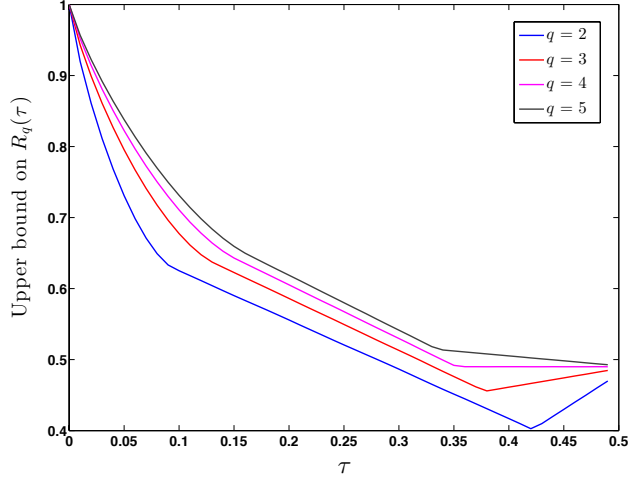


Fig. 1: The upper on the asymptotic rate function  $R_q(\tau)$  guaranteed by Theorem 4.4 for alphabet sizes  $q = 2, \dots, 5$  and  $\tau \in [0, \frac{1}{2})$ .

The proof is standard, but messy. We have relegated it to the Appendix.

Some remarks about this bound on  $R_q(\tau)$  are worth noting. Fig 1 contains plots of this bound pertaining to various alphabet sizes for  $\tau \in [0, \frac{1}{2})$ . For  $\tau = 0$ ,  $D(\rho; \tau) = 0$  and hence  $R_q(0) \leq \max_{\rho \in [0, 1]} h_q(\rho) = 1$ , which is expected. Thereafter for small values of  $\tau$  (say  $\tau \leq 1/10$ ), one finds that the rate drops quite sharply. For  $\tau \geq \frac{1}{2}$ , the above bound says  $R_q(\tau) \leq 1 - \tau$  and so  $R_q(1) = 0$ , as expected. One can easily see that this bound on the rate function is superior to Levenshtein's from [6].

However there are obvious shortcomings to our bound. Notice in Fig 1 that our bound never hits zero for any  $\tau \in [0, \frac{1}{2})$ ; in fact it becomes zero only for  $\tau = 1$ . Independently of his bound, Levenshtein [6] argues that  $R_q(\tau)$  must be zero for all  $\tau \geq \frac{q-1}{q}$ . Our bound does not imply this property (Levenshtein's bound on the rate function also does not imply this property). Furthermore, in each of the plots in Fig 1, our bound shows an increase beyond a certain value of  $\tau$ . The true asymptotic rate function  $R_q(\tau)$  must decrease monotonically with  $\tau$ . This indicates that our bound becomes vacuous after a certain value of  $\tau$ .

A fascinating lesson in this is that a non-asymptotic bound such as  $U_{q,s,n}$  that yields good asymptotics in one regime may not necessarily do so in other regimes.

#### V. BOUNDS ON CODES FOR CONSTRAINED SOURCES

The bounds obtained in the previous sections pertain to sizes of codebooks for the set of *all* strings of a particular string length and from a particular alphabet. We now consider the case where a codebook is sought for

a constrained set of source strings in  $\mathbb{F}_q^n$  and extend the results obtained above to present bounds for such codes.

**Definition 5.1:** Let  $S \subseteq \mathbb{F}_q^n$  be a set of strings and  $s \in \mathbb{N}$ . An  $s$ -deletion correcting code or  $s$ -deletion codebook for  $S$ , is a subset  $C \subseteq S$  such that the sets  $D_s(x), x \in C$ , are pairwise disjoint. The largest such code is denoted  $C_{S,s}^*$  and called the optimal  $s$ -deletion correcting code or optimal  $s$ -deletion codebook for  $S$ .

Finding a bound on the optimal codebook for an arbitrary set of strings  $S$  is significantly more challenging than finding one when  $S = \mathbb{F}_q^n$ . Specifically, arguments such as those based on Stirling's approximation employed by Levenshtein [2] and Tenengolts [5] rely on the availability of all strings in  $\mathbb{F}_q^n$ .

We construct our bound by using a suitable hypergraph. Let  $S \subseteq \mathbb{F}_q^n$  and define the hypergraph

$$\mathcal{H}_{S,s}^D = (D_s(S), \{D_s(x) : x \in S\}),$$

where  $D_s(S) = \bigcup_{x \in S} D_s(x)$ .  $\mathcal{H}_{S,s}^D$  is the partial hypergraph of  $\mathcal{H}_{q,s,n}^D$  generated by  $S$ . By arguments similar to those previously used, it follows that  $\nu(\mathcal{H}_{S,s}^D) = |C_{S,s}^*|$ . This matching problem for  $\mathcal{H}_{S,s}^D$  can be explicitly written as follows.

$$\begin{aligned} |C_{S,s}^*| = & \underset{z}{\text{maximize}} \sum_{y \in S} z(y) \\ \text{subject to} \quad & \sum_{y \in I_s(x) \cap S} z(y) \leq 1, \quad \forall x \in D_s(S), \\ & z(y) \in \mathbb{Z}_+, \quad \forall y \in S. \end{aligned}$$

Notice that in the constraint, the sum is over  $y$  belonging to  $I_s(x) \cap S$ ; this is because there may be a case where for some  $x \in D_s(S)$ , not all strings in  $I_s(x)$  are present in  $S$ , and may thereby not correspond to a hyperedge in  $\mathcal{H}_{S,s}^D$ . In the language of graphs, the codebook  $C_{S,s}^*$  is a maximum independent set in  $L_{S,s}$ , the subgraph of  $L_{q,s,n}$  induced by strings in  $S$ . As before, it is easy to see that  $L_{S,s}$  is the line graph of  $\mathcal{H}_{S,s}^D$ .

In constructing our bound we exploit the “decoupling” afforded by the fractional transversal problem for  $\mathcal{H}_{S,s}^D$ . This problem can be explicitly written as follows.

$$\begin{aligned} \tau^*(\mathcal{H}_{S,s}^D) = & \underset{w}{\text{minimize}} \sum_{x \in D_s(S)} w(x) \\ \text{subject to} \quad & \sum_{x \in D_s(y)} w(x) \geq 1, \quad \forall y \in S, \\ & w(x) \geq 0, \quad \forall x \in D_s(S). \end{aligned}$$

In this problem there is a separate constraint for each hyperedge, i.e. for each string in  $S$ . Consequently, a fractional transversal can be constructed for  $\mathcal{H}_{S,s}^D$  for any set  $S$  by applying the logic used in Theorem 4.1.

**Theorem 5.1:** Let  $q, s, n \in \mathbb{N}, n > s$  and let  $S$  be a set of strings in  $\mathbb{F}_q^n$ . Then

$$\frac{|S|}{\binom{n+s-1}{s} \iota_{q,s,n}} \leq |C_{S,s}^*| \leq \sum_{x \in D_s(S)} \frac{1}{|D_s(x)|}. \quad (26)$$

*Proof:* Notice that the fractional transversal problem for  $\mathcal{H}_{S,s}^D$  contains a constraint for each string  $y$  belonging to  $S$  and the sum in this constraint is over all  $x \in D_s(y)$ . Consequently, following Theorem 4.1, we see that  $w(x) = \frac{1}{|D_s(x)|}, x \in D_s(S)$ , is a fractional transversal of  $\mathcal{H}_{S,s}^D$ . The upper bound thus follows.

To obtain the lower bound consider the line graph  $L_{S,s}$  of  $\mathcal{H}_{S,s}^D$ . The maximum independent set in  $L_{S,s}$  is the optimal matching of  $\mathcal{H}_{S,s}^D$  and thereby the largest codebook  $C_{S,s}^*$ . A well known bound given by Brook's theorem or a “greedy” algorithm for independent set construction [35] gives that

$$\alpha(L_{S,s}) = |C_{S,s}^*| \geq \frac{|S|}{\Delta(L_{S,s}) + 1},$$

where  $\Delta(L_{S,s})$  is the maximum degree of a vertex in  $L_{S,s}$ . The neighborhood of a vertex  $x$  in  $L_{S,s}$  comprises of those strings obtained from  $x$  by deletion of  $s$  symbols in  $x$  followed by the insertion of  $s$  symbols in the resulting subsequence. Consequently,  $\Delta(L_{S,s}) \leq \max_{x \in S, y \in D_s(S)} |D_s(x)| |I_s(y)| - 1 \leq \binom{n+s-1}{s} \iota_{q,s,n} - 1$ , where we have used the upper bound on  $|D_s(\cdot)|$  from (21),  $\iota_{q,s,n}$  was defined in (6) as the size of the insertion set for strings in  $\mathbb{F}_q^{n-s}$ , and the subtracted 1 is because the string itself is counted at least once while counting neighbors produced by deletion and insertion. The result follows. ■

#### A. Run-length limited sources

In this section we will demonstrate the idea above by applying the results of Theorem 5.1 to the specific application of run-length limited codes. For simplicity we consider only the single-deletion case; but the idea is more general and can be extended readily to larger number of deletions. The background on these codes is sourced from the book chapter by Marcus, Roth and Siegel [43] and their extended monograph available online [44].

Recordings on a magnetic tape when encoded into a binary string result in strings that have no adjacent 1's and the number of 0's between two consecutive 1's is constrained to be in a certain range. Let  $0 \leq d \leq k$ . A binary string is said to satisfy a  $(d, k)$ -run-length limited (RLL) constraint if a) the string contains no adjacent 1's, i.e., the length of any 1-run is unity, b) the first and the last runs are 0-runs and c) the length of any 0-run is at least  $d$  and at most  $k$  [43]. In [44], the first and the last runs of 0's are allowed to have lengths less than  $d$ . In this section we assume, mainly for simplicity, that in a  $(d, k)$ -RLL string, the first and the last runs of the string must be 0-runs also having length at least  $d$ .

The problem of correcting errors in RLL strings has been considered by several authors (see [44, Chapter 9.5]) but most of these works consider erasure error or substitutions (see [22] and the discussion therein). Most works that consider deletion, consider the deletion of 0's only, since

that is most relevant to the application (see, e.g., the discussion in [17]). Recently Cheng et al. [23] and Palunčič et al. [24] have considered deletion errors in RLL strings for deletion of 0's and 1's.

Assume that a set of RLL strings as defined above are to be transmitted through a single-deletion channel, wherein both 0's and 1's can be deleted. In the theorem below we derive a bound on the size of the largest codebook for a  $(d, \infty)$ -RLL set of strings. For  $0 \leq d \leq k$ , by  $S_n(d, k) \subseteq \mathbb{F}_2^n$  we denote the set of binary strings of length  $n$  satisfying the  $(d, k)$ -RLL constraint. First, we characterize  $D_1(S_n(d, \infty))$ .

**Lemma 5.1:** Let  $n, d \in \mathbb{N}$  and  $1 < d \leq n$ . Then we have  $D_1(S_n(d, \infty)) = S_{n-1}(d, \infty) \cup S'_{n-1}(d, \infty)$ , where  $S'_{n-1}(d, \infty)$  is the set of binary strings of length  $n-1$  such that the first and last runs are 0-runs, between exactly one pair of consecutive 1's there are exactly  $d-1$  number of 0's and between all other pairs of consecutive 1's there are at least  $d$  0's.

*Proof:* “ $\subseteq$ ”: Consider a string in  $S_n(d, \infty)$ . A deleted symbol must be a 0 or a 1.

- 1) If a 0 is deleted there are two possibilities: either the run from which it is deleted has length  $d$ , or it has length  $> d$ . In the former case, the subsequence lies in  $S'_{n-1}(d, \infty)$ , while in the latter case, it lies in  $S_{n-1}(d, \infty)$ .
- 2) If a 1 is deleted, the 0-runs adjacent to the deleted 1 join to form a longer run of length at least  $2d$ ; the subsequence thus lies in  $S_{n-1}(d, \infty)$ .

This shows that in either case,  $D_1(S_n(d, \infty)) \subseteq S_{n-1}(d, \infty) \cup S'_{n-1}(d, \infty)$ .

“ $\supseteq$ ”: To show the opposite inclusion, it suffices to show that for any string  $x \in S_{n-1}(d, \infty) \cup S'_{n-1}(d, \infty)$  there exists a string  $y \in S_n(d, \infty)$  such that  $y \in I_1(x)$ . Consider an arbitrary  $x \in S_{n-1}(d, \infty) \cup S'_{n-1}(d, \infty)$ . Insert a 0 in the shortest 0-run of  $x$  and call the resulting string  $y$ . Since  $x$  has at most one 0-run of length  $d-1$ , it follows that  $y$  lies in  $S_n(d, \infty)$ . ■

Using this lemma and Theorem 5.1, we will prove an upper bound on the size of a code for  $S_n(d, \infty)$ .

**Theorem 5.2:** Let  $n, d \in \mathbb{N}$ ,  $1 < d \leq n$ . The optimal codebook for  $S_n(d, \infty)$ ,  $\mathcal{C}_{S_n(d, \infty)}^*$ , satisfies

$$|\mathcal{C}_{S_n(d, \infty)}^*| \leq \sum_{r=0}^{\bar{r}} \binom{n-2-r-(d-1)(r+1)}{r} \cdot \frac{1}{2r+1} + \sum_{r=1}^{\bar{r}'} (r+1) \binom{n-2-r-(d-1)(r+1)}{r-1} \cdot \frac{1}{2r+1}, \quad (27)$$

where  $\bar{r} = \lfloor \frac{n-1-d}{d+1} \rfloor$  and  $\bar{r}' = \lfloor \frac{n-d}{d+1} \rfloor$ .

*Proof:* From (26) and the size of single-deletion sets stated in (4),  $|\mathcal{C}_{S_n(d, \infty)}^*| \leq \sum_{x \in D_1(S_n(d, \infty))} \frac{1}{r(x)}$ . By

Lemma 5.1,  $D_1(S_n(d, \infty)) = S_{n-1}(d, \infty) \cup S'_{n-1}(d, \infty)$ . Notice that by definition of  $S'_{n-1}(d, \infty)$ , the sets  $S_{n-1}(d, \infty)$  and  $S'_{n-1}(d, \infty)$  are disjoint. Therefore,

$$|\mathcal{C}_{S_n(d, \infty)}^*| \leq \sum_{x \in S_{n-1}(d, \infty)} \frac{1}{r(x)} + \sum_{x \in S'_{n-1}(d, \infty)} \frac{1}{r(x)}. \quad (28)$$

Since all 0-runs of a string in  $S_{n-1}(d, \infty)$  have length at least  $d$  and all 1-runs have unit length, and the starting and ending runs are 0-runs, any string in  $S_{n-1}(d, \infty)$  has an odd number of runs and at most  $2\bar{r} + 1$  runs, where  $\bar{r}$  is as stated in the theorem. Therefore a string in  $S_{n-1}(d, \infty)$  with, say  $2r + 1$  runs, has  $r$  1-runs of unit length and  $r + 1$  0-runs of lengths say  $\ell_1, \dots, \ell_{r+1}$ , where each  $\ell_i \geq d$ . The number of strings with  $2r + 1$  runs in  $S_{n-1}(d, \infty)$  is thus equal to the number of integral solutions  $(\ell_1, \dots, \ell_{r+1})$  of

$$\sum_{i=1}^{r+1} \ell_i = n - 1 - r, \quad \ell_i \geq d, 1 \leq i \leq r + 1.$$

By Lemma 2.2 this number is  $\binom{n-2-r-(d-1)(r+1)}{r}$ , whereby the first term in the right hand side of (28) equals the first term in the right hand side of (27).

Each string in  $S'_{n-1}(d, \infty)$  also has odd number of runs. Furthermore, it has at least three runs and at most  $2\bar{r}' + 1$  runs, where  $\bar{r}'$  is defined in the statement of the theorem. Consider a string with  $2r + 1$  runs with  $r$  1-runs and  $r + 1$  0-runs. First choose the 0-run with length  $d - 1$ ; this can be chosen in  $r + 1$  ways. Let  $\ell_1, \dots, \ell_r$  be the lengths of the remaining 0-runs. The number of choices for the lengths of the remaining runs is the number of integral solutions of

$$\sum_{i=1}^r \ell_i = n - 1 - r - (d - 1), \quad \ell_i \geq d, 1 \leq i \leq r.$$

Using Lemma 2.2, the number of strings in  $S'_{n-1}(d, \infty)$  with  $2r + 1$  runs is thus  $(r + 1) \binom{n-2-r-(d-1)r-(d-1)}{r-1}$ . This proves that the second term in (27) equals its counterpart in (28). ■

Unfortunately, calculating these bounds in a simplified closed form does not appear to be easy. Our aim in this section was only to demonstrate the idea and the bound in Theorem 5.1. Exact calculation of these bounds is beyond the scope of this paper.

With this we conclude the theoretical portion of the paper. In the following sections we will study how our bounds compare numerically with the sizes of known codebooks and with other bounds.

## VI. NUMERICAL RESULTS

Recall that the upper bounds guaranteed by Theorems 3.1, 4.1 and 5.1 were obtained by constructing a fractional transversal for the hypergraphs involved. To obtain an upper bound on the size of optimal codebooks for the deletion

$n$	$\lfloor \text{Lev-UB} \rfloor$	$\lfloor \frac{2^n-2}{n-1} \rfloor$	$\lfloor \text{LP-UB} \rfloor$	$ \text{VT}_0(n) $
1	1	–	1	1
2	3	2	2	2
3	4	3	2	2
4	6	4	4	4
5	10	7	6	6
6	18	12	10	10
7	34	21	17	16
8	58	36	30	30
9	103	63	53	52
10	190	113	96	94
11	363	204	175	172
12	646	372	321	316
13	1182	682	593	586
14	2232	1260	1104	1096

(a)  $q = 2$ , binary

$n$	$\lfloor \text{Lev-UB} \rfloor$	$\lfloor \frac{q^n-q}{(n-1)(q-1)} \rfloor$	$\lfloor \text{LP-UB} \rfloor$	$ \text{Tenengolts} $
1	1	–	1	1
2	4	3	3	2
3	7	6	5	5
4	16	13	12	8
5	43	30	24	17
6	114	72	62	46
7	282	182	153	105
8	774	468	402	278

(b)  $q = 3$ 

$n$	$\lfloor \text{Lev-UB} \rfloor$	$\lfloor \frac{q^n-q}{(n-1)(q-1)} \rfloor$	$\lfloor \text{LP-UB} \rfloor$	$ \text{Tenengolts} $
1	1	–	1	1
2	6	4	4	3
3	12	10	8	6
4	36	28	25	20
5	132	85	69	52
6	405	272	231	178

(c)  $q = 4$ 

$n$	$\lfloor \text{Lev-UB} \rfloor$	$\lfloor \frac{q^n-q}{(n-1)(q-1)} \rfloor$	$\lfloor \text{LP-UB} \rfloor$	$ \text{Tenengolts} $
1	1	–	1	1
2	7	5	5	3
3	17	15	11	9
4	67	51	45	33
5	293	195	158	129
6	1146	781	657	527

(d)  $q = 5$ 

TABLE I: The columns of the table show, from left to right, the value of Levenshtein's bound from (2) ( $\text{Lev-UB}$ ), values of upper bound obtained in Theorem 3.1, the fractional matching number  $\nu^*(\mathcal{H}_{q,1,n}^D)$  ( $\text{LP-UB}$ ), and the sizes of best known codes, for values of  $q$  and  $n$ . For binary alphabet, the best known codes are the Varshamov-Tenengolts codes  $\text{VT}_0(n)$  [3], [2]. For larger alphabet, the best codes known to us are those of Tenengolts [5], whose size is denoted  $|\text{Tenengolts}|$ .

channel, it suffices to find the fractional matching number itself, and ideally one would like to have an expression for this number. We were not able to find such an expression and constructed a fractional transversal as a proxy for it.

In the case of a single deletion, there already exist codes which are known to be asymptotically good. This motivates a comparison between our bound for single-deletion correcting codes, the fractional matching number and the sizes of the best known codes in order to ascertain the quality of these codes. To do this, the fractional matching problem for hypergraph  $\mathcal{H}_{q,1,n}^D$  (for single deletions) was solved numerically on MATLAB for various values of  $q$  and  $n$ . Table I documents the results obtained.

In each subtable of Table I, the columns contain from left to right, the string length  $n$ , Levenshtein's upper bound (strongest one from (2); denoted  $\text{Lev-UB}$ ), the bound from Theorem 3.1, the value of the fractional matching number found numerically ( $= \nu^*(\mathcal{H}_{q,1,n}^D)$ ; denoted  $\text{LP-UB}$ ), and the best known code for each case. In the binary case the best known code is the Varshamov-Tenengolts code  $\text{VT}_0(n)$  where

$$\text{VT}_a(n) = \left\{ x_1 x_2 \dots x_n \in \mathbb{F}_2^n \mid \sum_i i x_i = a \bmod n+1 \right\}.$$

$\text{VT}_0(n)$  is also conjectured [1] to be optimal for all  $n$ . For larger alphabet the best codes we know of are those of Tenengolts [5] (these are denoted  $|\text{Tenengolts}|$ ). For each  $q$  the largest value of  $n$  is as far as we could compute with the resources available to us.

The first trend noticeable is that in any row values decrease from left to right. Thus the strongest of Levenshtein's bounds from (2) is weaker than our non-asymptotic bound. Our non-asymptotic bound is also weaker than the value of the fractional matching number (column  $\text{LP-UB}$ ); this shows that the fractional transversal we have constructed to obtain the upper bound is not the optimal fractional transversal.

Notice that in the binary case, shown in Table Ia, the size of the Varshamov-Tenengolts code  $\text{VT}_0(n)$  shows a good match with  $\text{LP-UB}$ . This indicates that these codes are either optimal (as conjectured) or close to being optimal, at least for  $n \leq 14$ . Sloane's website [4] carries numerically obtained bounds for  $n \leq 11$ , of which  $\text{VT}_0(n)$  has been confirmed as optimal for  $n \leq 10$ . The bounds on the website have been obtained by computing the Lovász  $\vartheta$  [35] on graphs  $L_{q,1,n}$ . The results in Table I may be considered as additions to Sloane's compilation.

For each value of  $q, n$ , Tenengolts' construction gives a two-parameter family of codes (the parameters being  $\beta, \gamma$  in [5, Eq (2)]). The column  $|\text{Tenengolts}|$  contains for the respective  $q, n$ , the largest code out of this family. Unlike in the VT codes where it is known that of the family  $\text{VT}_a(n), a = 0, \dots, n$ , the code  $\text{VT}_0(n)$  is the

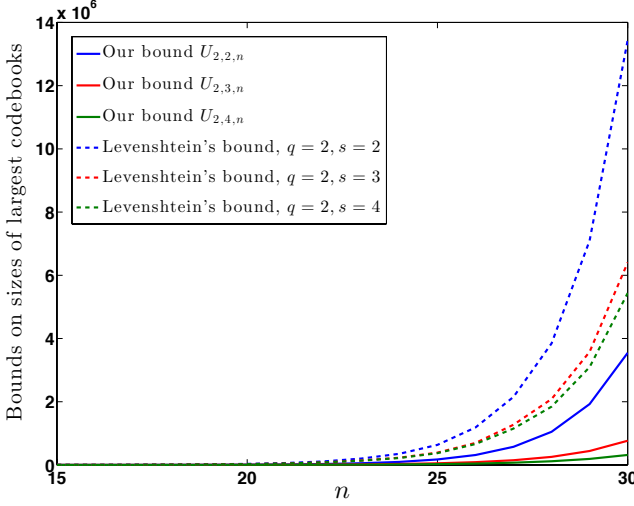


Fig. 2: Figure showing values of  $U_{q,s,n}$  (solid lines) and Levenshtein's bound (dotted lines) from (2) for  $q = 2$ ,  $s = 2, 3, 4$  and  $15 \leq n \leq 30$ .

largest, we are not aware of a similar characterization of the largest code from Tenengolts' family. Thus the column  $|\text{Tenengolts}|$  was populated by explicitly calculating the size of the code for each value of the parameters and thereafter identifying the largest of those. It is clear from this table that these codes are quite smaller than the fractional matching number in LP-UB. This may mean either that there is a large gap between the fractional matching number and the matching number for these hypergraphs, or that the Tenengolts codes are not optimal.

For larger number of deletions there exist no good codes apart from those found by search. So no interesting comparisons can be made for an existing code for a larger number of deletions. However, we may compare our bound with Levenshtein's from (2). Figure 2 shows the comparison for binary alphabet and  $s = 2, 3, 4$  and  $15 \leq n \leq 30$ . We have focused on this region of  $n$  so as to allow the distinctions between the lines for  $s = 2, 3, 4$  coming from Levenshtein's bound to be clearly discerned; for smaller values of  $n$  these lines overlap. One can easily eye-ball that our bound is significantly better than Levenshtein's.

We discuss the quality of our bound and prospects for improving it in the next section.

## VII. DISCUSSION

For the sake of this discussion, we limit ourselves to the case of the single-deletion channel. Table I shows that there is scope for improving our bound  $\frac{q^n - q}{(q-1)(n-1)}$  for the  $q$ -ary single-deletion channel. Since the bound is not equal to the fractional matching number LP-UB, one can obtain a better bound by merely finding a fractional

transversal with a smaller weight. However, in practice a construction to this effect has eluded us. In fact, our constructed transversal shows a close match to the optimal fractional transversal found numerically, which makes any improvement challenging. We discuss this below.

Figure 3 shows the optimal fractional transversal and the fractional transversal we have constructed ( $w(\cdot) \equiv \frac{1}{r(\cdot)}$ ) for hypergraph  $\mathcal{H}_{2,1,n}^D$ , i.e.  $q = 2, n = 8$  and  $s = 1$  and for hypergraph  $\mathcal{H}_{5,1,4}^D$  ( $q = 5, n = 4, s = 1$ ). Notice that in both cases, the constructed fractional transversal matches the general trend of the optimal fractional transversal. This continues to hold for larger values of  $n$ . Indeed, in the binary case, since

$$0 \leq \frac{\frac{2^n - 2}{n-1} - \nu^*(\mathcal{H}_{2,1,n}^D)}{2^{n-1}} \leq \frac{\frac{2^n - 2}{n-1} - \frac{2^n}{n+1}}{2^{n-1}} \rightarrow 0,$$

the average difference between the constructed and optimal transversal vanishes for large  $n$ . A tighter bound may be obtained by fine-tuning the constructed fractional transversal, but since the general trend of the optimal fractional transversal has already been captured by our constructed transversal, the logic for further fine-tuning is not obvious. Yet, this effort is not a lost cause: since the number of vertices grows exponentially, a small saving in this construction may imply a substantial improvement in the bound.

We end with one final consideration and speculate on what may be an alternative approach to obtaining better bounds. Since the most successful approaches to code construction for this problem have been number-theoretic one may be inclined to conjecture that the size of the optimal codebook  $|\mathcal{C}_{q,1,n}^*|$  depends not only on the numerical value of  $n$ , but also on properties  $n$  has as a number. In the binary case, in particular, since the fractional matching number  $\nu^*(\mathcal{H}_{2,1,n}^D)$  closely tracks  $|\text{VT}_0(n)|$ , which is given by a number-theoretic formula (see [1, Eq (7)]), it appears that  $\nu^*(\mathcal{H}_{2,1,n}^D)$  may also be given by a number-theoretic expression. In contrast, neither our bounds nor their proofs have any number-theoretic character. Perhaps a clue to tightening these bounds lies in giving a number-theoretic construction of the optimal fractional matching or a better (possibly optimal) fractional transversal.

In summary, this paper considered the deletion channel for general  $q$ -ary alphabet and an arbitrary number of deletions and proved new non-asymptotic upper bounds on the sizes of the optimal codebooks. The bounds are stronger than known bounds and imply classical asymptotic bounds. The bounds were derived via a hypergraph characterization of the optimal codebook and a linear programming argument. The approach was extended to derive bounds on codebooks for general constrained sources and was demonstrated for run-length limited sources. The paper concluded with a discussion on numerical results and on

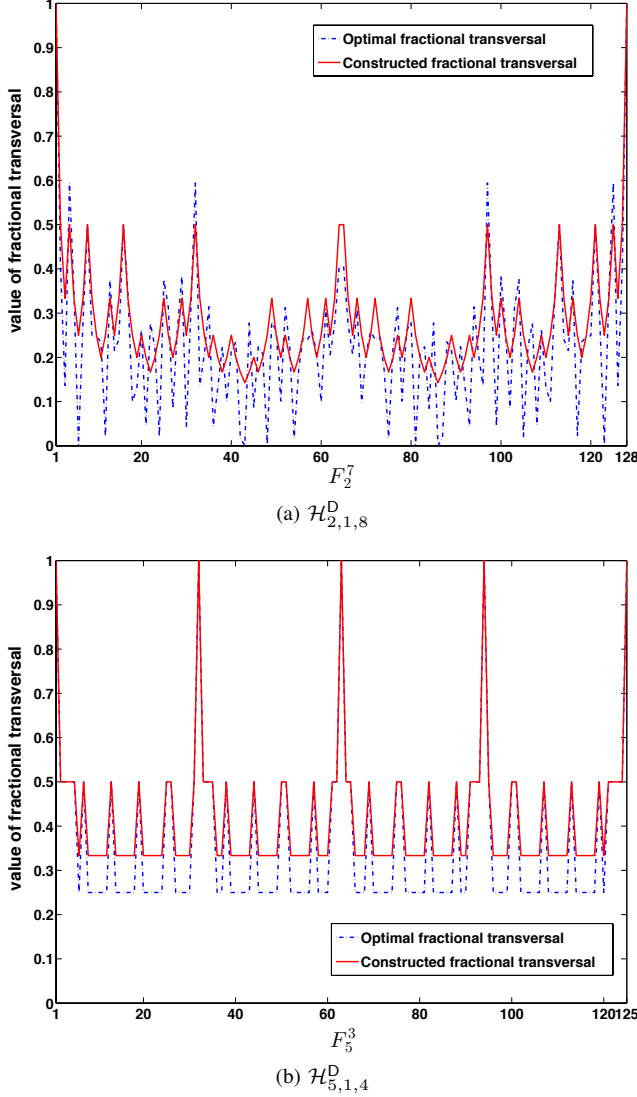


Fig. 3: The horizontal axis consists of elements of  $\mathbb{F}_2^7$  and  $\mathbb{F}_5^3$ , respectively, plotted in increasing order of their decimal value. The vertical axis is the value of the fractional transversals. In each case, the dotted line shows the optimal fractional transversal and the solid line shows the constructed fractional transversal  $w(x) \equiv \frac{1}{r(x)}$  for  $\mathcal{H}_{2,1,8}^D$  and  $\mathcal{H}_{5,1,4}^D$ , respectively. These lines are provided to aid in discerning the trends in their values; they have no meaning per se.

the quality of these bounds.

#### APPENDIX PROOF OF THEOREM 4.4

*Proof:* First consider  $\tau \in [0, \frac{1}{2})$ .

For such a value of  $\tau$ , the bound (20) applies. By (20),

$$U_{q,\tau n,n} = \sum_{r=3}^{(1-\tau)n} \frac{q(q-1)^{r-1} \binom{(1-\tau)n-1}{r-1}}{\delta(r, \tau n) + \sum_{i=(2\tau-1)n+r-1}^{\min(\tau n-2, r-3)} \delta(r-2, i)} + \sum_{r=1}^2 q(q-1)^{r-1} \binom{(1-\tau)n-1}{r-1}.$$

Notice that the second sum being a mere polynomial in  $n$  can be ignored in comparison to the first sum. Below, we focus only on the first term and estimate its asymptotics by finding its exponent.

Put  $r = \rho n$  so that  $\rho \in [0, 1 - \tau]$ , and let

$$N(\rho; \tau) = \lim_{n \rightarrow \infty} \frac{1}{n} \log_q q(q-1)^{\rho n-1} \binom{(1-\tau)n-1}{\rho n-1},$$

$$D_1(\rho; \tau) = \lim_{n \rightarrow \infty} \frac{1}{n} \log_q \delta(\rho n, \tau n),$$

$$D_2(\rho; \tau) = \lim_{n \rightarrow \infty} \frac{1}{n} \log_q \sum_{i=(2\tau-1+\rho)n-1}^{\min(\tau n-2, \rho n-3)} \delta(\rho n-2, i).$$

Here  $N(\rho; \tau)$  is the exponent of the numerator and the exponent of the denominator is

$$D(\rho; \tau) = \max(D_1(\rho; \tau), D_2(\rho; \tau)).$$

Therefore, the asymptotic rate function satisfies

$$R_q(\tau) \leq \max_{0 \leq \rho \leq 1-\tau} N(\rho; \tau) - D(\rho; \tau).$$

We now calculate the above exponents. It is easy to see that

$$N(\rho; \tau) = (1-\tau)h_q \left( \frac{\rho}{1-\tau} \right),$$

which is as required. Next consider  $D_1(\rho; \tau)$ . Clearly, if  $\rho \leq \tau$ ,  $D_1(\rho; \tau) = 0$ . If  $\tau \leq \frac{\rho-\tau}{2}$ , i.e.,  $\rho \geq 3\tau$ ,

$$D_1(\rho; \tau) = (\rho - \tau) \frac{h \left( \frac{\tau}{\rho - \tau} \right)}{\log_2 q}.$$

On the other hand if  $\rho < 3\tau$ ,  $D_1(\rho; \tau) = \frac{\rho - \tau}{\log_2 q}$ . In summary, we get

$$D_1(\rho; \tau) = \mathbb{I}_{\{\rho > \tau\}} \left( \frac{(\rho - \tau) h \left( \min \left( \frac{\tau}{\rho - \tau}, \frac{1}{2} \right) \right)}{\log_2 q} \right).$$

Now consider  $D_2(\rho; \tau)$ . Recall from (19) that if  $i < 0$ ,  $\delta(\rho n - 2, i) = 0$ . In the expression for  $D_2(\rho; \tau)$ , put

$i = \mu n$ , so that  $\mu \in [\max(2\tau + \rho - 1, 0), \min(\tau, \rho)]$ . Then arguing as above, we get

$$D_2(\rho; \tau) = \max_{m_{\tau, \rho} \leq \mu \leq \min(\tau, \rho)} \frac{(\rho - \mu)h\left(\min\left(\frac{\mu}{\rho - \mu}, \frac{1}{2}\right)\right)}{\log_2 q},$$

where  $m_{\tau, \rho} = \max(2\tau + \rho - 1, 0)$ , as stated in the theorem.

We now show that  $D_2(\rho; \tau)$  dominates  $D_1(\rho; \tau)$  for any  $\rho, \tau$ . If  $\rho \leq \tau$ ,  $D_1(\rho; \tau) \equiv 0$ , so, clearly,  $D_2(\rho; \tau) \geq D_1(\rho; \tau)$ . However, if  $\rho > \tau$ , we find that  $\mu = \tau$  satisfies  $\mu \in [m_{\tau, \rho}, \min(\tau, \rho)]$ . To see this, observe that a)  $\min(\tau, \rho) = \tau$ , since  $\rho > \tau$ , and b)  $\tau \geq m_{\tau, \rho}$  if and only if  $\rho \leq 1 - \tau$ , which is the assumed range on  $\rho$ . But for  $\mu = \tau$  the value of the maximand above equals  $D_1(\rho; \tau)$ . Consequently,  $D_2(\rho; \tau)$ , which involves a maximization over  $\mu$ , dominates  $D_1(\rho; \tau)$ . In summary,

$$D(\rho; \tau) = D_2(\rho; \tau),$$

as required. This completes the first part of the theorem pertaining to  $\tau \in [0, \frac{1}{2}]$ .

Now consider  $\tau \geq \frac{1}{2}$  and use the trivial bound from (25). In this case, clearly,

$$R_q(\tau) \leq (1 - \tau).$$

This covers all cases and the proof is complete. ■

## REFERENCES

- [1] N. J. A. Sloane, "On single-deletion-correcting codes," in *Codes and Designs: Proceedings of a Conference Honoring Professor Dijen K. Ray-Chaudhuri on the Occasion of His 65<sup>th</sup> Birthday, The Ohio State University, May 18-21, 2000*. Walter de Gruyter, 2002.
- [2] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.
- [3] R. R. Varshamov and G. M. Tenengolts, "Codes which correct single asymmetric errors (in Russian)," *Avtomatika i Telemekhanika*, vol. 6, no. 2, 1965.
- [4] N. J. A. Sloane, "Challenge problems: Independent sets in graphs," Jul. last updated 2011. [Online]. Available: <http://neilsloane.com/doc/graphs.html>
- [5] G. M. Tenengolts, "Nonbinary codes, correcting single deletion or insertion," *Information Theory, IEEE Transactions on*, vol. 30, no. 5, pp. 766 – 769, Sep. 1984.
- [6] V. I. Levenshtein, "Bounds for deletion/insertion correcting codes," in *2002 IEEE International Symposium on Information Theory, 2002. Proceedings*, Lausanne, Switzerland, 2002, p. 370.
- [7] V. S. Pless and W. C. Huffman, Eds., *Handbook of Coding Theory, Volume II*, 1st ed. North Holland, Nov. 1998.
- [8] H. Mercier, V. Bhargava, and V. Tarokh, "A survey of error-correcting codes for channels with symbol synchronization errors," *IEEE Communications Surveys Tutorials*, vol. 12, no. 1, pp. 87–96, 2010.
- [9] J. Ullman, "On the capabilities of codes to correct synchronization errors," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 95–105, Jan. 1967.
- [10] —, "Near-optimal, single-synchronization-error-correcting code," *IEEE Transactions on Information Theory*, vol. 12, no. 4, pp. 418–424, Oct. 1966.
- [11] A. Helberg and H. Ferreira, "On multiple insertion/deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 48, no. 1, pp. 305–308, Jan. 2002.
- [12] K. Abdel-Ghaffar, F. Palunčić, H. Ferreira, and W. Clarke, "On Helberg's generalization of the levenshtein code for multiple Deletion/Insertion error correction," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1804–1808, Mar. 2012.
- [13] L. Calabi and W. Hartnett, "Some general results of coding theory with applications to the study of codes for the correction of synchronization errors," *Information and Control*, vol. 15, no. 3, pp. 235–249, Sep. 1969.
- [14] E. Tanaka and T. Kasai, "Synchronization and substitution error-correcting codes for the Levenshtein metric," *IEEE Transactions on Information Theory*, vol. 22, no. 2, pp. 156–162, Mar. 1976.
- [15] R. R. Varshamov, "A class of codes for asymmetric channels and a problem from the additive theory of numbers," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 92–95, Jan. 1973.
- [16] S. Butenko, P. Pardalos, I. Sergienko, V. Shylo, and P. Stetsyuk, "Finding maximum independent sets in graphs arising from coding theory," in *Proceedings of the 2002 ACM symposium on Applied computing*, ser. SAC '02. New York, NY, USA: ACM, 2002, p. 542546.
- [17] L. Schulman and D. Zuckerman, "Asymptotically good codes correcting insertions, deletions, and transpositions," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2552–2557, Nov. 1999.
- [18] F. Khajouei, M. Zolghadr, and N. Kiyavash, "An algorithmic approach for finding deletion correcting codes," in *2011 IEEE Information Theory Workshop (ITW)*, Paraty, Brazil, Oct. 2011, pp. 25–29.
- [19] D. Cullina, A. A. Kulkarni, and N. Kiyavash, "A coloring approach to constructing deletion correcting codes from constant weight subgraphs," in *Proceedings of the ISIT*, Cambridge, USA, 2012.
- [20] R. Roth and P. Siegel, "Lee-metric BCH codes and their application to constrained and partial-response channels," *IEEE Transactions on Information Theory*, vol. 40, no. 4, pp. 1083–1096, Jul. 1994.
- [21] H. Hilden, D. Howe, and J. Weldon, E.J., "Shift error correcting modulation codes," *IEEE Transactions on Magnetics*, vol. 27, no. 6, pp. 4600–4605, Nov. 1991.
- [22] A. Bours, "Construction of fixed-length insertion/deletion correcting runlength-limited codes," *IEEE Transactions on Information Theory*, vol. 40, no. 6, pp. 1841–1856, Nov. 1994.
- [23] L. Cheng, H. Ferreira, and I. Broere, "Moment balancing templates for  $(d, k)$ -constrained codes and run-length limited sequences," *IEEE Transactions on Information Theory*, vol. 58, no. 4, pp. 2244–2252, Apr. 2012.
- [24] F. Palunčić, K. Abdel-Ghaffar, H. Ferreira, and W. Clarke, "A multiple Insertion/Deletion correcting code for run-length limited sequences," *IEEE Transactions on Information Theory*, vol. 58, no. 3, pp. 1809–1824, Mar. 2012.
- [25] V. I. Levenshtein, "On perfect codes in deletion and insertion metric," *Discrete Mathematics and Applications*, vol. 2, no. 3, pp. 241–258, Oct. 1992.
- [26] D. S. Hirschberg and M. Regnier, "Tight bounds on the number of string subsequences," *Journal of Discrete Algorithms*, vol. 1, no. 1, 2000.
- [27] T. Swart and H. Ferreira, "A note on double insertion/deletion correcting codes," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 269–273, Jan. 2003.
- [28] H. Mercier, M. Khabbazi, and V. Bhargava, "On the number of subsequences when deleting symbols from a string," *IEEE Transactions on Information Theory*, vol. 54, no. 7, pp. 3279–3285, Jul. 2008.
- [29] Y. Liron and M. Langberg, "A characterization of the number of subsequences obtained via the deletion channel," *CoRR*, vol. abs/1202.1644, 2012. [Online]. Available: <http://arxiv.org/abs/1202.1644>
- [30] V. I. Levenshtein, "Efficient reconstruction of sequences," *IEEE Transactions on Information Theory*, vol. 47, no. 1, pp. 2–22, Jan. 2001.
- [31] —, "Efficient reconstruction of sequences from their subsequences or supersequences," *J. Comb. Theory*, vol. 93, no. 2, pp. 310–332, 2001.
- [32] M. Mitzenmacher, "Polynomial time low-density parity-check codes with rates very close to the capacity of the  $q$ -ary random deletion

- channel for large  $q$ ,” *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5496–5501, Dec. 2006.
- [33] Y. Kanoria and A. Montanari. (2009) On the deletion channel with small deletion probability. [Online]. Available: <http://arxiv.org/abs/0912.5176>
  - [34] S. Diggavi, M. Mitzenmacher, and H. D. Pfister, “Capacity upper bounds for the deletion channel,” in *IEEE International Symposium on Information Theory, 2007. ISIT 2007*, Nice, France, Jun. 2007, pp. 1716–1720.
  - [35] D. B. West, *Introduction to Graph Theory*, 2nd ed. Prentice Hall, Sep. 2000.
  - [36] D. Sankoff and J. B. Kruskal, Eds., *Time warps, string edits, and macromolecules: the theory and practice of sequence comparison*. Addison-Wesley Pub. Co., Advanced Book Program, 1983.
  - [37] C. Berge, *Hypergraphs, Volume 45: Combinatorics of Finite Sets*, 1st ed. North Holland, Aug. 1989.
  - [38] E. R. Scheinerman and D. H. Ullman, *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*. Dover Publications, Dec. 2011.
  - [39] A. Schrijver, *Theory of Linear and Integer Programming*. John Wiley & Sons, Jun. 1998.
  - [40] J. Feldman, M. Wainwright, and D. Karger, “Using linear programming to decode binary linear codes,” *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.
  - [41] Z. Füredi, “Maximum degree and fractional matchings in uniform hypergraphs,” *Combinatorica*, vol. 1, no. 2, pp. 155–162, 1981.
  - [42] R. Aharoni, R. Holzman, and M. Krivelevich, “On a theorem of Lovász on covers in  $r$ -partite hypergraphs,” *Combinatorica*, vol. 16, no. 2, pp. 149–174, Jun. 1996.
  - [43] B. Marcus, P. Siegel, and R. Roth, “An introduction to coding for constrained systems,” in *Handbook of Coding Theory*, W. C. Huffman and V. Pless, Eds. Elsevier, 1998.
  - [44] ——. (2001) An introduction to coding for constrained systems. [Online]. Available: <http://www.math.ubc.ca/~marcus/Handbook/index.html>