

Efficient Compression of Long Arbitrary Sequences with No Reference at the Encoder

Yuval Cassuto, *Senior Member, IEEE*, and Jacob Ziv, *Life Fellow, IEEE*

Abstract

In a distributed information application an encoder compresses an arbitrary vector while a similar reference vector is available to the decoder as side information. For the Hamming-distance similarity measure, and when guaranteed perfect reconstruction is required, we present two contributions to the solution of this problem. One result shows that when a set of potential reference vectors is available to the encoder, lower compression rates can be achieved when the set satisfies a certain clustering property. Another result reduces the best known decoding complexity from exponential in the vector length n to $O(n^{1.5})$ by generalized concatenation of inner coset codes and outer error-correcting codes. One potential application of the results is the compression of DNA sequences, where similar (but not identical) reference vectors are shared among senders and receivers.

I. INTRODUCTION

Data compression exploits similarity between data to save transmission bandwidth or storage. Similarity can be internal to one data sequence, or external between multiple data sequences. In classical information theory, similarity is modeled through the abstraction of an *information source*, which is defined probabilistically [1]. Intra-sequence similarity exists because a long sequence is extracted from a source with a given probability distribution, and inter-sequence similarity is due to a non-trivial joint distribution between the sources that generate the sequences. Often times it is challenging to define the information source by a probability distribution. Such is the case, for example, in DNA sequences that are generated by nature, with a distribution that is unclear and hard to define. Still, compressing long sequences from unstructured sources is highly desired with the advent of data-rich applications, which generate, analyze, and manipulate volumes of these sequences.

In this paper we study and develop tools for compression of sequences lacking probabilistic models. The setup of interest is compressing at the encoder a sequence (vector) \mathbf{y} that is similar to a reference vector \mathbf{z} available at the decoder, while similarity is expressed by a bound on the Hamming distance between \mathbf{y} and \mathbf{z} . Our particular contributions to this setup are in two directions: first is a theoretical study of the case where the encoder has a set of candidate reference vectors, but does not know which particular \mathbf{z} from the set the decoder has; second is low-complexity compression and decompression for guaranteed zero-error reconstruction.

An encoder compressing a vector \mathbf{y} for a decoder having *side-information* \mathbf{z} is a classical and well-studied problem in information theory. In particular, it is covered (as a special case) by the *Slepian-Wolf* coding scheme [2] when the distributions of \mathbf{y} and \mathbf{y} -given- \mathbf{z} are known. For cases when the distributions are unknown, Ziv [3] pursued the *individual-sequence* approach where statistical properties are replaced by combinatorial finite-state complexity measures. However, these combinatorial measures too are hard to characterize for general sequences of certain type, e.g., DNA sequences. This leaves us with the *Hamming distance* as the most rudimentary and robust measure of similarity between sequences. Compressing \mathbf{y} given side-information \mathbf{z} at the decoder, where \mathbf{y} and \mathbf{z} have bounded Hamming distance, was studied by Orlitsky and Viswanathan in [4]. They show a reduction of the Hamming-bounded compression problem to error-correcting codes in the Hamming metric, under the framework of *coset coding*. A similar scheme but for sets instead of sequences appears in [5], and followed by extensions of the techniques motivated by biometric authentication [6]. Many results exist, starting with [7], that apply coset coding to source coding (see an extensive study in [8]), but the uniqueness of [4] is that zero-error reconstruction is *guaranteed*, as needed in the applications that drive our present study.

This paper continues the line of work on guaranteed-success compression with Hamming-bounded side information. In the first part of the paper (Section III), we study the case where the encoder as usual does not know the decoder's reference vector \mathbf{z} , but it does have a set Z of vectors that contains \mathbf{z} (among many other vectors). Our results in this part show that if the vectors in Z have a certain well-defined "clustering" property, then it is possible to reduce the compression rate below the best known. This can be achieved without any probabilistic assumptions on the set Z , and without directly enforcing a bound on its size. Our results in this part are for guaranteed-decoding *average* compression rate, where the average is taken over the random hash¹ function used, and *not* over the input \mathbf{y} (which has no probability distribution). For the same model our results also include a lower bound on compression rate for any scheme that uses random hashing. In the second part of the paper (Section IV), we return to the classical model of [4] (no Z in the encoder), and propose coding schemes with low

Parts of this work were presented at the 2020 International Zurich Seminar on Information and Communication.

Authors are with the Andrew and Erna Viterbi Department of Electrical Engineering, Technion - Israel Institute of Technology, Haifa, Israel 3200003 (emails: {ycassuto,jz}@ee.technion.ac.il).

¹hash functions are also known as binning functions in information theory.

complexity of encoding and decoding. For guaranteed decoding of length- n vectors with a constant fractional distance bound p , existing schemes require decoding complexity that is exponential in n due to the complexity of decoding an error-correcting code. Our proposed schemes have $O(n\sqrt{n})$ decoding complexity, which is low enough for practical implementation even for long input sequences. For low distance fractions p , our scheme has low compression rates, although not as low as the prior schemes that do not consider the decoding complexity. We use codes with structure similar to *generalized concatenation* (GC) codes [9], [10] – in particular *generalized error-locating* (GEL) codes [11], [12]. Applying the GEL code concatenation for compression requires to combine inner coset codes with outer error-correcting codes, while in the known construction both inner and outer codes are error-correcting codes. Moreover, using the known decoding algorithms for GEL (and GC) codes [13] results in total decoding complexity that is above quadratic in n , thus we use lower-complexity decoders to get the desired $O(n\sqrt{n})$. Our results show that when the distance fractions p are small, low compression rates are achieved, which thanks to the low complexity may offer an alternative to compression algorithms not using side information at all. If one lifts the requirement for guaranteed decoding, then existing work (e.g. [14], [15]) using classical concatenation [16] can achieve lower compression rates. Uyematsu [14] uses classical concatenation for Slepian-Wolf coding that succeeds with high probability over the source distribution, and Smith [15] provides a scheme for compression with side information at the decoder that succeeds with high probability over the shared randomness between encoder and decoder (this capability is extended to Slepian-Wolf coding in [17].)

The theoretical setups studied in this paper are general, and may find use in various data-rich distributed applications involving storage and communications. However, applications involving *DNA sequences* are a particular motivation for this study. DNA sequences are extremely long (hundreds of megabytes for full-genome sequences), and in emerging personal-medicine applications they are stored and communicated by various resource-limited entities. For DNA applications, the set Z of candidate reference vectors in Section III models similar sequences available in the sender’s local storage. The scheme of Section IV with its low guaranteed-decoding complexity is motivated by the long lengths of DNA sequences, and the importance of their perfect reconstruction. Most current compression schemes for DNA sequences use a reference sequence *in the encoder* (see, e.g., [18], [19]), and are thus forced to use generic reference vectors with weak similarity to the compressed vector \mathbf{y} . Freeing the encoder from having the reference vector allows compressing \mathbf{y} with a smaller distance parameter p , building on the many similar vectors the decoder has in its local storage.

The advantage of applying the generalized-concatenation approach for compression is that different inner codes can in future work accommodate additional similarity measures, for example \mathbf{y} and \mathbf{z} differing by *insertions and deletions*.

II. PROBLEM MODEL AND DEFINITIONS

In the problem setup we consider, there is an input vector we wish to convey (transmit or store) under the assumption that the party requesting this vector has a “similar” vector as a side-information vector (also called reference vector in the sequel). “Similar” here refers to having a bounded Hamming distance from the input vector. A length- n vector \mathbf{y} is given as input to the *encoder*, which maps \mathbf{y} to a vector $ENC(\mathbf{y})$ such that the *decoder* will be able to perfectly reproduce \mathbf{y} from $ENC(\mathbf{y})$ given a vector \mathbf{z} that satisfies $d_H(\mathbf{y}, \mathbf{z}) \leq pn$, where $0 < p < 1$ is a real-valued parameter and $d_H(\cdot, \cdot)$ is the standard Hamming distance between vectors. The vector \mathbf{z} at the decoder is *not* known to the encoder. An encoder+decoder pair is called a *coding scheme*. The objective is to find a coding scheme that minimizes $|ENC(\mathbf{y})|$, the number of bits in $ENC(\mathbf{y})$, where either the worst-case or average-case $|ENC(\mathbf{y})|$ will be of interest, and the average is taken with respect to the randomization used by the algorithms without assuming any probability distribution on \mathbf{y} . In both the worst case and the average case the decoder must recover \mathbf{y} without error.

A. New model: reference-vector set known to encoder

Let $Z = \{\mathbf{z}_1, \dots, \mathbf{z}_M\}$ be a set of vectors, where each vector \mathbf{z}_i is a binary vector of length n . The set Z is known to the encoder, and it contains the reference vector \mathbf{z} available at the decoder. While the encoder knows Z , it does *not* know the specific \mathbf{z} that the decoder has. The situation that the encoder knows Z (but not \mathbf{z}) can be encountered in practice when the encoder has access to a large repository of reference vectors, some of which are available to the decoder (but not clear which exactly).

B. Structured reference vectors: the p -spread parameter

Throughout the paper we will generally consider the set Z of reference vectors as general and arbitrary, and in particular not assumed to have any stochastic properties. One useful parameter to characterize Z is D_p we define next.

Definition 1. Given a set Z of reference vectors we define D_p as

$$D_p(Z) \triangleq \max_{\mathbf{z}_i, \mathbf{z}_j: d_H(\mathbf{z}_i, \mathbf{z}_j) \leq 2pn} d_H(\mathbf{z}_i, \mathbf{z}_j). \quad (1)$$

In words, $D_p(Z)$ is the maximal distance between a pair of vectors in Z whose distance is at most $2pn$.

Note that for any Z we have the upper bound $D_p(Z) \leq 2pn$. When this upper bound is strict, it means that the set Z has a “clustering” property, where vectors that are in the same neighborhood (have distance $\leq 2pn$) are not very far from each other (have distance $\leq D_p < 2pn$). For convenience, we define the p -spread parameter p' of Z as

$$p'(Z, p) \triangleq \frac{D_p(Z)}{2n}. \quad (2)$$

Later in the paper we will omit the arguments Z and p that are clear from the context, and just use p' . With this notation we have the upper bound

$$p' \leq p.$$

The definition of the p -spread parameter p' introduces structure to the set Z . When $p' = p$ the vectors in Z can be arbitrary, while $p' < p$ implies that the vectors in Z are more “clustered” in the sense that pairs are either close $d_H(\mathbf{z}_i, \mathbf{z}_j) \leq 2p'$ or far $d_H(\mathbf{z}_i, \mathbf{z}_j) > 2p$, with a forbidden distance range in between. The p -spread parameter is the simplest combinatorial way we have found to model vector clustering, which is an important feature in applications like DNA compression. It is important to note that the p -spread parameter does *not* degenerate Z to disjoint clusters of vectors with $d_H(\mathbf{z}_i, \mathbf{z}_j) \leq 2p'$, as seen in the next example.

Example 1. For $n = 7$, consider the following example of Z .

$$Z = \{0000000, 0111000, 1110000, 1111000, 1111111\}. \quad (3)$$

When $p = 3/7$, we see that $p'(Z, p) = 2/7$, because any two vectors in Z that are at distance 6 or less are also at distance 4 or less. The set Z models that both subsets $\{0000000, 0111000, 1110000, 1111000\}$ and $\{0111000, 1110000, 1111000, 1111111\}$ (which overlap) have some degree of similarity expressed in being at distance at most 4 from each other. Because 0000000 and 1111111 are not similar according to this definition, they must be dissimilar in the sense of being at distance more than 6 from each other.

C. Hamming balls and anticodes

In our results we define the proximity between input and reference vectors using the Hamming metric. Hence the following definitions will be useful. We denote by $B_l(\mathbf{x})$ the *Hamming ball* of radius l around the vector \mathbf{x} , that is, $B_l(\mathbf{x}) = \{\mathbf{s} \in \{0, 1\}^n : d_H(\mathbf{s}, \mathbf{x}) \leq l\}$. The size (number of vectors) of the Hamming ball is denoted $|B_l(\mathbf{x})|$, and because it does not depend on the argument \mathbf{x} we denote it $|B_l|$. We will use a well-known combinatorial inequality

$$\forall \alpha < 1/2, |B_{\alpha n}| \leq 2^{nH(\alpha)},$$

where $H(\alpha) \triangleq -\alpha \log_2(\alpha) - (1 - \alpha) \log_2(1 - \alpha)$ is the binary entropy function.

We also use the definition of an anticode. A set of vectors $S \subset \{0, 1\}^n$ is called an *anticode* of diameter l if any two vectors $\mathbf{s}_1, \mathbf{s}_2 \in S$ satisfy $d_H(\mathbf{s}_1, \mathbf{s}_2) \leq l$.

D. Random hash functions

A central tool in our proofs is *random hash functions*. A hash function $u : \{0, 1\}^n \rightarrow \{0, 1\}^m$ is a mapping from vectors of n bits to vectors of $m < n$ bits. A random hash function is a function u chosen randomly and uniformly from the set of hash functions $U_m = \{u : \{0, 1\}^n \rightarrow \{0, 1\}^m\}$, such that $\forall \mathbf{s}, \mathbf{x} \in \{0, 1\}^n, \mathbf{s} \neq \mathbf{x} : Pr[u(\mathbf{s}) = u(\mathbf{x})] \leq 1/2^m$. If this property is satisfied by a sub-class $\bar{U}_m \subseteq U_m$ under uniform sampling, then \bar{U}_m is called a *universal* class of hash functions [20]. An immediate fact about random hash functions from U_m or from any universal sub-class \bar{U}_m is that for any set of vectors $S \subset \{0, 1\}^n$ with $|S| = s$ and a vector $\mathbf{x} \notin S$, we have $Pr[\exists \mathbf{s} \in S : u(\mathbf{s}) = u(\mathbf{x})] \leq s/2^m$, which follows from the union bound. Note that this probability bound does not assume any probability distribution on the vectors \mathbf{x}, S .

III. COMPRESSION RATE VS. p -SPREAD PARAMETER

In this section we seek coding schemes that given a parameter p encode \mathbf{y} while knowing Z ; the p -spread parameter p' is known to the encoder from Z and p . We investigate how the compression rate $|ENC(\mathbf{y}, Z)|/n$ depends on p' . We seek coding schemes that guarantee the reconstruction of any \mathbf{y} *without error*. The achievable compression rates we derive are given as average over the shared randomness between encoder and decoder, but we emphasize that unlike similar results in information theory, we do not allow any (even vanishing) decoding error, and we do not assume any stochastic model for \mathbf{y} or Z . Formally, our coding schemes in this section operate over the following coding model.

Definition 2. A coding scheme with parameters p, p' has **zero-error average-rate** R if for any choice of \mathbf{y} and Z with $p'(Z, p) = p'$, \mathbf{y} can be uniquely recovered from $ENC(\mathbf{y}, Z)$ and any $\mathbf{z} \in Z$ s.t. $d_H(\mathbf{y}, \mathbf{z}) \leq pn$, and $|ENC(\mathbf{y}, Z)|/n = R$ on average over randomness shared by the encoder and decoder.

A useful subclass of Definition 2 is *simple-hashing* zero-error average-rate coding schemes, which we define next.

Definition 3. A zero-error average-rate coding scheme is called a **simple-hashing scheme** if with probability tending to 1 (as $n \rightarrow \infty$) it encodes \mathbf{y} as $u(\mathbf{y})$ such that for all $\mathbf{z} \in Z$ and $\mathbf{y}' \in B_{pn}(\mathbf{z})$, we have $u(\mathbf{y}') \neq u(\mathbf{y})$ unless $\mathbf{y}' = \mathbf{y}$. The probability is taken over the drawings of $u(\cdot) \in U_m$, where m is fixed given p, p', n .

Note that a simple-hashing scheme is free to encode \mathbf{y} arbitrarily with some (vanishing) probability, such that for every input it maintains the zero-error property of Definition 2.

A. Achievable rate with random hashing

In the first result we show a scheme in which the compression rate can be bounded by a simple function of p and the p -spread parameter of the set of reference vectors Z .

Theorem 1. Given the parameters p and p' , there exists a simple-hashing zero-error average-rate coding scheme with

$$\lim_{n \rightarrow \infty} \frac{|ENC(\mathbf{y}, Z)|}{n} \leq H(p) + H(p') + \epsilon, \quad (4)$$

and $\epsilon > 0$ is an arbitrary small real constant.

Before presenting the proof, we specify the encoder and decoder of the proposed coding scheme. The encoder and decoder share a random hash function from U_m (e.g., by sharing random bits independent of the input), where m is fixed and equal to n times the right-hand side of (4). The scheme in fact works with any universal subclass of U_m , which by using known universal classes with structure can significantly reduce the number of bits shared by the encoder and decoder. In the following we use the definition

$$Z(\mathbf{x}, \alpha) \triangleq Z \cap B_{\alpha n}(\mathbf{x}),$$

which is the set of reference vectors that are within distance αn from \mathbf{x} .

Construction 1. Let $u(\cdot)$ be a random hash function from U_m , where $m = n[H(p) + H(p') + \epsilon]$.

Encoder: 1) List all reference vectors in $Z(\mathbf{y}, p)$. 2) For each $\mathbf{z}_i \in Z(\mathbf{y}, p)$ apply the hash function u on all vectors in $B_{pn}(\mathbf{z}_i)$. In other words, apply u on all vectors in $\cup_{\mathbf{z}_i \in Z(\mathbf{y}, p)} B_{pn}(\mathbf{z}_i)$. 3) If no vector in these Hamming balls except \mathbf{y} is hashed to $u(\mathbf{y})$, output the bit 0 followed by $u(\mathbf{y})$; otherwise output the bit 1 followed by \mathbf{y} .

Decoder: 1) If first bit is 1, output the received \mathbf{y} . If first bit is 0, apply the hash function u on all vectors in $B_{pn}(\mathbf{z})$ and output the unique vector whose hash equals the received $u(\mathbf{y})$.

Proof: Given \mathbf{y} , by the problem statement the reference vector \mathbf{z} at the decoder satisfies $d_H(\mathbf{y}, \mathbf{z}) \leq pn$. The encoder can list all vectors $\mathbf{z}_i \in Z$ that satisfy $d_H(\mathbf{y}, \mathbf{z}_i) \leq pn$. From the triangle inequality we get that if \mathbf{z}_i and \mathbf{z}_j are each at distance at most pn from \mathbf{y} , then $d_H(\mathbf{z}_i, \mathbf{z}_j) \leq 2pn$. From the p -spread parameter of Z it follows that $d_H(\mathbf{z}_i, \mathbf{z}_j) \leq 2p'n$. Hence the list of potential \mathbf{z} vectors given \mathbf{y} is an anticode with diameter $2p'n$. It is known that the maximal size of an anticode with diameter $2p'n$ is $|B_{p'n}|$ [21]. Hence the set of vectors $\cup_{\mathbf{z}_i} B_{pn}(\mathbf{z}_i)$ hashed by the encoder has size bounded from above by $|B_{p'n}| \cdot |B_{pn}| \leq 2^{n[H(p') + H(p)]}$. From the properties of random hash functions, the probability that a vector in the set except \mathbf{y} will hash to $u(\mathbf{y})$ is at most $2^{-n\epsilon}$, going to zero as n grows. Hence the fraction of instances where the encoder outputs $u(\mathbf{y})$ tends to 1. This gives $|ENC(\mathbf{y})| \rightarrow m = n[H(p) + H(p') + \epsilon]$ as n tends to infinity. ■

The implication of Theorem 1 is that knowing the set Z at the encoder can improve the compression rate over known schemes when $p' < p$. For comparison, the scheme in [4] (which implicitly assumes the trivial $Z = \{0, 1\}^n$) gives $|ENC(\mathbf{y})| = nH(2p)$ with Gilbert-Varshamov non-explicit codes. Whenever $H(p) + H(p') < H(2p)$, Construction 1 offers a better compression rate. Note that Construction 1 indeed fulfills the *zero-error average-rate* property of Definition 2: the average rate is bounded by (4) for the worst-case Z given any \mathbf{y} , and for any \mathbf{z} at the decoder. Moreover, it is also a *simple-hashing scheme* because a fixed- m $u(\cdot)$ provides unique decoding with probability tending to 1.

In practice, Z may consist of reference vectors that are more “favorable” for compression than the cardinality upper bounds taken in the proof of Theorem 1. That means the benefits of knowing Z at the encoder exceed the tighter compression-rate upper bounds presented in this paper.

B. A converse result for random hashing

The scheme of Construction 1 encodes the input by random hashing of the vector \mathbf{y} . The next result shows that simple-hashing zero-error average-rate coding schemes are subject to a fundamental lower bound on $|ENC(\mathbf{y}, Z)|$.

Theorem 2. Given the parameters p and p' , any simple-hashing zero-error average-rate coding scheme must have

$$\lim_{n \rightarrow \infty} \frac{|ENC(\mathbf{y}, Z)|}{n} \geq H(p' + p). \quad (5)$$

Proof: First, since m is fixed and $u(\mathbf{y})$ is the encoder output with probability tending to 1, (5) is equivalent to the condition

$$\lim_{n \rightarrow \infty} \frac{m}{n} \geq H(p' + p). \quad (6)$$

By Definition 3, the encoder can output $u(\mathbf{y})$ only when there is no $\mathbf{y}' \neq \mathbf{y}$ within distance pn from $\mathbf{z} \in Z$ such that $u(\mathbf{y}') = u(\mathbf{y})$. In the proof we show that the probability over the functions $u(\cdot) \in U_m$ that no such \mathbf{y}' exists is vanishing with n if m does not satisfy (6). Given \mathbf{y} , an adversary sets $Z = B_{p'n}(\mathbf{y})$ and examines all the vectors $\mathbf{y}' \in B_{(p'+p)n}(\mathbf{y})$ and their hash values $u(\mathbf{y}')$. If there exists a \mathbf{y}' with $u(\mathbf{y}') = u(\mathbf{y})$, the adversary sets \mathbf{z} to be a vector in $Z = B_{p'n}(\mathbf{y})$ that is within distance pn from \mathbf{y}' ; such a vector exists because $\mathbf{y}' \in B_{(p'+p)n}(\mathbf{y})$, and as a result both \mathbf{y}, \mathbf{y}' are within distance pn from $\mathbf{z} \in Z$, as required. Asymptotically there are $s \triangleq 2^{nH(p'+p)}$ potential \mathbf{y}' vectors in $B_{(p'+p)n}(\mathbf{y})$. Denote $r = H(p' + p)$, and assume that m violates (6), thus $\lim_{n \rightarrow \infty} nr - m = \infty$. Going over all the functions in U_m , there are $(2^m)^s$ mappings from the vectors in $B_{(p'+p)n}(\mathbf{y})$ to the 2^m hash values. Out of these, there are $(2^m - 1)^s$ mappings in which all hash values are different from $u(\mathbf{y})$, which allow the encoder to successfully output $u(\mathbf{y})$. Taking the ratio between the number of successful mappings and the total number of mappings, we get

$$\frac{(2^m - 1)^s}{(2^m)^s} = [1 - 2^{-m}]^s = [1 - 2^{-m}]^{2^{nr}} = \left([1 - 2^{-m}]^{2^m}\right)^{2^{nr-m}} \xrightarrow[n \rightarrow \infty]{} e^{-2^{\lim_{n \rightarrow \infty} (nr-m)}} \rightarrow 0. \quad (7)$$

Since the fraction of successful mappings of $B_{(p'+p)n}(\mathbf{y})$ is vanishing with n , and uniformly drawing $u(\cdot) \in U_m$ induces a uniform distribution on these mappings, we proved that (6) is necessary to output $u(\mathbf{y})$ with non-vanishing probability, and (5) is necessary to output $u(\mathbf{y})$ with probability tending to 1. ■

The gap between $H(p' + p)$ (Theorem 2) and $H(p') + H(p)$ (Theorem 1) leaves room to potentially improve over Construction 1 while still using simple hashing. It is also possible that (5) can be improved by schemes that allow having \mathbf{y} and \mathbf{y}' with the same hash value, while finding a decoder that can somehow distinguish between the two hypotheses.

C. Reference-based coding

The random-hashing scheme of Section III-A is attractive thanks to its simplicity. However, when the decoder knows the near neighborhood of its reference vector \mathbf{z} in Z , the following coding scheme may achieve smaller values of $|ENC(\mathbf{y}, Z)|$. The idea of the next Construction 2 is that hashing is done *not* on the input \mathbf{y} , but on the reference vector in Z nearest to \mathbf{y} , which is used to encode \mathbf{y} along with a low-weight difference vector.

Construction 2. Let $u(\cdot)$ be a random hash function from U_m , where $m = n[2H(p') + \epsilon]$.

Encoder: 1) List all reference vectors in $Z(\mathbf{y}, p)$. 2) Find in the list the vector nearest to \mathbf{y} , denote it \mathbf{z}_1 and define $d \triangleq d_H(\mathbf{y}, \mathbf{z}_1)$ and $\mathbf{v}_1 \triangleq \mathbf{y} - \mathbf{z}_1$ (\mathbf{v}_1 is the difference vector between \mathbf{y} and \mathbf{z}_1 .) 3) For each $\mathbf{z}_i \in Z(\mathbf{y}, p)$ apply the hash function u on all vectors $\mathbf{z}_j \in Z(\mathbf{z}_i, 2p')$ such that $\mathbf{z}_j + \mathbf{v}_1 \in B_{pn}(\mathbf{z}_i)$. In other words, apply u on all vectors in $\cup_{\mathbf{z}_i \in Z(\mathbf{y}, p)} [Z(\mathbf{z}_i, 2p') \cap B_{pn}(\mathbf{z}_i - \mathbf{v}_1)]$. 4) If none of these vectors except \mathbf{z}_1 is hashed to $u(\mathbf{z}_1)$, output the bit 0 followed by $[u(\mathbf{z}_1), \text{enum}_d(\mathbf{v}_1)]$, where $\text{enum}_d(\mathbf{v}_1)$ is the index of \mathbf{v}_1 in an enumeration of $B_d(\mathbf{0})$ using $nH(\frac{d}{n})$ bits; otherwise output the bit 1 followed by \mathbf{y} .

Decoder: 1) If first bit is 1, output the received \mathbf{y} . If first bit is 0, apply the hash function u on all vectors in $Z(\mathbf{z}, 2p') \cap B_{pn}(\mathbf{z} - \mathbf{v}_1)$, and for the unique vector \mathbf{z}_1 whose hash equals $u(\mathbf{z}_1)$, output $\mathbf{z}_1 + \mathbf{v}_1$.

With the scheme in Construction 2 we get the following result, obtained under the same assumptions of Theorem 1, that is, asymptotically as $n \rightarrow \infty$ and on average over the random hash functions u .

Theorem 3. Let Z be a set of reference vectors with p -spread parameter p' . Then there exists a zero-error average-rate coding scheme with

$$\lim_{n \rightarrow \infty} \frac{|ENC(\mathbf{y}, Z)|}{n} \leq 2H(p') + \epsilon + H(\delta_H(\mathbf{y}, Z)), \quad (8)$$

where $\epsilon > 0$ is an arbitrary small real constant and $\delta_H(\mathbf{y}, Z) = d_H(\mathbf{y}, Z)/n$ is the fractional distance between \mathbf{y} and the nearest vector in Z .

Proof: We first note that the vectors $\mathbf{z}_i \in Z(\mathbf{y}, p)$ in part 3 of the encoder are all possible \mathbf{z} vectors at the decoder. In the proof of Theorem 1 we already saw that there are at most $2^{nH(p')}$ such vectors. Now for each \mathbf{z}_i considered as a possible \mathbf{z} vector at the decoder, the decoder does not know \mathbf{z}_1 , but knows that it is some $\mathbf{z}_j \in Z(\mathbf{z}_i, 2p')$ (because both $\mathbf{z}_i, \mathbf{z}_1$ are at distance at most pn from \mathbf{y}). We prove that for each $\mathbf{z}_i \in Z(\mathbf{y}, p)$ there are at most $2^{nH(p')}$ vectors $\mathbf{z}_j \in Z(\mathbf{z}_i, 2p')$ such that $\mathbf{z}_j + \mathbf{v}_1 \in B_{pn}(\mathbf{z}_i)$ (part 3 in the encoder); the latter property is required for \mathbf{z}_j to be consistent with \mathbf{z}_i at the decoder. To get this bound, observe that any pair $\mathbf{z}_j, \mathbf{z}_{j'}$ that both satisfy $\mathbf{z}_j + \mathbf{v}_1, \mathbf{z}_{j'} + \mathbf{v}_1 \in B_{pn}(\mathbf{z}_i)$ also satisfy $d_H(\mathbf{z}_j, \mathbf{z}_{j'}) \leq 2pn$, because both are in $B_{pn}(\mathbf{z}_i - \mathbf{v}_1)$. From the p -spread parameter this implies $d_H(\mathbf{z}_j, \mathbf{z}_{j'}) \leq 2p'n$. Now with the same argument

as in the proof of Theorem 1, we upper bound by $|B_{p'n}| \leq 2^{nH(p')}$ the number of vectors hashed in part 3 of the encoder for each $z_i \in Z(\mathbf{y}, p)$. Having bounded by $2^{2nH(p')}$ the union over all z_i of z_j vectors that may confuse the decoder given $z = z_i$, we conclude that a hash function with $n(2H(p') + \epsilon)$ output bits is sufficient with probability $1 - 2^{-n\epsilon}$ that tends to 1. To complete the proof, we add to the encoder output an enumeration of the difference vector v_1 , which can be done with $nH(\delta_H(\mathbf{y}, Z))$ bits according to [22]. ■

Discussion: If \mathbf{y} is relatively close in Hamming distance to *any* vector in Z (in particular not necessarily the z at the decoder), then Construction 2 allows to reduce the fractional encoding size from the $H(p) + H(p')$ of Theorem 1 closer to $2H(p')$ in the first term of (8). In the worst case $\delta_H(\mathbf{y}, Z)$ equals p , and then (8) becomes $2H(p') + H(p)$, which is not competitive with the upper bound offered by Construction 1. However, with “rich” Z sets many times the input \mathbf{y} would have a much closer z_1 vector. We have not been able to derive a converse result for reference-based coding. The core difficulty is to bound the advantage from the encoder’s freedom to choose the reference vector in Z (we do know how to get lower bounds when the encoder always uses the nearest vector as reference, like in Construction 2).

We add that it is easy to combine Constructions 2 and 1 such that the encoder chooses to hash z_1 when one is close to \mathbf{y} , and \mathbf{y} itself when its near neighborhood in Z is empty. This combination will only require another bit to mark to the decoder which of the constructions is used for each \mathbf{y} .

IV. FIXED-RATE COMPRESSION WITH LOW COMPLEXITY

In addition to this paper’s focus on having no statistical assumptions on the source and side information, in this section we aim to get schemes with *guaranteed worst-case* compression rates, and not just average rates with random hashing as in Section III. We also return here to the more classical setup where the encoder does not have a list of possible reference vectors, so its knowledge is limited to the fact that the decoder’s z vector is at distance at most pn from the input \mathbf{y} . In the terminology of Section II we thus have $Z = \{0, 1\}^n$, and $p'(Z, p) = p$ (trivial p -spread parameter). This problem is classical and well studied, but our proposed schemes will allow to solve it efficiently even for long sequences, for example DNA sequences.

A. Background: a known guaranteed fixed-rate scheme

For the setup of compressing a length- n vector \mathbf{y} with an unknown $z \in B_{pn}(\mathbf{y})$ at the decoder, [4] proposed a coset-coding approach, where a length- n binary linear code with minimum distance $> 2pn$ is taken and used as follows.

Construction 3. [4] Let \mathcal{C} be a binary linear code with minimum distance $> 2pn$, and S be a $\rho n \times n$ parity-check matrix for \mathcal{C} , $\rho \in (0, 1)$.

Encoder: Given an input row vector \mathbf{y} , calculate $\mathbf{s} = S\mathbf{y}^T$, and output \mathbf{s} .

Decoder: Find the lowest-weight vector \mathbf{v} such that $S\mathbf{v}^T = \mathbf{s} + S\mathbf{z}^T$; output $\mathbf{z} + \mathbf{v}$.

The output vector $\tilde{\mathbf{y}} = \mathbf{z} + \mathbf{v}$ satisfies $S\tilde{\mathbf{y}}^T = \mathbf{s}$, like \mathbf{y} , and having more than one such vector in $B_{pn}(z)$ would violate the minimum distance of \mathcal{C} . Hence $\tilde{\mathbf{y}} = \mathbf{y}$. This construction is guaranteed to succeed in recovering \mathbf{y} so long that indeed $d_H(z, \mathbf{y}) \leq pn$ as specified. In terms of complexity, the encoder of Construction 3 performs a matrix-vector product, with ρn^2 bit operations. The decoding complexity is much higher (equivalent to maximum-likelihood decoding of an error-correcting code); even if polynomial-time sub-optimal decoding is used, decoding complexity may be prohibitive for the values of n typical in applications like DNA sequences. Because of that issue, in the remainder of the section we develop guaranteed-decoding constructions that reduce decoding complexity by encoding the long sequence into a codeword composed of shorter sub-block codewords.

B. Construction idea

Our low-complexity constructions are based on the idea of *generalized concatenation (GC)* [9], adapted to the use of the codes for compression. As in GC, a long (length n) binary vector is broken to much shorter (length k) sub-vectors, and non-binary outer codes encode a desired dependence among the sub-vectors. Different from GC, the encoder output is not a concatenated codeword, but only parity symbols of the outer codes. The key difference is that here for compression, the concatenation needs to design outer error-correcting codes for inner *coset* codes, and not inner error-correcting codes as usual. Moreover, to keep the decoding complexity below quadratic in n , we design our codes with single-shot decoders for the outer codes. This is in contrast to the common use of GC constructions employing iterative decoders that decode up to half the minimum distance [13], building on the generalized minimum distance (GMD) method [23]. The particular sub-class of GC codes found useful here is *generalized error-locating (GEL)* codes [12], because their construction through inner syndromes fits well the syndrome method of Construction 3.

C. First efficient construction

We first define a partition of length- n vectors to $t \triangleq n/k$ sub-vectors of length k each, where k is some integer that divides n . Thus for example $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_t]$, where \cdot represents vector concatenation. Let $\{\mathbf{S}^{(i)}\}_{i=1}^m$ be a set of binary matrices where $\mathbf{S}^{(i)}$ has dimensions $r_i \times k$. For a sub-vector \mathbf{y}_j we further define the *partial i -th syndrome* as

$$\mathbf{s}_j^{(i)} = \mathbf{S}^{(i)} \mathbf{y}_j^T.$$

$\mathbf{s}_j^{(i)}$ is a column vector of dimension r_i . We take the matrices $\{\mathbf{S}^{(i)}\}_{i=1}^m$ to be a nested set, meaning that for $i' > i$ the r_i rows of $\mathbf{S}^{(i)}$ appear in $\mathbf{S}^{(i')}$ in concatenation with additional $r_{i'} - r_i$ rows. This implies that r_i is increasing with i . When $\mathbf{S}^{(i)}$ is seen as a parity-check matrix of a length- k code $\mathcal{C}^{(i)}$, we denote its minimum distance by d_i . From the nesting property we know that d_i is non-decreasing with i . We define the differential matrix $\tilde{\mathbf{S}}^{(i)}$ to contain the rows in $\mathbf{S}^{(i)}$ that do not appear in $\mathbf{S}^{(i-1)}$, and the number of rows in $\tilde{\mathbf{S}}^{(i)}$ is denoted $\tilde{r}_i \triangleq r_i - r_{i-1}$. For these definitions, $\mathbf{S}^{(0)}$ is defined as the empty matrix, hence $\tilde{\mathbf{S}}^{(1)} = \mathbf{S}^{(1)}$ (and $\tilde{r}_1 = r_1$). Define also \mathbf{I}_b as the identity matrix of order b . Our first concatenated construction now follows.

Construction 4. Let $\{\mathcal{C}^{(i)}\}_{i=1}^m$ be a nested set of length- k binary codes with parity-check matrices $\{\mathbf{S}^{(i)}\}_{i=1}^m$ and minimum distances $\{d_i\}_{i=1}^m$. In addition, define the set $\{\mathbf{H}^{(i)}\}_{i=2}^m$ where $\mathbf{H}^{(i)}$ is a $\rho_i \times t$ parity-check matrix over the finite field $F_{2^{\tilde{r}_i}}$ that defines a code with minimum distance δ_i . Let $\mathcal{E}^{(i)} : F_{2^{\tilde{r}_i}}^t \rightarrow F_{2^{\rho_i}}^{\rho_i}$ be an encoder function mapping a length t vector over $F_{2^{\tilde{r}_i}}$ to the parity symbols of the code whose parity-check matrix is $[\mathbf{H}^{(i)}, \mathbf{I}_{\rho_i}]$. Define $\mathcal{D}^{(i)} : F_{2^{\tilde{r}_i}}^t \times F_{2^{\rho_i}}^{\rho_i} \rightarrow F_{2^{\tilde{r}_i}}^t$ to be a decoder function with inputs \mathbf{a}, \mathbf{b} that finds the vector \mathbf{x} nearest to \mathbf{a} that satisfies $\mathbf{H}^{(i)} \mathbf{x}^T = \mathbf{b}$.

Encoder: Given an input row vector \mathbf{y} :

- 1) Partition $\mathbf{y} = [\mathbf{y}_1, \dots, \mathbf{y}_t]$.
- 2) Calculate $\mathbf{u}_j^{(i)} := \tilde{\mathbf{S}}^{(i)} \mathbf{y}_j^T$ for each $i \in \{1, \dots, m\}$ and $j \in \{1, \dots, t\}$.
- 3) Encode $\mathbf{p}^{(i)} := \mathcal{E}^{(i)}(\mathbf{u}_1^{(i)}, \dots, \mathbf{u}_t^{(i)})$ for each $i \in \{2, \dots, m\}$, and define $\mathbf{p}^{(1)} := [\mathbf{u}_1^{(1)}, \dots, \mathbf{u}_t^{(1)}]$.
- 4) Output $\mathbf{p}^{(i)}$, for every $i \in \{1, \dots, m\}$.

Decoder: Given encoder outputs $\mathbf{p}^{(i)}$ and reference row vector \mathbf{z} :

- 1) Partition $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_t]$.
- 2) Initialize $\hat{\mathbf{s}}_j^{(1)} := \mathbf{u}_j^{(1)}$, for each $j \in \{1, \dots, t\}$.

Iterate on $i = 2, \dots, m$ in 3-5 below:

- 3) For each j , find the lowest-weight vector \mathbf{v}_j such that $\mathbf{S}^{(i-1)} \mathbf{v}_j^T = \hat{\mathbf{s}}_j^{(i-1)} + \mathbf{S}^{(i-1)} \mathbf{z}_j^T$.
- 4) Take $\hat{\mathbf{y}}_j = \mathbf{z}_j + \mathbf{v}_j$ and calculate

$$\hat{\mathbf{u}}^{(i)} := \mathcal{D}^{(i)}(\tilde{\mathbf{S}}^{(i)} \hat{\mathbf{y}}_1^T, \dots, \tilde{\mathbf{S}}^{(i)} \hat{\mathbf{y}}_t^T; \mathbf{p}^{(i)}).$$

- 5) Concatenate $\hat{\mathbf{s}}_j^{(i)} := [\hat{\mathbf{s}}_j^{(i-1)}; \hat{\mathbf{u}}_j^{(i)}]$, for each $j \in \{1, \dots, t\}$.

Output:

- 6) For each j , find the lowest-weight vector \mathbf{v}_j such that $\mathbf{S}^{(m)} \mathbf{v}_j^T = \hat{\mathbf{s}}_j^{(m)} + \mathbf{S}^{(m)} \mathbf{z}_j^T$.
- 7) Output $\hat{\mathbf{y}}_j = \mathbf{z}_j + \mathbf{v}_j$, for each $j \in \{1, \dots, t\}$.

In each iteration i the decoder of Construction 4 takes the partial syndromes $\hat{\mathbf{s}}_j^{(i-1)}$ from the previous iteration, uses decoders for the (inner) code $\mathbf{S}^{(i-1)}$ to find the nearest word to \mathbf{z}_j with partial syndrome $\hat{\mathbf{s}}_j^{(i-1)}$, and then calculates the next differential syndromes $\tilde{\mathbf{S}}^{(i)} \hat{\mathbf{y}}_j^T$ of these nearest words. The iteration ends with correcting errors in the differential syndromes using the (outer) code $\mathbf{H}^{(i)}$, and obtaining the next partial syndromes $\hat{\mathbf{s}}_j^{(i)}$. The efficient realization of the steps in the encoder and decoder is discussed in Section IV-E.

D. Code parameters for guaranteed decoding

Construction 4 needs to work with the only specification being that $d_H(\mathbf{z}, \mathbf{y}) \leq pn$, that is, a distance bound for the full block. Then we specify parameters for the codes $\{\mathbf{S}^{(i)}\}_{i=1}^m$ and $\{\mathbf{H}^{(i)}\}_{i=2}^m$ that are sufficient for guaranteed decoding with Construction 4. The following lemma is the main tool for setting these parameters.

Lemma 4. Let $d_H(\mathbf{z}, \mathbf{y}) \leq pn$, and take Construction 4 with parity-check matrices $\{\mathbf{S}^{(i)}\}_{i=1}^m$ of binary codes with minimum distances $\{d_i\}_{i=1}^m$. Then correct decoding of $\mathbf{s}_j^{(m)}$ by $\hat{\mathbf{s}}_j^{(m)}$ is guaranteed if for each $i \in \{2, \dots, m\}$ we use a parity-check matrix $\mathbf{H}^{(i)}$ of a code with minimum distance $\delta_i > 4pn/(d_{i-1} - 1)$.

Proof: The basic observation is that $d_H(\mathbf{z}_j, \mathbf{y}_j) > (d_{i-1} - 1)/2$ can occur in less than $2pn/(d_{i-1} - 1)$ of the indices $j \in \{1, \dots, t\}$. Since the previous inequality is necessary for $\hat{\mathbf{y}}_j \neq \mathbf{y}_j$ in step 4, the decoder $\mathcal{D}^{(i)}$ will see less than $2pn/(d_{i-1} - 1)$ errors, and can correct them with distance $\delta_i > 4pn/(d_{i-1} - 1)$ for the code $\mathbf{H}^{(i)}$. Recovering the correct $\mathbf{u}_j^{(i)}$ for all i, j guarantees that at every iteration i , $\hat{\mathbf{s}}_j^{(i)} = \mathbf{s}_j^{(i)}$, including in iteration m . ■

Recall $n = kt$, and pick an integer m . For the matrix $S^{(i)}$ we specify the minimum distance

$$d_i = 4pk + \frac{i}{m} \left(\frac{1}{2} - 4p \right) k + 1, \quad i \in \{1, \dots, m-1\}, \quad (9)$$

and for $S^{(m)}$ we take a square full-rank matrix, hence $d_m = \infty$ meaning that the last code is the trivial code with just the all-zero codeword. Note that the d_i from $i = 1$ to $m-1$ form an affine progression between $4pk+1$ and $\frac{1}{2}k+1$ (not inclusive). For the parity-check matrices $H^{(i)}$ we define the corresponding distances to satisfy Lemma 4

$$\delta_i = \lfloor 4pn/(d_{i-1} - 1) \rfloor + 1, \quad i \in \{2, \dots, m\}. \quad (10)$$

Rate calculation: To calculate the compression rate of Construction 4 we use the simple formula in the next lemma.

Lemma 5. *The total number of bits output by the encoder of Construction 4 is*

$$\sum_{i=1}^m |\mathbf{p}^{(i)}| = r_1 t + \sum_{i=2}^m \tilde{r}_i \rho_i. \quad (11)$$

Proof: Immediate from item 3 in the encoder of Construction 4. r_i is the redundancy of the binary code $S^{(i)}$ with minimum distance d_i , specifically $r_m = k$, and recall the definition $\tilde{r}_i = r_i - r_{i-1}$. ρ_i is the redundancy of the $2^{\tilde{r}_i}$ -ary code $H^{(i)}$ with minimum distance δ_i . \blacksquare

To get the asymptotic compression rate achievable with Construction 4 we use the Gilbert-Varshamov bound for the binary codes $S^{(i)}$

$$r_i = kH \left(\frac{d_i}{k} \right), \quad (12)$$

and the Singleton bound for the $2^{\tilde{r}_i}$ -ary code $H^{(i)}$

$$\rho_i = \delta_i.$$

The Singleton bound is achievable, e.g. with Reed-Solomon codes, when $2^{\tilde{r}_i} \geq t$. Since every \tilde{r}_i grows linearly with k , for this condition to be met it is sufficient that k is at least logarithmic in $t = n/k$, for example when $k = \log n$. Now we get the compression rate:

Proposition 6. *For any constant integer m the compression rate of Construction 4, which is the total number of bits output by the encoder divided by n is*

$$H \left(4p + \frac{1}{m} \left(\frac{1}{2} - 4p \right) \right) + \sum_{i=2}^m \left[H \left(4p + \frac{i}{m} \left(\frac{1}{2} - 4p \right) \right) - H \left(4p + \frac{i-1}{m} \left(\frac{1}{2} - 4p \right) \right) \right] \cdot \frac{4p}{4p + \frac{i-1}{m} \left(\frac{1}{2} - 4p \right)}. \quad (13)$$

Proof: The expression in (13) is obtained by substituting in (11) the Gilbert Varshamov bound for r_i corresponding to d_i in (9), and the Singleton bound for ρ_i corresponding to δ_i in (10), then normalizing by n . \blacksquare

E. Realization and complexity

1) *Realization:* To realize Construction 4 efficiently, we reduce encoding and decoding operations to known operations from error-correcting codes. Because error-correcting codes are used in a substantially different way for compression, we next explain their adaptations in the concatenated scheme.

The function $\mathcal{E}^{(i)} : F_{2^{\tilde{r}_i}}^t \rightarrow F_{2^{\tilde{r}_i}}^{\rho_i}$ calculates the parity symbols of the code with parity-check matrix $[H^{(i)}, I_{\rho_i}]$, where $H^{(i)}$ is a parity-check matrix of a length- t code with minimum distance δ_i , given in systematic form. The code $[H^{(i)}, I_{\rho_i}]$ is a lengthened version of the code $H^{(i)}$. Note that this code is a poor error-correcting code, but works here (with better parameters) because there are no errors in the symbols of $\mathbf{p}^{(i)}$. Given a systematic encoder for the code defined by $H^{(i)}$ (for example a Reed-Solomon code), we can realize $\mathcal{E}^{(i)}$ by first encoding the first $t - \rho_i$ input symbols to a word of $H^{(i)}$, and then subtracting from the ρ_i parity symbols the remaining ρ_i inputs. This guarantees a 1-1 mapping from length- t input vectors to length- $(t + \rho_i)$ output vectors \mathbf{c} with $[H^{(i)}, I_{\rho_i}] \mathbf{c}^T = \mathbf{0}$.

The function $\mathcal{D}^{(i)} : F_{2^{\tilde{r}_i}}^t \times F_{2^{\tilde{r}_i}}^{\rho_i} \rightarrow F_{2^{\tilde{r}_i}}^t$ needs to find the vector \mathbf{x} nearest to \mathbf{a} that satisfies $H^{(i)} \mathbf{x}^T = \mathbf{b}$ (\mathbf{a}, \mathbf{b} are the first and second inputs to $\mathcal{D}^{(i)}$, respectively). \mathbf{a} is the vector of differential syndromes of the estimated $\hat{\mathbf{y}}_j$ sub-vectors; \mathbf{b} is the output of $\mathcal{E}^{(i)}$ that is available to the decoder without error. Given a syndrome decoder for the code $H^{(i)}$ (for example a Berlekamp-Massey Reed-Solomon decoder), $\mathcal{D}^{(i)}$ can be implemented by invoking the decoder on the syndrome $H^{(i)} \mathbf{a}^T - \mathbf{b}$, and subtracting the output minimal-weight error word from \mathbf{a} to obtain \mathbf{x} . This gives the desired output because we look for the minimal-weight \mathbf{e} such that $\mathbf{x} = \mathbf{a} - \mathbf{e}$ and $H^{(i)} \mathbf{x}^T = \mathbf{b}$, implying $H^{(i)} \mathbf{e}^T = H^{(i)} \mathbf{a}^T - \mathbf{b}$. Since \mathbf{b} is error-free, the correction capability of $\mathcal{D}^{(i)}$ is the same as that of the syndrome decoder operating on the code $H^{(i)}$.

Another function needed in Construction 4 appears in item 3 of its decoder: finding low-weight vectors with a given syndrome can be realized by known syndrome decoders for the codes $S^{(i)}$.

2) *Complexity*: Per the realizations above of the functions in Construction 4, we obtain the following encoding and decoding asymptotic complexities.

Decoding complexity: for the codes $H^{(i)}$ we take Reed-Solomon codes over a field of size t , which can be decoded with complexity $O(t \log^2 t)$ [24]. For the codes $S^{(i)}$ we take binary linear codes that can be decoded with complexity at most $k \cdot 2^{k/2}$ using the trellis representation of the code (it is known [25] that every linear block code can be represented by a trellis with at most $2^{\min(r_i, k-r_i)} \leq 2^{k/2}$ states in each coordinate). Now taking $k = O(\log n)$ we get the total complexity of $O(n^{1.5})$, because for each block we invoke $t = n/\log n$ binary trellis decoders with total complexity

$$O\left(\frac{n}{\log n} \sqrt{n} \log n\right) = O(n^{1.5}).$$

The complexity of the Reed-Solomon decoders is asymptotically negligible compared to $O(n^{1.5})$ because we invoke a constant number m of Reed-Solomon decoders, which give $O\left(\frac{n}{\log n} \log^2 \frac{n}{\log n}\right)$ operations over finite-field elements represented as size $\alpha \log n$ binary vectors (for some real $\alpha < 1$), giving in total not more than $O(n \log^3 n)$ bit operations.

Note that a construction using the standard generalized-concatenation half-minimum-distance decoder would have a higher complexity of $O(n^2 \log n)$ [13]. Because n^2 is considered prohibitive for long sequences, the decoder and parameters specified for Construction 4 give a more practical alternative for realization.

F. Improved construction

To reduce the overall compression rate of the scheme, we now propose an improvement of Construction 4 that still enjoys $O(n^{1.5})$ decoding complexity. The idea is that employing error-and-erasure decoding allows to set the correction parameters of the codes $S^{(i)}$ and $H^{(i)}$ such that less total redundancy is required. In the following improved scheme, we allow the decoder of $S^{(i)}$ to declare decoding failure when the distance of $\hat{\mathbf{y}}_j$, the closest vector to \mathbf{z}_j , is greater than $d_i/3$.

Construction 5. We repeat Construction 4, only changing the specification of $\mathcal{D}^{(i)}$. Define $\mathcal{D}^{(i)} : (F_{2^{\bar{r}_i}} \cup *)^t \times F_{2^{\bar{r}_i}}^{\rho_i} \rightarrow F_{2^{\bar{r}_i}}^t$ to be a decoder function with inputs \mathbf{a}, \mathbf{b} that finds a vector \mathbf{x} that satisfies: 1) $H^{(i)}\mathbf{x} = \mathbf{b}$, and 2) \mathbf{x} is nearest to \mathbf{a} on the subset of coordinates that are not $*$ in \mathbf{a} .

Encoder: same as Construction 4.

Decoder: Given an input row vector \mathbf{z} :

- 1) Partition $\mathbf{z} = [\mathbf{z}_1, \dots, \mathbf{z}_t]$.
- 2) Initialize $\hat{\mathbf{s}}_j^{(1)} := \mathbf{u}_j^{(1)}$, for each $j \in \{1, \dots, t\}$.

Iterate on $i = 2, \dots, m$ in 3-5 below:

- 3) For each j , find the lowest-weight vector \mathbf{v}_j such that $S^{(i-1)}\mathbf{v}_j^T = \hat{\mathbf{s}}_j^{(i-1)} + S^{(i-1)}\mathbf{z}_j^T$.
- 4) If the weight of \mathbf{v}_j is at most $(d_{i-1} - 1)/3$, calculate $\hat{\mathbf{y}}_j = \mathbf{z}_j + \mathbf{v}_j$ and take $\mathbf{a}_j := \tilde{S}^{(i)}\hat{\mathbf{y}}_j^T$; otherwise take $\mathbf{a}_j := *$.
Now calculate

$$\hat{\mathbf{u}}^{(i)} := \mathcal{D}^{(i)}(\mathbf{a}_1, \dots, \mathbf{a}_t; \mathbf{p}^{(i)}).$$

- 5) Concatenate $\hat{\mathbf{s}}_j^{(i)} := [\hat{\mathbf{s}}_j^{(i-1)}; \hat{\mathbf{u}}_j^{(i)}]$, for each $j \in \{1, \dots, t\}$.

Output: same as Construction 4.

Now we specify parameters for the codes $\{S^{(i)}\}_{i=1}^m$ and $\{H^{(i)}\}_{i=2}^m$ that are sufficient for guaranteed decoding with Construction 5. The following lemma is the modification of Lemma 4 to the improved construction.

Lemma 7. Let $d_H(\mathbf{z}, \mathbf{y}) \leq pn$, and take Construction 5 with parity-check matrices $\{S^{(i)}\}_{i=1}^m$ of binary codes with minimum distances $\{d_i\}_{i=1}^m$. Then correct decoding of $\mathbf{s}_j^{(m)}$ by $\hat{\mathbf{s}}_j^{(m)}$ is guaranteed if for each $i \in \{2, \dots, m\}$ we use a parity-check matrix $H^{(i)}$ of a code with minimum distance $\delta_i \geq 3pn/(d_{i-1} - 1)$.

Proof: Denote by τ_1 the number of indices $j \in \{1, \dots, t\}$ where $(d_{i-1} - 1)/3 < d_H(\mathbf{z}_j, \mathbf{y}_j) \leq 2(d_{i-1} - 1)/3$, and by τ_2 the number of indices where $d_H(\mathbf{z}_j, \mathbf{y}_j) > 2(d_{i-1} - 1)/3$. From the global distance constraint it is implied that $\tau_1 + 2\tau_2 < 3pn/(d_{i-1} - 1)$. The code $S^{(i-1)}$ has minimum distance d_{i-1} and can thus simultaneously correct up to $(d_{i-1} - 1)/3$ errors and detect up to $2(d_{i-1} - 1)/3$ errors. Hence the decoder $\mathcal{D}^{(i)}$ will see $\tau_e \leq \tau_2$ errors and $\tau_* = \tau_1 + \tau_2 - \tau_e$ erasures ($*$ symbols in Construction 5). It is observed that $\tau_* + 2\tau_e \leq \tau_1 + 2\tau_2 < 3pn/(d_{i-1} - 1)$, and hence minimum distance of $\delta_i \geq 3pn/(d_{i-1} - 1)$ is sufficient for the code $H^{(i)}$ to recover $\mathbf{u}_j^{(i)}$ and in turn $\mathbf{s}_j^{(i)}$ correctly. ■

Recall $n = kt$, and pick an integer m . For the matrix $S^{(i)}$ we specify the minimum distance

$$d_i = 3pk + \frac{i}{m} \left(\frac{1}{2} - 3p \right) k + 1, \quad i \in \{1, \dots, m-1\}, \quad (14)$$

and for $S^{(m)}$ we take a square full-rank matrix, hence $d_m = \infty$ meaning that the last code is the trivial code with just the all-zero codeword. Note that the d_i from $i = 1$ to $m - 1$ form an affine progression between $3pk + 1$ and $\frac{1}{2}k + 1$ (not inclusive). For the parity-check matrices $H^{(i)}$ we define the corresponding distances

$$\delta_i = \lceil 3pn / (d_{i-1} - 1) \rceil, \quad i \in \{2, \dots, m\}. \quad (15)$$

To get the asymptotic compression rate achievable with Construction 5 we adjust Proposition 6 to the d_i and δ_i of the improved construction.

Proposition 8. *For any constant integer m the compression rate of Construction 5, which is the total number of bits output by the encoder divided by n is*

$$H\left(3p + \frac{1}{m}\left(\frac{1}{2} - 3p\right)\right) + \sum_{i=2}^m \left[H\left(3p + \frac{i}{m}\left(\frac{1}{2} - 3p\right)\right) - H\left(3p + \frac{i-1}{m}\left(\frac{1}{2} - 3p\right)\right) \right] \cdot \frac{3p}{3p + \frac{i-1}{m}\left(\frac{1}{2} - 3p\right)}. \quad (16)$$

Proof: The expression in (16) is obtained by substituting in (11) the Gilbert Varshamov bound for r_i corresponding to d_i in (14), and the Singleton bound for ρ_i corresponding to δ_i in (15), then normalizing by n . ■

We plot in Fig. 1 the resulting compression rates of Construction 4 (dashed) and Construction 5 (solid), as a function of p , in the range $p \in [0, 2.5 \cdot 10^{-3}]$; the plots evaluate the expressions in (13), (16), respectively, with $m = 20000$. We do not compare these rates to the better rates of the basic Construction 3 ($H(2p)$ assuming error-correcting codes meeting the Gilbert Varshamov bound), because of its exponential decoding complexity. A more relevant comparison is with practical DNA compression algorithms, which currently give compression rates in the range $[0.1, 0.2]$ (where the lower rates are achieved by algorithms with reference at both the encoder and decoder) [18]. We conclude that for the range plotted in Fig. 1, Construction 5 gives rates competitive with the state-of-the-art in DNA compression, and without need to use a reference at the encoder.

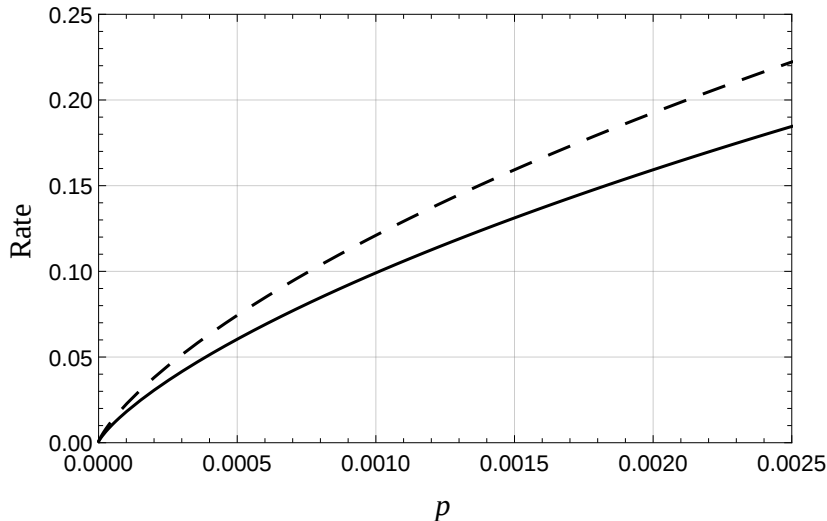


Fig. 1: Compression rates of Construction 4 (dashed) and Construction 5 (solid) as a function of the distance fraction p .

V. CONCLUSION

The first part of the paper refines the classical problem of compression with side information using a combinatorial characterization of the size-information vectors Z . In addition to the p -spread parameter investigated here, it is interesting in future work to study compressibility with respect to other characterizations of Z . For example, instead of the max in (1), one can characterize Z by the full *spectrum* of distances in Z . The second part of the paper develops a concatenated scheme for efficient guaranteed compression with Hamming-bounded side information. A natural future work is to extend the scheme to also allow side information with insertions and deletions. While for long blocks insertions and deletions are notoriously difficult to handle, the short inner codes of the concatenated scheme may enable an efficient solution.

VI. ACKNOWLEDGEMENT

We thank Neri Merhav for valuable discussions. We also thank the anonymous reviewers for valuable comments. This work was supported in part by the US-Israel Binational Science Foundation and in part by the Israel Science Foundation.

REFERENCES

- [1] C. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 9, pp. 379–423, Oct. 1948.
- [2] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973.
- [3] J. Ziv, "Fixed-rate encoding of individual sequences with side information," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 348–352, 1984.
- [4] A. Orlitsky and K. Viswanathan, "One-way communication and error-correcting codes," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1781–1788, 2003.
- [5] Y. Minsky, A. Trachtenberg, and R. Zippel, "Set reconciliation with nearly optimal communication complexity," *IEEE Transactions on Information Theory*, vol. 49, no. 9, pp. 2213–2218, 2003.
- [6] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate keys from biometrics and other noisy data," *SIAM J. Computing*, vol. 38, no. 1, pp. 97–139, 2008.
- [7] A. Wyner, "Recent results in the Shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, 1974.
- [8] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.
- [9] E. L. Blokh and V. V. Zyablov, *Generalized Concatenated Codes*. Moscow, Sviaz' (in Russian), 1976.
- [10] V. V. Zyablov, S. Shavgulidze, and M. Bossert, "An introduction to generalized concatenated codes," *European Transactions on Telecommunications*, vol. 10, no. 6, pp. 609–622, 1999.
- [11] V. V. Zyablov, "New interpretation of localization error codes, their error correcting capability and algorithms for decoding," *Transmission of Discrete Information over Channels with Clustered Errors (in Russian)*, pp. 8–17, 1972.
- [12] M. Bossert, *Channel Coding for Telecommunications*. Chichester, UK: Wiley, 1999.
- [13] V. V. Zyablov, "Decoding complexity and concatenated codes," *Coding and Complexity (G. Longo ed.)*, Springer-Verlag, pp. 131–162, 1975.
- [14] T. Uyematsu, "An algebraic construction of codes for Slepian-Wolf source networks," *IEEE Transactions on Information Theory*, vol. 47, no. 7, pp. 3082–3088, 2001.
- [15] A. Smith, "Scrambling adversarial errors using few random bits, optimal information reconciliation, and better private codes," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2007, pp. 395–404.
- [16] G. D. Forney, *Concatenated Codes*. Cambridge, MA: MIT Press, 1966.
- [17] D. Chumbalov and A. Romashchenko, "On the combinatorial version of the Slepian-Wolf problem," *IEEE Transactions on Information Theory*, vol. 64, no. 9, pp. 6054–6069, 2018.
- [18] J. Bonfield and M. Mahoney, "Compression of FASTQ and SAM format sequencing data," *PLOS ONE*, vol. 8, no. 3, p. e59190, 2013.
- [19] Y. Zhang, L. Li, Y. Yang, X. Yang, S. He, and Z. Zhu, "Light-weight reference-based compression of FASTQ data," *BMC Bioinformatics*, vol. 16, no. 188, 2015.
- [20] J. Carter and M. Wegman, "Universal classes of hash functions," *Journal of Computer and System Sciences*, vol. 18, pp. 143–154, 1979.
- [21] R. Ahlswede and L. Khachatrian, "The diametric theorem in Hamming spaces – optimal anticodes," *Advances in Applied Mathematics*, vol. 20, pp. 429–449, 1998.
- [22] T. Cover, "Enumerative source encoding," *IEEE Transactions on Information Theory*, vol. 19, no. 1, pp. 73–77, 1973.
- [23] G. D. Forney, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, 1966.
- [24] J. Justesen, "On the complexity of decoding Reed-Solomon codes," *IEEE Transactions on Information Theory*, vol. 22, no. 2, pp. 237–238, 1976.
- [25] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, no. 1, pp. 76–80, 1978.