Concatenated codes can achieve list-decoding capacity

Venkatesan Guruswami[*] Atri Rudra[†]

Abstract

We prove that binary linear concatenated codes with an outer algebraic code (specifically, a *folded* Reed-Solomon code) and *independently and randomly chosen* linear inner codes achieve the list-decoding capacity with high probability. In particular, for any $0 < \rho < 1/2$ and $\varepsilon > 0$, there exist concatenated codes of rate at least $1 - H(\rho) - \varepsilon$ that are (combinatorially) list-decodable up to a fraction ρ of errors. (The best possible rate, aka list-decoding capacity, for such codes is $1 - H(\rho)$, and is achieved by random codes.) A similar result, with better list size guarantees, holds when the outer code is also randomly chosen. Our methods and results extend to the case when the alphabet size is any fixed prime power $q \geqslant 2$.

Our result shows that despite the structural restriction imposed by code concatenation, the family of concatenated codes is rich enough to include capacity achieving list-decodable codes. This provides some encouraging news for tackling the problem of constructing *explicit* binary list-decodable codes with optimal rate, since code concatenation has been the preeminent method for constructing good codes over small alphabets.

1 Introduction

Concatenated Codes. Ever since its discovery and initial use by Forney [3], code concatenation has been a powerful tool for constructing error-correcting codes. At its core, the idea of code concatenation is really simple and natural. A concatenated code over a small alphabet, say a binary code for definiteness, is constructed in two steps. In the first step, the message is encoded via an error-correcting code C_1 over a large alphabet, say a large finite field $\mathbb{F}_{2^m}$. C_1 is referred to as the *outer* code. Each of the symbols of the resulting codeword of C_1 is then encoded via a binary code C_2 that has 2^m codewords (corresponding to the 2^m outer codeword symbols). The code C_2 is referred to as the *inner* code. The popularity of code concatenation arises due to the fact that is often difficult to give a direct construction of good (long) binary codes. On the other hand, over large alphabets, an array of powerful algebraic constructions (such as Reed-Solomon and algebraic-geometric codes) with excellent parameters are available. While the concatenated construction still requires an inner code that is binary, this is a small/short code with block length $O(m)$, which is typically logarithmic or smaller in the length of the outer code. A good choice for the inner code can therefore be found efficiently by a brute-force search, leading to a polynomial time construction of the final concatenated code.

This paradigm draws its power from the fact that a concatenated code, roughly speaking, inherits the good features of both the outer and inner codes. For example, the rate of the concatenated code is the product of the rates of the outer and inner codes, and the minimum distance is at least the product of the distances of the outer and inner codes. The alphabet of the concatenated code equals that of the inner code. Above, we assumed that all the inner codes were identical. This is not necessary and one can use different inner codes for encoding different symbols of the outer codeword. One way to leverage this is to use an explicit ensemble of inner codes most of which are "good." This was the idea behind Justesen's celebrated explicit construction of asymptotically good binary codes [10]. In this work, we will use random, i.i.d. choices for the different inner codes, and the independence of the inner encodings will be crucial in our analysis.

By concatenating an outer Reed-Solomon code of high rate with short inner codes achieving Shannon capacity (known to exist by a random coding argument), Forney [3] gave a construction of binary linear codes that achieve the capacity of the binary symmetric channel with a polynomial time decoding complexity. By using as outer code a linear time encodable/decodable code, one can make the encoding/decoding complexity linear in the block length [13]. In comparison, Shannon's nonconstructive proof of his capacity theorem used an exponential time maximum likelihood decoder.

The List Decoding Context. Our focus in this work is on the worst-case error model, with the goal being to recover from an arbitrary fraction ρ of errors with the best possible rate. In this setting, notions such

[*] University of Washington, Department of Computer Science and Engineering, Seattle, WA 98195, and (on leave at) Institute for Advanced Study, School of Mathematics, Princeton, NJ. Research supported by Sloan and Packard Fellowships, and NSF Career Award CCF-0343672. *Email*: venkat@cs.washington.edu

[†] Department of Computer Science and Engineering, University at Buffalo, State University of New York, Buffalo, NY 14260. *Email*: atri@cse.buffalo.edu. This work was done while the author was at the University of Washington and supported by NSF CCF-0343672.

as minimum distance and list decoding become central, and concatenated codes have been the key tool in many developments concerning these notions. (All the basic coding theory notions are formally defined in Sections 2.2-2.4.) In fact, for the longest time, till the work on expander codes by Sipser and Spielman [12], code concatenation schemes gave the *only known* explicit construction of a family of asymptotically good codes (i.e., with rate and relative distance both bounded away from zero as the block length grew). Even today, the best trade-offs between rate and distance for explicit codes are achieved by variants of concatenated codes; see [2] for further details.

Let us consider the problem of constructing a family of binary codes for correcting a fraction ρ of worst-case errors, for some $0 < \rho < 1/2$. For large n, there are about $2^{H(\rho)n}$ binary strings of weight ρn, where $H(\rho) = -\rho \log_2 \rho - (1-\rho)\log_2(1-\rho)$ is the binary entropy function. Therefore, when up to a ρ fraction of symbols can be corrupted, a transmitted codeword $\mathbf{c}$ can get distorted into any one of about $2^{H(\rho)n}$ possible received words. Since the decoder must be able to associate $\mathbf{c}$ with all such received words, it is easy to argue that there can be at most about $2^{(1-H(\rho))n}$ codewords. In other words, the rate of the code must be at most $1 - H(\rho)$.

Perhaps surprisingly, the above simplistic upper bound on rate is in fact accurate, and at least nonconstructively, a rate arbitrarily close to $1 - H(\rho)$ can be realized. In fact, with high probability a completely random code of rate $(1 - H(p) - \varepsilon)$, obtained by picking $2^{(1-H(p)-\varepsilon)n}$ codewords randomly and independently, has the property that every Hamming ball of radius ρn has at most $O(1/\varepsilon)$ codewords. One can thus use such a code to (list) decode from a fraction ρ of errors, where in the worst-case the decoder may output a list of $O(1/\varepsilon)$ answers. The trade-off $R = 1 - H(\rho)$ between the rate R and fraction of errors ρ is called the *list-decoding capacity.* The choice of binary alphabet in this discussion is only for definiteness. Over an alphabet size $q \geqslant 2$, the list-decoding capacity equals $1 - H_q(\rho)$, where $H_q(\cdot)$ is the q-ary entropy function.

Unfortunately, the above is a nonconstructive argument and the codes achieving list-decoding capacity are shown to exist by a random coding argument, and are not even succinctly, let alone explicitly, specified. It can also be shown that random *linear* codes achieve list-decoding capacity, though the known proofs only achieve a list size of $2^{O(1/\varepsilon)}$ when the rate is within ε of the list-decoding capacity.[1] The advantage with linear codes is that being subspaces they can be described succinctly by a basis for the subspace (called generator matrix in coding parlance). Yet, a generic linear code offers little in terms of algorithmically useful structure, and in general only brute-force decoders running in exponential time are known for such a code.

Turning to constructive results for list decoding, recently explicit codes approaching list-decoding capacity together with polynomial time list-decoding algorithms were constructed over *large* alphabets [6]. Using these as outer codes in a concatenation scheme led to polynomial time constructions of *binary* codes that achieved a rate vs. list-decoding radius trade-off called the *Zyablov* bound [6]. By using a multilevel generalization of code concatenation, the trade-off was recently improved to the so-called *Blokh-Zyablov* bound [8]. Still, these explicit constructions fall well short of achieving the list-decoding capacity for binary (and other small alphabet) codes, which remains a challenging open problem.

Given the almost exclusive stronghold of concatenated codes on progress in explicit constructions of list-decodable codes over small alphabets, the following natural question arises: *Do there exist concatenated codes that achieve list-decoding capacity, or does the stringent structural restriction imposed on the code by concatenation preclude achieving list-decoding capacity?*

The natural way to analyze the list-decoding performance of concatenated codes suggests that perhaps concatenation is too strong a structural bottleneck to yield optimal list-decodable codes. Such an analysis proceeds by decoding the blocks of the received word corresponding to various inner encodings, which results in a small set S_i of possible symbols for each position i of the outer code. One then argues that there cannot be too many outer codewords whose i'th symbol belongs to S_i for many positions i (this is called a "list recovery" bound).[2] Even assuming optimal, capacity-achieving bounds on the individual list-decodability of the outer and inner codes, the above "two-stage" analysis bottlenecks at the Zyablov bound.[3]

The weakness of the two-stage analysis is that it treats the different inner decodings independently, and fails to exploit the fact that the various inner blocks encode a structured set of symbols, namely those arising in a codeword of the outer code. Exploiting this

[1] For the case of binary alphabet alone, it is shown in [4], that a list size of $O(1/\varepsilon)$ suffices. But this result is not known to hold with high probability.

[2] When the outer code is algebraic such as Reed-Solomon or folded Reed-Solomon, the list recovery step admits an efficient algorithm which leads to a polynomial time list-decoding algorithm for the concatenated code, such as in [6, 8].

[3] One can squeeze out a little more out of the argument and achieve the Blokh-Zyablov bound, by exploiting the fact that subcodes of the inner codes, being of lower rate, can be list decoded to a larger radius [8].

and arguing that the structure of the outer codewords prevents many "bad" inner blocks from occurring simultaneously, and using this to get improved bounds, however, seems like an intricate task. In part this is because the current understanding of "bad list-decoding configurations," i.e., Hamming balls of small radius containing many codewords, for codes is rather poor.

Our Results. In this work, we prove that there exist binary (and q-ary for any fixed prime power q) linear concatenated codes that achieve list-decoding capacity for any desired rate. In fact, we prove that a random concatenated code drawn from a certain ensemble achieves capacity with overwhelming probability. This is encouraging news for the eventual goal of achieving list-decoding capacity (or at least, going beyond the above-mentioned Blokh-Zyablov bottleneck) over small alphabets with polynomial time decodable codes.

The outer codes in our construction are the folded Reed-Solomon codes which were shown in [6] to have near-optimal list-recoverability properties.[4] The inner codes for the various positions are random linear codes (which can even have a rate of 1), with a completely *independent* random choice for each outer codeword position. To get within ε of list decoding capacity, our result guarantees an output list size bound that is a large polynomial (greater than $N^{1/\varepsilon}$) in the block length N. We also prove that one can achieve capacity when a random linear code is chosen for the outer code; we get a better list size upper bound of a constant depending only on ε in this case. A corollary of our result is that one can construct binary codes achieving list-decoding capacity with a number of random bits that grows quasi-linearly in the block length, compared to the quadratic bound (achieved by a random linear code) known earlier.

Our results are inspired by results of Blokh and Zyablov [1] and Thommesen [14] showing the existence of binary concatenated codes whose rate vs. distance trade-off meets the Gilbert-Varshamov (GV) bound. We recall that the GV bound is the best known trade-off between rate and relative distance for binary (and q-ary for $q < 49$) codes and is achieved w.h.p. by random linear codes. Blokh and Zyablov show the result for independent random choices for the outer code and the various inner encodings. Thommesen establishes that one can fix the outer code to be a Reed-Solomon code and only pick the inner codes randomly (and independently). We give a high level overview of our proof and how it compares with Thommesen's proof in Section 3.

2 Preliminaries

For an integer $m \geqslant 1$, we will use $[m]$ to denote the set $\{1, \ldots, m\}$.

2.1 q-ary Entropy and Related Functions

Let $q \geqslant 2$ be an integer. $H_q(x) = x \log_q(q-1) - x \log_q x - (1-x)\log_q(1-x)$ will denote the q-ary entropy function. We will make use of the following property of this function.

Lemma 2.1. ([11]) *For every $0 \leqslant y \leqslant 1 - 1/q$ and for every small enough $\varepsilon > 0$, we have $H_q^{-1}(y - \varepsilon^2/c'_q) \geqslant H_q^{-1}(y) - \varepsilon$, where $c'_q \geqslant 1$ is a constant that depends only on q.*

For $0 \leqslant z \leqslant 1$ define

$$\alpha_q(z) = 1 - H_q(1 - q^{z-1}). \tag{2.1}$$

We will need the following property of the function above.

Lemma 2.2. *Let $q \geqslant 2$ be an integer. For every $0 \leqslant z \leqslant 1$, $\alpha_q(z) \leqslant z$.*

Proof. The proof follows from the subsequent sequence of relations:

$$\begin{aligned}\alpha_q(z) &= 1 - H_q(1 - q^{z-1}) \\ &= 1 - (1 - q^{z-1})\log_q(q-1) \\ &\qquad + (1 - q^{z-1})\log_q(1 - q^{z-1}) + q^{z-1}(z-1) \\ &= zq^{z-1} + (1 - q^{z-1})\left(1 - \log_q\left(\frac{q-1}{1-q^{z-1}}\right)\right) \\ &\leqslant z,\end{aligned}$$

where the last inequality follows from the facts that $q^{z-1} \leqslant 1$ and $1 - q^{z-1} \leqslant 1 - 1/q$, which implies that $\log_q\left(\frac{q-1}{1-q^{z-1}}\right) \geqslant 1$. □

We will also consider the following function

$$f_{x,q}(\theta) = (1-\theta)^{-1} \cdot H_q^{-1}(1 - \theta x),$$

where $0 \leqslant \theta, x \leqslant 1$. We will need the following property of this function, which was proven in [14] for the $q = 2$ case. The following is an easy extension of the result for general q (the proof can be found in [11, Chap. 5]).

Lemma 2.3. ([14]) *Let $q \geqslant 2$ be an integer. For any $x \geqslant 0$ and $0 \leqslant y \leqslant \alpha_q(x)/x$,*

$$\min_{0 \leqslant \theta \leqslant y} f_{x,q}(\theta) = (1-y)^{-1} H_q^{-1}(1 - xy).$$

[4] We note that the excellent list-recoverability of folded Reed-Solomon codes is crucial for our argument, and we do not know how to prove a similar result using just Reed-Solomon codes as outer codes.

2.2 Basic Coding Definitions A code of *dimension* k and *block length* n over an alphabet Σ is a subset of Σ^n of size $|\Sigma|^k$. The *rate* of such a code equals k/n. Each vector in C is called a codeword. In this paper, we will focus on the case when Σ is a finite field. We will denote by $\mathbb{F}_q$ the field with q elements. A code C over $\mathbb{F}_q$ is called a linear code if C is a subspace of $\mathbb{F}_q^n$. In this case the dimension of the code coincides with the dimension of C as a vector space over $\mathbb{F}_q$. By abuse of notation we will also think of a code C as a map from elements in $\mathbb{F}_q^k$ to their corresponding codeword in $\mathbb{F}_q^n$. If C is linear, this map is a linear transformation, mapping a row vector $x \in \mathbb{F}_q^k$ to a vector $xG \in \mathbb{F}_q^n$ for a $k \times n$ matrix G over $\mathbb{F}_q$ called the generator matrix.

The Hamming distance between two vectors in Σ^n is the number of places they differ in. The (minimum) distance of a code C is the minimum Hamming distance between any two pairs of distinct codewords from C. The relative distance is the ratio of the distance to the block length.

2.3 Code Concatenation Concatenated codes are constructed from two different kinds of codes that are defined over alphabets of different sizes. Say we are interested in a code over $\mathbb{F}_q$ (in this paper, we will always think of $q \geqslant 2$ as being a fixed constant). Then the *outer code* C_{out} is defined over $\mathbb{F}_Q$, where $Q = q^k$ for some positive integer k and has block length N. The second type of code, called the *inner codes*, which are denoted by $C_{in}^1, \ldots, C_{in}^N$ are defined over $\mathbb{F}_q$ and are each of dimension k (note that the message space of C_{in}^i for all i and the alphabet of C_{out} have the same size). The concatenated code, denoted by $C = C_{out} \circ (C_{in}^1, \ldots, C_{in}^N)$, is defined as follows. Let the rate of C_{out} be R and let the block lengths of C_{in}^i be n (for $1 \leqslant i \leqslant N$). Define $K = RN$ and $r = k/n$. The input to C is a vector $\mathbf{m} = \langle m_1, \ldots, m_K \rangle \in (\mathbb{F}_q^k)^K$. Let $C_{out}(\mathbf{m}) = \langle x_1, \ldots, x_N \rangle$. The codeword in C corresponding to $\mathbf{m}$ is defined as follows

$$C(\mathbf{m}) = \langle C_{in}^1(x_1), C_{in}^2(x_2), \ldots, C_{in}^N(x_N) \rangle.$$

The outer code C_{out} will either be a random linear code over $\mathbb{F}_Q$ or the folded Reed-Solomon code from [6]. In the case when C_{out} is random, we will pick C_{out} by selecting $K = RN$ vectors uniformly at random from $\mathbb{F}_Q^N$ to form the rows of the generator matrix. For every position $1 \leqslant i \leqslant N$, we will choose an inner code C_{in}^i to be a random linear code over $\mathbb{F}_q$ of block length n and rate $r = k/n$. In particular, we will work with the corresponding generator matrices $\mathbf{G}_i$, where every $\mathbf{G}_i$ is a random $k \times n$ matrix over $\mathbb{F}_q$. All the generator matrices $\mathbf{G}_i$ (as well as the generator matrix for C_{out}, when we choose a random C_{out}) are chosen independently. This fact will be used crucially in our proofs.

Given the outer code C_{out} and the inner codes C_{in}^i, recall that for every codeword $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_N) \in C_{out}$, the codeword $\mathbf{uG} \overset{def}{=} (\mathbf{u}_1\mathbf{G}_1, \mathbf{u}_2\mathbf{G}_2, \ldots, \mathbf{u}_N\mathbf{G}_N)$ is in $C = C_{out} \circ (C_{in}^1, \ldots, C_{in}^N)$, where the operations are over $\mathbb{F}_q$.

We will need the following notions of the weight of a vector. Given a vector $\mathbf{v} \in \mathbb{F}_q^{nN}$, its Hamming weight is denoted by $wt(\mathbf{v})$. Given a vector $\mathbf{y} = (y_1, \ldots, y_N) \in (\mathbb{F}_q^n)^N$ and a subset $S \subseteq [N]$, we will use $wt_S(\mathbf{y})$ to denote the Hamming weight over $\mathbb{F}_q$ of the subvector $(y_i)_{i \in S}$. Note that $wt(\mathbf{y}) = wt_{[N]}(y)$.

We will need the following simple lemma due to Thommesen, which is stated in a slightly different form in [14]. For the sake of completeness we also present its proof.

LEMMA 2.4. ([14]) *Given a fixed outer code C_{out} of block length N and an ensemble of random inner linear codes of block length n given by generator matrices $\mathbf{G}_1, \ldots, \mathbf{G}_N$ the following is true. Let $\mathbf{y} \in \mathbb{F}_q^{nN}$. For any codeword $\mathbf{u} \in C_{out}$, any non-empty subset $S \subseteq [N]$ such that $\mathbf{u}_i \neq 0$ for all $i \in S$ and any integer $h \leqslant n|S| \cdot \left(1 - \frac{1}{q}\right)$:*

$$\Pr[wt_S(\mathbf{uG} - \mathbf{y}) \leqslant h] \leqslant q^{-n|S|\left(1 - H_q\left(\frac{h}{n|S|}\right)\right)},$$

where the probability is taken over the random choices of $\mathbf{G}_1, \ldots, \mathbf{G}_N$.

Proof. Let $|S| = s$ and w.l.o.g. assume that $S = [s]$. As the choices for $\mathbf{G}_1, \ldots, \mathbf{G}_N$ are made independently, it is enough to show that the claimed probability holds for the random choices for $\mathbf{G}_1, \ldots, \mathbf{G}_s$. For any $1 \leqslant i \leqslant s$ and any $y \in \mathbb{F}_q^n$, since $\mathbf{u}_i \neq 0$, we have $\Pr_{\mathbf{G}_i}[\mathbf{u}_i\mathbf{G}_i = y] = q^{-n}$. Further, these probabilities are independent for every i. Thus, for any $\mathbf{y} = \langle y_1, \ldots, y_s \rangle \in (\mathbb{F}_q^n)^s$, $\Pr_{\mathbf{G}_1,\ldots,\mathbf{G}_s}[\mathbf{u}_i\mathbf{G}_i = y_i \text{ for every } 1 \leqslant i \leqslant s] = q^{-ns}$. This implies that:

$$\Pr_{\mathbf{G}_1,\ldots,\mathbf{G}_s}[wt_S(\mathbf{uG} - \mathbf{y}) \leqslant h] = q^{-ns} \sum_{j=0}^{h} \binom{ns}{j} (q-1)^j.$$

The claimed result follows from the following well known inequality for $h/(ns) \leqslant 1 - 1/q$ ([9]):

$$\sum_{j=0}^{h} \binom{ns}{j} (q-1)^j \leqslant q^{nsH_q\left(\frac{h}{ns}\right)}.$$

□

2.4 List Decoding and List Recovery

DEFINITION 2.1. (LIST DECODABLE CODE) *For $0 < \rho < 1$ and an integer $L \geqslant 1$, a code $C \subseteq \mathbb{F}_q^n$ is said to be (ρ, L)-list decodable if for every $y \in \mathbb{F}_q^n$, the number of codewords in C that are within Hamming distance ρn from y is at most L.*

We will also crucially use a generalization of list decoding called list recovery, a term first coined in [5] even though the notion had existed before. List recovery has been extremely useful in list-decoding concatenated codes. The input for list recovery is not a sequence of symbols but rather a sequence of subsets of allowed codeword symbols, one for each codeword position.

DEFINITION 2.2. (LIST RECOVERABLE CODE) *A code $C \subseteq \mathbb{F}_q^n$, is called (ρ, ℓ, L)-list recoverable if for every sequence of sets $S_1, S_2, \ldots, S_n$, where $S_i \subseteq \mathbb{F}_q$ and $|S_i| \leqslant \ell$ for every $1 \leqslant i \leqslant n$, there are at most L codewords in $c \in C$ such that $c_i \in S_i$ for at least $(1 - \rho)n$ positions i.*

The classical family of Reed-Solomon (RS) codes over a field $\mathbb{F}$ are defined to be the evaluations of low-degree polynomials at a sequence of distinct points of $\mathbb{F}$. Folded Reed-Solomon codes are obtained by viewing the RS code as a code over a larger alphabet $\mathbb{F}^s$ by bundling together consecutive s symbols for some folding parameter s. We will not need any specifics of folded RS codes (in fact even its definition) beyond (i) the strong list recovery property guaranteed by the following theorem from [6], and (ii) the fact that specifying any $K + 1$ positions in a dimension K folded Reed-Solomon code suffices to identify the codeword (equivalently, a dimension K and length N folded RS code has distance at least $N - K$).

THEOREM 2.1. ([6]) *For every integer $\ell \geqslant 1$, for all constants $\varepsilon > 0$, for all $0 < R < 1$, and for every prime p, there is an* explicit *family of folded Reed-Solomon codes, over fields of characteristic p that have rate at least R and which can be $(1 - R - \varepsilon, \ell, L(N))$-list recovered in polynomial time, where for codes of block length N, $L(N) = (N/\varepsilon^2)^{O(\varepsilon^{-1}\log(\ell/R))}$ and the code is defined over alphabet of size $(N/\varepsilon^2)^{O(\varepsilon^{-2}\log \ell/(1-R))}$.*

3 Overview of the Proof

Our proofs are inspired by Thommesen's proof [14] of the following result concerning the rate vs. distance trade-off of concatenated codes: Binary linear concatenated codes with an outer Reed-Solomon code and independently and randomly chosen inner codes meet the Gilbert-Varshamov bound with high probability[5], provided a moderate condition on the outer and inner rates is met. Given that our proof builds on the proof of Thommesen, we start out by reviewing the main ideas in his proof.

The outer code C_{out} in [14] is a Reed-Solomon code of length N and rate R (over $\mathbb{F}_Q$ where $Q = q^k$ for some integer $k \geqslant 1$). The inner linear codes (over $\mathbb{F}_q$) are generated by N randomly chosen $k \times n$ generator matrices $\mathbf{G} = (\mathbf{G}_1, \ldots, \mathbf{G}_N)$, where $r = k/n$. Note that since the final code will be linear, to show that with high probability the concatenated code will have distance close to $H_q^{-1}(1 - rR)$, it is enough to show that the probability of the Hamming weight of $\mathbf{uG}$ over $\mathbb{F}_q$ being at most $(H_q^{-1}(1 - rR) - \varepsilon)nN$ (for some Reed-Solomon codeword $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$ and $\varepsilon > 0$), is small. Fix a codeword $\mathbf{u} \in C_{out}$. Now note that if for some $1 \leqslant i \leqslant N$, $\mathbf{u}_i = 0$, then for every choice of $\mathbf{G}_i$, $\mathbf{u}_i\mathbf{G}_i = 0$. Thus, only the non-zero symbols of $\mathbf{u}$ contribute to $wt(\mathbf{uG})$. Further, for a non-zero $\mathbf{u}_i$, $\mathbf{u}_i\mathbf{G}_i$ takes all the values in $\mathbb{F}_q^n$ with equal probability over the random choices of $\mathbf{G}_i$. Since the choice of the $\mathbf{G}_i$'s are independent, this implies that $\mathbf{uG}$ takes each of the possible $q^{n\cdot wt(\mathbf{u})}$ values in $\mathbb{F}_q^{nN}$ with the same probability. Thus, the total probability that $\mathbf{uG}$ has a Hamming weight of at most h is $\sum_{w=0}^{h} \binom{n\cdot wt(\mathbf{u})}{w} q^{-n\cdot wt(\mathbf{u})} \leqslant q^{-n\cdot wt(\mathbf{u})\left(1 - H_q\left(\frac{h}{n\cdot wt(\mathbf{u})}\right)\right)}$. The rest of the argument follows by doing a careful union bound of this probability for all non zero codewords in C_{out}, using the weight distribution of the RS code. This step imposes an upper bound on the outer rate R (specifically, $R \leqslant \alpha_q(r)/r$), but still offers enough flexibility to achieve any desired value in $(0, 1)$ for the overall rate rR (even with the choice $r = 1$, i.e., when the inner encodings don't add any redundancy).

Let us now try to extend the idea above to show a similar result for list decoding. We want to show that for any Hamming ball of radius at most $h = (H_q^{-1}(1 - rR) - \varepsilon)nN$ has at most L codewords from the concatenated code C (assuming we want to show that L is the worst case list size). To show this let us look at a set of $L + 1$ codewords from C and try to prove that the probability that all of them lie within some fixed ball $\mathcal{B}$ of radius h is small. Let $\mathbf{u}^1, \ldots, \mathbf{u}^{L+1}$ be the corresponding codewords in C_{out}. Extending Thommesen's proof would be straightforward if the events corresponding to $\mathbf{u}^j\mathbf{G}$ belonging to the ball $\mathcal{B}$ for various $1 \leqslant j \leqslant L + 1$ were independent. In particular, if we can show that for every position $1 \leqslant i \leqslant N$, all the non-zero symbols in $\{\mathbf{u}_i^1, \mathbf{u}_i^2, \ldots, \mathbf{u}_i^{L+1}\}$ are linearly independent over $\mathbb{F}_q$ then the generalization of Thommesen's proof is immediate.

Unfortunately, the notion of independence discussed above does *not* hold for every $L + 1$ tuple of codewords from C_{out}. The natural way to get inde-

[5] A q-ary code of rate $\mathcal{R}$ meets the Gilbert-Varshamov bound if it has relative distance at least $H_q^{-1}(1 - \mathcal{R})$.

pendence when dealing with linear codes is to look at messages that are linearly independent. It turns out that if C_{out} is also a *random linear code* over $\mathbb{F}_Q$ then we have a good approximation of the the notion of independence above. Specifically, we show that with very high probability for a linearly independent (over $\mathbb{F}_Q$) set of messages[6] $\mathbf{m}^1, \ldots, \mathbf{m}^{L+1}$, the set of codewords $\mathbf{u}^1 = C_{out}(\mathbf{m}^1), \ldots, \mathbf{u}^{L+1} = C_{out}(\mathbf{m}^{L+1})$ have the following approximate independence property. For many positions $1 \leqslant i \leqslant N$, many non-zero symbols in $\{\mathbf{u}_i^1, \ldots, \mathbf{u}_i^{L+1}\}$ are linearly independent over $\mathbb{F}_q$. It turns out that this approximate notion of independence is enough for Thommesen's proof to go through.

We remark that the notion above crucially uses the fact that the outer code is a random linear code. The argument gets more tricky when C_{out} is fixed to be (say) the Reed-Solomon code. Now even if the messages $\mathbf{m}^1, \ldots, \mathbf{m}^{L+1}$ are linearly independent it is not clear that the corresponding codewords will satisfy the notion of independence in the above paragraph. Interestingly, we can show that this notion of independence is equivalent to showing good list recoverability properties for C_{out}. Reed-Solomon codes are however not known to have optimal list recoverability (which is what is required in our case). In fact, the results in [7] show that this is *impossible* for Reed-Solomon codes in general. However, folded RS codes *do* have near-optimal list recoverability and we exploit this in our proof.

4 Using Folded Reed-Solomon Code as Outer Code

In this section, we will prove that concatenated codes with the outer code being the folded Reed-Solomon code from [6] and using random and independent inner codes can achieve list-decoding capacity. The proof will make crucial use of the list recoverability of the outer code as stated in Theorem 2.1.

4.1 Linear Independence from List Recoverability

Definition 4.1. (Independent tuples) *Let C be a code of block length N and rate R defined over $\mathbb{F}_{q^k}$. Let $J \geqslant 1$ and $0 \leqslant d_1, \ldots, d_J \leqslant N$ be integers. Let $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$. An ordered tuple of codewords $(c^1, \ldots, c^J)$, $c^j \in C$ is said to be $(\mathbf{d}, \mathbb{F}_q)$-independent if the following holds. $d_1 = wt(c^1)$ and for every $1 < j \leqslant J$, d_j is the number of positions i such that c_i^j is $\mathbb{F}_q$-independent of the vectors $\{c_i^1, \ldots, c_i^{j-1}\}$, where $c^\ell = (c_1^\ell, \ldots, c_N^\ell)$.*

Note that for any tuple of codewords $(c^1, \ldots, c^J)$ there exists a unique $\mathbf{d}$ such that it is $(\mathbf{d}, \mathbb{F}_q)$-independent. The next result will be crucial in our proof.

Lemma 4.1. *Let C be a folded Reed-Solomon code of block length N that is defined over $\mathbb{F}_Q$ with $Q = q^k$ as guaranteed by Theorem 2.1. For any L-tuple of codewords from C, where $L \geqslant J \cdot (N/\varepsilon^2)^{O\left(\varepsilon^{-1} J \log(q/R)\right)}$ (where $\varepsilon > 0$ is same as the one in Theorem 2.1), there exists a sub-tuple of J codewords such that the J-tuple is $(\mathbf{d}, \mathbb{F}_q)$-independent, where $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$ such that for every $1 \leqslant j \leqslant J$, $d_j \geqslant (1 - R - \varepsilon)N$.*

Proof. The proof is constructive. In particular, given an L-tuple of codewords, we will construct a J sub-tuple with the required property. The correctness of the procedure will hinge on the list recoverability of the folded Reed-Solomon code as guaranteed by Theorem 2.1.

We will construct the final sub-tuple iteratively. In the first step, pick any non-zero codeword in the L-tuple– call it c^1. As C has distance at least $(1 - R)N$ (and $\mathbf{0} \in C$), c^1 is non-zero in at least $d_1 \geqslant (1 - R)N > (1 - R - \varepsilon)N$ many places. Note that c^1 is vacuously independent of the "previous" codewords in these positions. Now, say that the procedure has chosen codewords $c^1, \ldots, c^s$ such that the tuple is $(\mathbf{d}', \mathbb{F}_q)$-independent for $\mathbf{d}' = \langle d_1, \ldots, d_s \rangle$, where for every $1 \leqslant j \leqslant s$, $d_j \geqslant (1-R-\varepsilon)N$. For every $1 \leqslant i \leqslant N$, define S_i to be the $\mathbb{F}_q$-span of the vectors $\{c_i^1, \ldots, c_i^s\}$ in $\mathbb{F}_q^k$. Note that $|S_i| \leqslant q^s$. Call $c = (c_1, \ldots, c_N) \in C$ to be a *bad* codeword, if there does not exist any $d_{s+1} \geqslant (1-R-\varepsilon)N$ such that $(c^1, \ldots, c^s, c)$ is $(\mathbf{d}, \mathbb{F}_q)$-independent for $\mathbf{d} = \langle d_1, \ldots, d_{s+1} \rangle$. In other words, c is a bad codeword if and only if some $T \subset [N]$ with $|T| = (R + \varepsilon)N$ satisfies $c_i \in S_i$ for every $i \in T$. Put differently, c satisfies the condition of being in the output list for list recovering C with input $S_1, \ldots, S_N$ and agreement fraction $R + \varepsilon$. Thus, by Theorem 2.1, the number of such bad codewords is $U = (N/\varepsilon^2)^{O\left(\varepsilon^{-1} s \log(q/R)\right)} \leqslant (N/\varepsilon^2)^{O\left(\varepsilon^{-1} J \log(q/R)\right)}$, where J is the number of steps for which this greedy procedure can be applied. Thus, as long as at each step there are strictly more than U codewords from the original L-tuple of codewords left, we can continue this greedy procedure. Note that we can continue this procedure J times, as long as $J \leqslant L/U$. □

Finally, we will need the following bound on the number of independent tuples for folded Reed-Solomon codes. Its proof follows from the fact that a codeword in a dimension K folded RS code is completely determined once values at $K + 1$ of its positions are fixed.

[6] Again any set of $L + 1$ messages need not be linearly independent. However, it is easy to see that some subset of $J = \lceil \log_Q(L + 1) \rceil$ of messages are indeed linearly independent. Hence, we can continue the argument by replacing $L + 1$ with J.

LEMMA 4.2. *Let C be a folded Reed-Solomon code of block length N and rate $0 < R < 1$ that is defined over $\mathbb{F}_Q$, where $Q = q^k$. Let $J \geqslant 1$ and $0 \leqslant d_1, \ldots, d_J \leqslant N$ be integers and define $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$. Then the number of $(\mathbf{d}, \mathbb{F}_q)$-independent tuples in C is at most $q^{NJ(J+1)} \prod_{j=1}^{J} Q^{\max(d_j - N(1-R)+1,0)}$.*

Proof. Given a tuple $(c^1, \ldots, c^J)$ that is $(\mathbf{d}, \mathbb{F}_q)$-independent, define $T_j \subseteq [N]$ with $|T_j| = d_j$, for $1 \leqslant j \leqslant J$ to be the set of positions i, where c_i^j is $\mathbb{F}_q$-independent of $\{c_i^1, \ldots, c_i^{j-1}\}$. We will estimate the number of $(\mathbf{d}, \mathbb{F}_q)$-independent tuples by first estimating a bound U_j on the number of choices for the j^{th} codeword in the tuple (given a fixed choice of the first $j-1$ codewords). To complete the proof, we will show that

$$U_j \leqslant q^{N(J+1)} \cdot Q^{\max(d_j - N(1-R)+1,0)}.$$

A codeword $c \in C$ can be the j^{th} codeword in the tuple in the following way. For every position in $[N] \backslash T_j$, c can take at most $q^{j-1} \leqslant q^J$ values (as in these position the value has to lie in the $\mathbb{F}_q$ span of the values of the first $j-1$ codewords in that position). Since C is folded Reed-Solomon, once we fix the values at positions in $[N] \setminus T_j$, the codeword will be completely determined once any $\max(RN - (N - d_j) + 1, 0) = \max(d_j - N(1-R) + 1, 0)$ positions in T_j are chosen (w.l.o.g. assume that they are the "first" so many positions). The number of choices for T_j is $\binom{N}{d_j} \leqslant 2^N \leqslant q^N$. Thus, we have

$$\begin{aligned} U_j &\leqslant q^N \cdot (q^J)^{N-d_j} \cdot Q^{\max(d_j - N(1-R)+1,0)} \\ &\leqslant q^{N(J+1)} \cdot Q^{\max(d_j - N(1-R)+1),0)}, \end{aligned}$$

as desired. □

4.2 The Main Result

THEOREM 4.1. (MAIN) *Let q be a prime power and let $0 < r \leqslant 1$ be an arbitrary rational. Let $0 < \varepsilon < \alpha_q(r)$ an arbitrary real, where $\alpha_q(r)$ is as defined in (2.1), and $0 < R \leqslant (\alpha_q(r) - \varepsilon)/r$ be a rational. Then the following holds for large enough integers n, N such that there exist integers k and K that satisfy $k = rn$ and $K = RN$. Let C_{out} be a folded Reed-Solomon code over $\mathbb{F}_{q^k}$ of block length N and rate R. Let $C_{in}^1, \ldots, C_{in}^N$ be random linear codes over $\mathbb{F}_q$, where C_{in}^i is generated by a random $k \times n$ matrix $\mathbf{G}_i$ over $\mathbb{F}_q$ and the random choices for $\mathbf{G}_1, \ldots, \mathbf{G}_N$ are all independent.*[7] *Then the concatenated code $C = C_{out} \circ (C_{in}^1, \ldots, C_{in}^N)$ is a $\left(H_q^{-1}(1 - Rr) - \varepsilon, \left(\frac{N}{\varepsilon^2}\right)^{O\left(\varepsilon^{-4}(1-R)^{-2}\log(1/R)\right)}\right)$-list decodable code with probability at least $1 - q^{-\Omega(nN)}$ over the choices of $\mathbf{G}_1, \ldots, \mathbf{G}_N$. Further, C has rate rR w.h.p.*

REMARK 4.1. *For any desired rate $R^* \in (0, 1-\varepsilon)$ for the final concatenated code (here $\varepsilon > 0$ is arbitrary), one can pick the outer and inner rates R, r such that $Rr = R^*$ while also satisfying $R \leqslant (\alpha_q(r) - \varepsilon)/r$. In fact we can pick $r = 1$ and $R = R^*$ so that the inner encodings are linear transformations specified by random $k \times k$ matrices and do not add any redundancy.*

The rest of this section is devoted to proving Theorem 4.1.

Define $Q = q^k$. Let L be the worst-case list size that we are shooting for (we will fix its value at the end). By Lemma 4.1, any $L+1$-tuple of C_{out} codewords $(\mathbf{u}^0, \ldots, \mathbf{u}^L) \in (C_{out})^{L+1}$ contains at least $J = \left\lfloor (L+1)/(N/\gamma^2)^{O\left(\gamma^{-1} J \log(q/R)\right)} \right\rfloor$ codewords that form an $(\mathbf{d}, \mathbb{F}_q)$-independent tuple, for some $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$, with $d_j \geqslant (1 - R - \gamma)N$ (we will specify γ, $0 < \gamma < 1 - R$, later). Thus, to prove the theorem it suffices to show that with high probability, no Hamming ball in $\mathbb{F}_q^{nN}$ of radius $(H_q^{-1}(1 - rR) - \varepsilon)nN$ contains a J-tuple of codewords $(\mathbf{u}^1\mathbf{G}, \ldots, \mathbf{u}^J\mathbf{G})$, where $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ is a J-tuple of folded Reed-Solomon codewords that is $(\mathbf{d}, \mathbb{F}_q)$-independent. For the rest of the proof, we will call a J-tuple of C_{out} codewords $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ a *good* tuple if it is $(\mathbf{d}, \mathbb{F}_q)$-independent for some $\mathbf{d} = \langle d_1, \ldots, d_J \rangle$, where $d_j \geqslant (1 - R - \gamma)N$ for every $1 \leqslant j \leqslant J$.

Define $\rho = H_q^{-1}(1 - Rr) - \varepsilon$. For every good J-tuple of C_{out} codewords $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ and received word $\mathbf{y} \in \mathbb{F}_q^{nN}$, define an indicator variable $\mathbf{I}(\mathbf{y}, \mathbf{u}^1, \ldots, \mathbf{u}^J)$ as follows. $\mathbf{I}(\mathbf{y}, \mathbf{u}^1, \ldots, \mathbf{u}^J) = 1$ if and only if for every $1 \leqslant j \leqslant J$, $wt(\mathbf{u}^j\mathbf{G} - \mathbf{y}) \leqslant \rho n N$. That is, it captures the bad event that we want to avoid. Define

$$X_C = \sum_{\mathbf{y} \in \mathbb{F}_q^{nN}} \sum_{\text{good } (\mathbf{u}^1, \ldots, \mathbf{u}^J) \in (C_{out})^J} \mathbf{I}(\mathbf{y}, \mathbf{u}^1, \ldots, \mathbf{u}^J).$$

We want to show that with high probability $X_C = 0$. By Markov's inequality, the theorem would follow if we can show that:

$$\begin{aligned} \mathbb{E}[X_C] &= \sum_{\mathbf{y} \in \mathbb{F}_q^{nN}} \sum_{\substack{\text{good } (\mathbf{u}^1, \ldots, \mathbf{u}^J) \\ \in (C_{out})^J}} \mathbb{E}[\mathbf{I}(\mathbf{y}, \mathbf{u}^1, \ldots, \mathbf{u}^J)] \\ &\leqslant q^{-\Omega(nN)}. \qquad (4.2) \end{aligned}$$

Before we proceed, we need a final bit of notation. For a good tuple $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ and every $1 \leqslant j \leqslant J$, define $T_j(\mathbf{u}^1, \ldots, \mathbf{u}^J) \subseteq [N]$ to be the set of positions i such that $\mathbf{u}_i^j$ is $\mathbb{F}_q$-independent of the set $\{\mathbf{u}_i^1, \ldots, \mathbf{u}_i^{j-1}\}$. Note that since the tuple is good, $|T_j(\mathbf{u}^1, \ldots, \mathbf{u}^J)| \geqslant (1 - R - \gamma)N$.

Let $h = \rho n N$. Consider the following sequence of inequalities (where below we have suppressed the

[7] We stress that we do *not* require that the $\mathbf{G}_i$'s have rank k.

dependence of T_j on $(\mathbf{u}^1, \ldots, \mathbf{u}^J)$ for clarity):

$$\mathbb{E}[X_C] = \sum_{\mathbf{y}\in\mathbb{F}_q^{nN}} \sum_{\substack{\text{good}\\ (\mathbf{u}^1,..,\mathbf{u}^J)\\ \in(C_{out})^J}} \Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J} wt(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right] \tag{4.3}$$

$$\leqslant \sum_{\mathbf{y}\in\mathbb{F}_q^{nN}} \sum_{\substack{\text{good}\\ (\mathbf{u}^1,..,\mathbf{u}^J)\\ \in(C_{out})^J}} \Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right] \tag{4.4}$$

$$= \sum_{\mathbf{y}\in\mathbb{F}_q^{nN}} \sum_{\substack{\text{good}\\ (\mathbf{u}^1,..,\mathbf{u}^J)\\ \in(C_{out})^J}} \prod_{j=1}^{J} \Pr_{\mathbf{G}}\left[wt_{T_j}(\mathbf{u}^i\mathbf{G}-\mathbf{y}) \leqslant h\right] \tag{4.5}$$

In the above (4.3) follows from the definition of the indicator variable. (4.4) follows from the simple fact that for every vector $\mathbf{u}$ of length N and every $T \subseteq [N]$, $wt_T(\mathbf{u}) \leqslant wt(\mathbf{u})$. (4.5) follows from the subsequent argument. By definition of conditional probability, $\Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right]$ is the same as $\Pr_{\mathbf{G}}\left[wt_{T_J}(\mathbf{u}^J\mathbf{G}-\mathbf{y}) \leqslant h \middle| \bigwedge_{j=1}^{J-1} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right] \cdot \Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J-1} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right]$. Now as all symbols corresponding to T_J are good symbols, for every $i \in T_J$, the value of $\mathbf{u}_i^J\mathbf{G}_i$ is independent of the values of $\{\mathbf{u}_i^1\mathbf{G}_i, \ldots, \mathbf{u}_i^{J-1}\mathbf{G}_i\}$. Further since each of $\mathbf{G}_1, \ldots, \mathbf{G}_N$ are chosen independently (at random), the event $wt_{T_J}(\mathbf{u}^J\mathbf{G}-\mathbf{y}) \leqslant h$ is independent of the event $\bigwedge_{j=1}^{J-1} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h$. Thus,

$$\Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right] =$$
$$\Pr_{\mathbf{G}}\left[wt_{T_J}(\mathbf{u}^J\mathbf{G}-\mathbf{y}) \leqslant h\right] \Pr_{\mathbf{G}}\left[\bigwedge_{j=1}^{J-1} wt_{T_j}(\mathbf{u}^j\mathbf{G}-\mathbf{y}) \leqslant h\right]$$

Inductively applying the argument above gives (4.5). Further (where below we use $\overline{D}$ to denote $(1-R-\gamma)N$),

$$\mathbb{E}[X_C] \leqslant \sum_{\mathbf{y}\in\mathbb{F}_q^{nN}} \sum_{\substack{\text{good}\\ (\mathbf{u}^1,\ldots,\mathbf{u}^J)\\ \in(C_{out})^J}} \prod_{j=1}^{J} q^{-n|T_j|\left(1-H_q\left(\frac{h}{n|T_j|}\right)\right)} \tag{4.6}$$

$$= \sum_{\mathbf{y}\in\mathbb{F}_q^{nN}} \sum_{\substack{(d_1,..,d_J)\in\\ \{\overline{D},..,N\}^J}} \sum_{\substack{\text{good}\\ (\mathbf{u}^1,..,\mathbf{u}^J)\\ \in(C_{out})^J,\\ |T_1|=d_1,..,\\ |T_J|=d_J}} \prod_{j=1}^{J} q^{-nd_j+nd_jH_q\left(\frac{h}{nd_j}\right)} \tag{4.7}$$

$$\leqslant \sum_{\substack{(d_1,\ldots,d_J)\in\\ \{\overline{D},\ldots,N\}^J}} q^{nN} \cdot q^{NJ(J+1)} \cdot \prod_{j=1}^{J} Q^{\max(d_j-(1-R)N+1,0)} \prod_{j=1}^{J} q^{-nd_j\left(1-H_q\left(\frac{h}{nd_j}\right)\right)} \tag{4.8}$$

$$\leqslant \sum_{\substack{(d_1,\ldots,d_J)\in\\ \{\overline{D},\ldots,N\}^J}} q^{nN} \cdot q^{NJ(J+1)} \cdot \prod_{j=1}^{J} Q^{d_j-(1-R-\gamma)N} \prod_{j=1}^{J} q^{-nd_j\left(1-H_q\left(\frac{h}{nd_j}\right)\right)} \tag{4.9}$$

$$= \sum_{\substack{(d_1,\ldots,d_J)\in\\ \{\overline{D},\ldots,N\}^J}} \prod_{j=1}^{J} q^{-nd_jE}, \tag{4.10}$$

where E is the expression

$$1 - H_q\left(\frac{h}{nd_j}\right) - r\left(1 - \frac{(1-R-\gamma)N}{d_j}\right) - \frac{N}{Jd_j} - \frac{N(J+1)}{nd_j}.$$

(4.6) follows from (4.5) and Lemma 2.4. (4.7) follows from rearranging the summand and using the fact that the tuple is good (and hence $d_j \geqslant (1-R-\gamma)N$). (4.8) follows from the fact that there are q^{nN} choices for $\mathbf{y}$ and Lemma 4.2.[8] (4.9) follows from the fact that $d_j - (1-R)N + 1 \leqslant d_j - (1-R-\gamma)N$ (for $N \geqslant 1/\gamma$) and that $d_j \geqslant (1-R-\gamma)N$. (4.10) follows by rearranging the terms.

Now, as long as $n \geqslant J(J+1)$, we have $\frac{N(J+1)}{nd} \leqslant \frac{N}{Jd}$. Now (4.10) will imply (4.2) if we can show that for every $(1-R-\gamma)N \leqslant d \leqslant N$,

$$\frac{h}{nd} \leqslant H_q^{-1}\left(1 - r\left(1 - \frac{(1-R-\gamma)N}{d}\right) - \frac{2N}{Jd}\right) - \delta,$$

for $\delta = \varepsilon/3$. By Lemma 4.3 (which is stated at the end of this section), as long as $J \geqslant 4c_q'/(\delta^2(1-R))$ (and the

[8] As the final code C will be linear, it is sufficient to only look at received words that have Hamming weight at most ρnN. However, this gives a negligible improvement to the final result and hence, we just bound the number of choices for $\mathbf{y}$ by q^{nN}.

conditions on γ are satisfied), the above can be satisfied by picking

$$h/(nN) = H_q^{-1}(1 - rR) - 3\delta = \rho,$$

as required. We now verify that the conditions on γ in Lemma 4.3 are satisfied by picking $\gamma = \frac{4}{Jr}$. Note that if we choose $J = 4c'_q/(\delta^2(1-R))$, we will have $\gamma = \frac{\delta^2(1-R)}{c'_q r}$. Now, as $0 < R < 1$, we also have $\gamma \leqslant \delta^2/(rc'_q)$. Finally, we show that $\gamma \leqslant (1-R)/2$. Indeed

$$\gamma = \frac{\delta^2(1-R)}{c'_q r} = \frac{\varepsilon^2(1-R)}{9c'_q r} \leqslant \frac{\varepsilon(1-R)}{9r} < \frac{\alpha_q(r)(1-R)}{9r} < \frac{1-R}{2},$$

where the first inequality follows from the facts that $c'_q \geqslant 1$ and $\varepsilon \leqslant 1$. The second inequality follows from the assumption on ε. The third inequality follows from Lemma 2.2. As J is in $\Theta\left(\frac{1}{\varepsilon^2(1-R)}\right)$ (and γ is in $\Theta(\varepsilon^2(1-R)/r)$), we can choose $L = (N/\varepsilon^2)^{O\left(\varepsilon^{-4}(1-R)^{-2}\log(q/R)\right)}$, as required.

We still need to argue that with high probability the rate of the code $C = C_{out} \circ (C_{in}^1, \ldots, C_{in}^N)$ is rR. One way to argue this would be to show that with high probability all of the generator matrices have full rank. However, this is not the case: in fact, with some non-negligible probability at least one of them will not have full rank. However, we claim that with high probability C has distance > 0, and thus is a subspace of dimension $rRnN$. The proof above in fact implies that with high probability C has distance $(H_q^{-1}(1-rR) - \delta)nN$ for any small enough $\delta > 0$ with high probability. It is easy to see that to show that C has distance at least h, it is enough to show that with high probability $\sum_{\mathbf{m}\in\mathbb{F}_Q^K} \mathbf{I}(\mathbf{0}, \mathbf{m}) = 0$. Note that this is a special case of our proof, with $J = 1$ and $\mathbf{y} = \mathbf{0}$ and hence, with probability at least $1 - q^{\Omega(nN)}$, the code C has large distance.

The proof is thus complete, modulo the following lemma, which we prove next (following a similar argument in [14]).

Lemma 4.3. *Let q be a prime power, and $1 \leqslant n \leqslant N$ be integers. Let $0 < r, R \leqslant 1$ be rationals and $\delta > 0$ be a real such that $R \leqslant (\alpha_q(r) - \delta)/r$ and $\delta \leqslant \alpha_q(r)$, where $\alpha_q(r)$ is as defined in (2.1). Let $\gamma > 0$ be a real such that $\gamma \leqslant \min\left(\frac{1-R}{2}, \frac{\delta^2}{c'_q r}\right)$, where c'_q is the constant that depends only on q from Lemma 2.1. Then for all integers $J \geqslant \frac{4c'_q}{\delta^2(1-R)}$ and $h \leqslant (H_q^{-1}(1-rR) - 2\delta)nN$ the following is satisfied. For every integer $(1-R-\gamma)N \leqslant d \leqslant N$,*

(4.11)
$$\frac{h}{nd} \leqslant H_q^{-1}\left(1 - r\left(1 - \frac{N(1-R-\gamma)}{d}\right) - \frac{2N}{Jd}\right).$$

Proof. Using the fact H_q^{-1} is an increasing function, (4.11) is satisfied if the following is an upper bound on $h/(nN)$ for *every* $d^* \leqslant d \leqslant N$ (where $d^* = (1-R-\gamma)N$):

$$\left(\frac{d}{N}\right) \cdot H_q^{-1}\left(1 - r\left(1 - \frac{N(1-R-\gamma)}{d}\right) - \frac{2N}{d^* J}\right).$$

Define a new variable $\theta = 1 - N(1-R-\gamma)/d$. Note that as $d^* = (1-R-\gamma)N \leqslant d \leqslant N$, $0 \leqslant \theta \leqslant R + \gamma$. Also $d/N = (1-R-\gamma)(1-\theta)^{-1}$. Thus, the above inequality would be satisfied if $h/(nN(1-R-\gamma)$ is at most

$$\min_{0 \leqslant \theta \leqslant R+\gamma} \left\{ (1-\theta)^{-1} H_q^{-1}\left(1 - r\theta - \frac{2}{(1-R-\gamma)J}\right) \right\}.$$

Again using the fact that H_q^{-1} is an increasing function along with the fact that $\gamma \leqslant (1-R)/2$, we get that the above is satisfied if $h/(nN(1-R-\gamma)$ is at most

$$\min_{0 \leqslant \theta \leqslant R+\gamma} \left\{ (1-\theta)^{-1} H_q^{-1}\left(1 - r\theta - \frac{4}{(1-R)J}\right) \right\}.$$

By Lemma 2.1, if $J \geqslant \frac{4c'_q}{\delta^2(1-R)}$, then[9] $H_q^{-1}\left(1 - r\theta - \frac{4}{(1-R)J}\right) \geqslant H_q^{-1}(1 - r\theta) - \delta$. Since for every $0 \leqslant \theta \leqslant R + \gamma$, $(1-R-\gamma)(1-\theta)^{-1}\delta \leqslant \delta$, the above equation would be satisfied if

$$\frac{h}{nN} \leqslant (1-R-\gamma) \min_{0 < \theta \leqslant R+\gamma} f_{r,q}(\theta) - \delta.$$

Note that the assumptions $\gamma \leqslant \delta^2/(rc'_q) \leqslant \delta/r$ (as $\delta \leqslant 1$ and $c'_q \geqslant 1$) and $R \leqslant (\alpha_q(r) - \delta)/r$, we have $R + \gamma \leqslant \alpha_q(r)/r$. Thus, by using Lemma 2.3 we get that $(1-R-\gamma)\min_{0<\theta\leqslant R+\gamma} f_{r,q}(\theta) = H_q^{-1}(1-rR-r\gamma)$. By Lemma 2.1, the facts that $\gamma \leqslant \delta^2/(rc'_q)$ and H_q^{-1} is increasing, we have $H_q^{-1}(1-rR-r\gamma) \geqslant H_q^{-1}(1-rR) - \delta$. This implies that (4.11) is satisfied if $h/(nN) \leqslant H_q^{-1}(1-rR) - 2\delta$, as desired. □

5 List Decodability of Random Concatenated Codes

In this section, we will look at the list decodability of concatenated codes when both the outer code and the inner codes are (independent) random linear codes. The following is the main result of this section. The proof is similar to that of Theorem 5.1 and is omitted due to space limitations (see [11, Chap. 5] for the full proof).

[9] We also use the fact that H_q^{-1} is increasing.

Theorem 5.1. *Let q be a prime power and let $0 < r \leqslant 1$ be an arbitrary rational. Let $0 < \varepsilon < \alpha_q(r)$ be an arbitrary real, where $\alpha_q(r)$ is as defined in (2.1), and $0 < R \leqslant (\alpha_q(r) - \varepsilon)/r$ be a rational. Then the following holds for large enough integers n, N such that there exist integers k and K that satisfy $k = rn$ and $K = RN$. Let C_{out} be a random linear code over $\mathbb{F}_{q^k}$ that is generated by a random $K \times N$ matrix over $\mathbb{F}_{q^k}$. Let $C_{in}^1, \ldots, C_{in}^N$ be random linear codes over $\mathbb{F}_q$, where C_{in}^i is generated by a random $k \times n$ matrix $\mathbf{G}_i$ and the random choices for $C_{out}, \mathbf{G}_1, \ldots, \mathbf{G}_N$ are all independent. Then the concatenated code $C = C_{out} \circ (C_{in}^1, \ldots, C_{in}^N)$ is a $\left(H_q^{-1}(1 - Rr) - \varepsilon, q^{O\left(\frac{rn}{\varepsilon^2(1-R)}\right)}\right)$-list decodable code with probability at least $1 - q^{-\Omega(nN)}$ over the choices of $C_{out}, \mathbf{G}_1, \ldots, \mathbf{G}_N$. Further, with high probability, C has rate rR.*

Remark 5.1. *In a typical use of concatenated codes, the block lengths of the inner and outer codes satisfy $n = \Theta(\log N)$, in which case the concatenated code of Theorem 5.1 is list decodable with lists of size $N^{O\left(\varepsilon^{-2}(1-R)^{-1}\right)}$. However, the proof of Theorem 5.1 also works with smaller n. In particular as long as n is at least $3J^2$, the proof of Theorem 5.1 goes through. Thus, with n in $\Theta(J^2)$, one can get concatenated codes that are list decodable up to the list-decoding capacity with lists of size $q^{O\left(\varepsilon^{-6}(1-R)^{-3}\right)}$.*

6 Open Questions

In this work, we have shown that the family of concatenated codes is rich enough to contain codes that achieve the list-decoding capacity. But realizing the full potential of concatenated codes and achieving capacity (or even substantially improving upon the Blokh-Zyablov bound [8]) with explicit codes and polynomial time decoding remains a huge challenge. Achieving an explicit construction even without the requirement of an efficient list-decoding algorithm (but only good combinatorial list-decodability properties) is itself wide open.

The difficulty with explicit constructions is that we do not have any handle on the structure of inner codes that lead to concatenated codes with the required properties. In fact, we do not know of any efficient algorithm to even *verify* that a given set of inner codes will work, so even a Las Vegas construction appears difficult (a similar situation holds for binary codes meeting the Gilbert-Varshamov trade-off between rate and relative distance).

References

[1] E. L. Blokh and Victor V. Zyablov. Existence of linear concatenated binary codes with optimal correcting properties. *Prob. Peredachi Inform.*, 9:3–10, 1973.

[2] Ilya I. Dumer. Concatenated codes and their multilevel generalizations. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding Theory*, volume 2, pages 1911–1988. North Holland, 1998.

[3] G. David Forney. *Concatenated Codes.* MIT Press, Cambridge, MA, 1966.

[4] Venkatesan Guruswami, Johan Hastad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1035, 2002.

[5] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 658–667, 2001.

[6] Venkatesan Guruswami and Atri Rudra. Explicit capacity-achieving list-decodable codes. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 1–10, May 2006.

[7] Venkatesan Guruswami and Atri Rudra. Limits to list decoding Reed-Solomon codes. *IEEE Transactions on Information Theory*, 52(8), August 2006.

[8] Venkatesan Guruswami and Atri Rudra. Better binary list-decodable codes via multilevel concatenation. In *Proceedings of the 11th International Workshop on Randomization and Computation (RANDOM)*, pages 554–568, 2007.

[9] F. J. MacWilliams and Neil J. A. Sloane. *The Theory of Error-Correcting Codes.* Elsevier/North-Holland, Amsterdam, 1981.

[10] Jørn Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18:652–656, 1972.

[11] Atri Rudra. *List Decoding and Property Testing of Error Correcting Codes.* PhD thesis, University of Washington, 2007.

[12] Michael Sipser and Daniel Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[13] Daniel Spielman. The complexity of error-correcting codes. In *11th International Symposium on Fundamentals of Computation Theory,* Krakow, Poland, LNCS #1279, pp. 67-84, 1997

[14] Christian Thommesen. The existence of binary linear concatenated codes with Reed-Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 29(6):850–853, November 1983.