Electronic Colloquium on Computational Complexity, Report No. 109 (2007)

Better Binary List-Decodable Codes via Multilevel Concatenation*

Venkatesan Guruswami[1†] and Atri Rudra[2‡]

[1]Department of Computer Science and Engineering,
University of Washington,
Seattle, WA, 98195.
venkat@cs.washington.edu

[2]Department of Computer Science and Engineering,
University at Buffalo, State University of New York,
Buffalo, NY, 14260.
atri@cse.buffalo.edu

Abstract

We give a polynomial time construction of binary codes with the best currently known trade-off between rate and error-correction radius. Specifically, we obtain linear codes over fixed alphabets that can be list decoded in polynomial time up to the so called Blokh-Zyablov bound. Our work builds upon [8] where codes list decodable up to the Zyablov bound (the standard product bound on distance of concatenated codes) were constructed. Our codes are constructed via a (known) generalization of code concatenation called multilevel code concatenation. A probabilistic argument, which is also derandomized via conditional expectations, is used to show the existence of inner codes with a certain nested list decodability property that is appropriate for use in multilevel concatenated codes. A "level-by-level" decoding algorithm, which crucially uses the list recovery algorithm for folded Reed-Solomon codes from [8], enables list decoding up to the designed distance bound, aka the Blokh-Zyablov bound, for multilevel concatenated codes.

*A preliminary version of this paper appeared in the *Proceedings of the 11th International Workshop on Randomization and Computation (RANDOM)* as [7].

†Supported in part by NSF CCF-0343672, a Sloan Research Fellowship and a David and Lucile Packard Foundation Fellowship.

‡This work was done when the author was at University of Washington. Supported in part by NSF CCF-0343672.

ISSN 1433-8092

1 Introduction

1.1 Background and Context

A fundamental trade-off in the theory of error-correcting codes is the one between the proportion of redundancy built into codewords and the fraction of errors that can be corrected. Let us say we are interested in binary codes that can be used to recover the correct codeword even when up to a fraction ρ of its symbols could be corrupted by the channel. Such a channel can distort a codeword c (that is n bits long) into about $2^{H(\rho)n}$ possible received words, where $H(\rho) = -\rho \log_2 \rho - (1-\rho)\log_2(1-\rho)$ stands for the binary entropy function. Now for each of these words, the error-recovery procedure must identify c as a possibility for the true codeword. (In fact, even if the errors are random, the algorithm must identify c as a candidate codeword for *most* of these $2^{H(\rho)n}$ received words, if we seek a low decoding error probability.) To put it differently, if we require the error-recovery procedure to pin down a relatively small number of candidate codewords for all (or even most) received words, then there must be "nearly-disjoint" Hamming balls of size $2^{H(\rho)n}$ centered at each of the codewords. This implies that there can be at most about $2^{(1-H(\rho))n}$ codewords. Therefore the best rate of communication we can hope for when a fraction ρ of the bits can be corrupted is $1 - H(\rho)$.

If we could pack about $2^{(1-H(\rho))n}$ pairwise disjoint Hamming balls of radius ρn in $\{0,1\}^n$, then one can achieve a rate approaching $1 - H(\rho)$ while guaranteeing correct and unambiguous recovery of the codeword from an arbitrary fraction ρ of errors. Unfortunately, it is well known that such a "perfect" packing of Hamming balls in $\{0,1\}^n$ does not exist. Perhaps surprisingly (and fortunately), it turns out that it is possible to pack more than $2^{(1-H(\rho)-\varepsilon)n}$ such Hamming balls such that no $O(1/\varepsilon)$ of them intersect at a point. In fact a random packing has such a property with high probability.

In turn, this implies that for $0 < \rho < 1/2$ and any $\varepsilon > 0$, and all large enough n, there *exist* binary codes of rate $1 - H(\rho) - \varepsilon$ that enable correcting a fraction ρ of errors, outputting a list of at most $O(1/\varepsilon)$ answers in the worst-case (this error-recovery model is called "list decoding").[1] Therefore, one can approach the information-theoretically optimal rate of $1 - H(\rho)$. A similar result holds for codes over alphabet with q symbols – for correction of a fraction ρ, $0 < \rho < 1 - 1/q$, of errors, we can approach the optimal rate of $1 - H_q(\rho)$, where $H_q(\rho) = \rho \log_q(q-1) - \rho \log_q \rho - (1-\rho)\log_q(1-\rho)$ is the q-ary entropy function.

While the above pinpoints $R = 1 - H(\rho)$ as the optimal trade-off between the rate R of the code and the fraction ρ of errors that can corrected, it is a non-constructive result. The codes achieving this trade-off are shown to exist via a random coding argument and are not explicitly specified. Further, for a code to be useful, the decoding algorithm must be efficient, and for a random, unstructured code only brute-force decoders running in exponential time are known.

The big challenge then is to approach the above trade-off with explicit codes and polynomial time list decoding algorithms. Recently, in [8], we were able to achieve such a result for large alphabets. For large q, the optimal rate $1 - H_q(\rho)$ approaches $1 - \rho$, and in [8], we give explicit codes of rate $1 - \rho - \varepsilon$ over an alphabet of size $2^{(1/\varepsilon)^{O(1)}}$ with a polynomial time list decoding algorithm for a fraction ρ of errors (for any $0 < \rho < 1$). However, approaching the *list decoding capacity* of $1 - H_q(\rho)$ for any fixed small alphabet size q, such as $q = 2$, remains an important open question.

[1] The proof of Shannon's theorem for the binary symmetric channel also says that for most received words at most one codeword would be output.

The best known trade-off between R and ρ (from [8]) that can be achieved by an explicit binary code along with efficient list decoding algorithm is the so called Zyablov bound [14]. Figure 1 gives a pictorial comparison between the Zyablov bound and the list decoding capacity. As one can see, there is a still a huge gap between the nonconstructive results and what is known explicitly, closing which is a challenging open problem. Narrowing this gap serves as the primary motivation for this work.

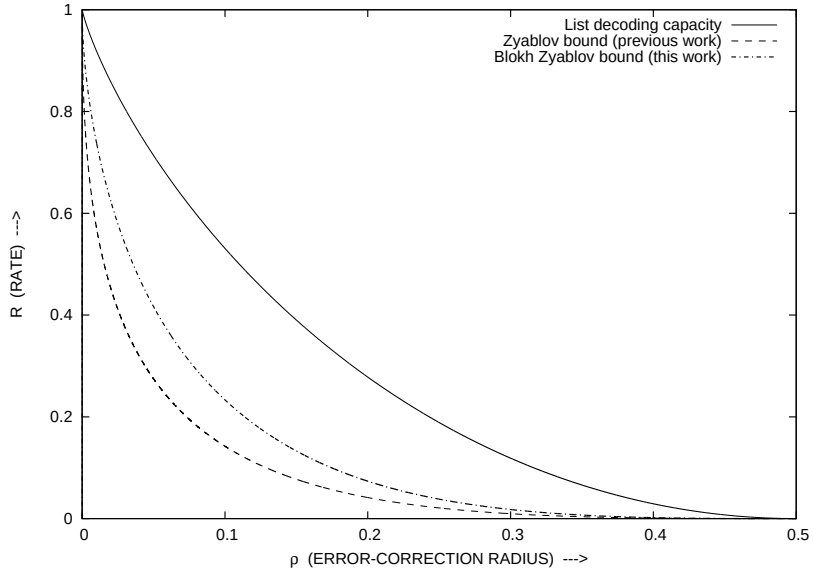

Figure 1: Rate R of our binary codes plotted against the error-correction radius ρ of our algorithm. The best possible trade-off, i.e., capacity, is $\rho = H^{-1}(1 - R)$, and the Zyablov bound are also plotted.

ρ	0.01	0.02	0.03	0.05	0.10	0.15	0.20	0.25	0.30	0.35
Capacity rate	0.919	0.858	0.805	0.713	0.531	0.390	0.278	0.188	0.118	0.065
Zyablov rate	0.572	0.452	0.375	0.273	0.141	0.076	0.041	0.020	0.009	0.002
Blokh Zyablov rate	0.739	0.624	0.539	0.415	0.233	0.132	0.073	0.037	0.017	0.006

Table 1: Values of rate at different error correction radius for List decoding capacity, Zyablov bound and Blokh Zyablov bound in the binary case. For rates above 0.4, the Blokh Zyablov bound is 0 up to 3 decimal places, hence we have not shown this.

1.2 Our Results and Techniques

In this paper, we present linear codes over any fixed alphabet that can be constructed in polynomial time and can be efficiently list decoded up to the so called Blokh-Zyablov bound. This achieves a sizable improvement over the previous best known result (see Figure 1 and Table 1 for the binary case).

Our codes are constructed via multilevel concatenated codes. We will provide a formal definition later on — we just sketch the basic idea here. For an integer $s \geqslant 1$, a multilevel concatenated code C over $\mathbb{F}_q$ is obtained by combining s "outer" codes $C^0_{out}, C^1_{out}, \ldots, C^{s-1}_{out}$ of the same block length , say N, over large alphabets of size say $q^{a_0}, q^{a_1}, \ldots, q^{a_{s-1}}$, respectively, with a suitable "inner" code over $\mathbb{F}_q$. The inner code, say C_{in}, is of dimension $a_0 + a_1 \cdots + a_{s-1}$. Given messages $m_0, m_1, \ldots, m_{s-1}$ for the s outer codes, the encoding as per the multilevel generalized concatenation codes proceeds by first encoding each m_j as per C^j_{out}. Then for every $1 \leqslant i \leqslant N$, the collection of the ith symbols of $C^j_{out}(m_j)$ for $0 \leqslant j \leqslant s-1$, which can be viewed as a string over $\mathbb{F}_q$ of length $a_0 + a_1 + \cdots + a_{s-1}$, is encoded by the inner code. For $s = 1$ this reduces to the usual definition of code concatenation. In other words, this is like normal code concatenation with inner code C_{in} and outer code obtained by juxtaposing the symbols of codewords of $C^0_{out}, \ldots, C^{s-1}_{out}$.

We present a list decoding algorithm for C, given list recovery algorithms for the outer codes (list recovery is a generalization of list decoding that will be defined later) and list decoding algorithms for the inner code and some of its subcodes. What makes this part more interesting than the usual code concatenation, is that the inner code in addition to having list decodable properties, also needs to have good list decodable properties for certain subcodes. Specifically, the subcodes of dimension $a_j + a_{j+1} + \cdots + a_{s-1}$ of the inner code obtained by arbitrarily fixing the first $a_0 + \cdots + a_{j-1}$ symbols of the message, must have better list-decodability properties for increasing j (which is intuitively possible since they have lower rate). In turn, this allows the outer codes C^j_{out} to have rates increasing with j, leading to an overall improvement in the rate for a certain list-decoding radius.

To make effective use of the above approach, we also prove, via an application of the probabilistic method, that a random linear code over $\mathbb{F}_q$ has the required stronger condition on list decodability. By applying the method of conditional expectation ([1]), we can construct such a code deterministically in time singly exponential in the block length of the code (which is polynomial if the inner code encodes messages of length $O(\log n)$). Note that constructing such an inner code, given the existence of such codes, is easy in quasi-polynomial time by trying all possible generator matrices. The lower time complexity is essential for constructing the final code C in polynomial time.

1.3 Related Work

Our work can be seen as a generalization of the result of list decoding concatenated codes from [8]. The outer codes used in our work are the same as the ones used in [8]. However, the inner codes used in [8] are not sufficient for our purposes. Our proof of existence of the requisite inner codes (and in particular the derandomization of the construction of such codes using conditional expectation) is similar to the one used to establish list decodability properties of random "pseudolinear" codes in [6] (see also [5, Sec. 9.3]).

Concatenated codes were defined in the seminal thesis of Forney [4]. Its generalizations to linear multilevel concatenated codes were introduced by Blokh and Zyablov [2] and general multilevel concatenated codes were introduced by Zinoviev [12]. Our list decoding algorithm is inspired by the argument for "unequal error protection" property of multilevel concatenated codes [13].

1.4 Organization of the Paper

In Section 2, we start with some definitions and preliminaries. Section 3 presents a construction of a linear code that has good "nested" list decodable properties. In section 4, we present our

algorithm for list decoding multilevel concatenated codes. In Section 5, we present the main result of the paper. We conclude with some remarks in Section 6.

2 Preliminaries

2.1 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.

2.2 Multilevel Concatenated Codes

We will be working with multilevel concatenation coding schemes [3]. We start this section with the definition of multilevel concatenated codes. As the name suggests, these are generalizations of the well-studied concatenated codes. Recall that for a concatenated code, we start with a code C_{out} over a large alphabet (called the outer code). Then we need a code C_{in} that maps all symbols of the larger alphabet to strings over a smaller alphabet (called the inner code). The encoding for the concatenated code (denoted by $C_{out} \circ C_{in}$) is done as follows. We think of the message as being a string over the large alphabet and then encode it using C_{out}. Now we use C_{in} to encode each of the symbols in the codeword of C_{out} to get our codeword (in $C_{out} \circ C_{in}$) over the smaller alphabet. Most of the constructions of good binary codes are achieved via code concatenation. In particular, binary codes with the best known trade-off (called the Zyablov bound) between rate and list decoding radius are constructed via code concatenation [8]. These codes have folded Reed-Solomon codes as outer codes and suitably chosen binary codes as inner codes, and can be list decoded up to the designed minimum distance, which is equal to the product of the outer and inner code distances.

Multilevel concatenation codes generalize the usual code concatenations in the following manner. Instead of there being one outer code, there are multiple outer codes. In particular, we "stack" codewords from these multiple outer codes and construct a matrix. The inner codes then act on the columns of these intermediate matrix. We now formally define multilevel concatenated codes (this will also contain the formal definition of the concatenated codes as a special case).

There are $s \geqslant 1$ outer codes, denoted by $C_{out}^0, C_{out}^1, \ldots, C_{out}^{s-1}$. For every $0 \leqslant i \leqslant s-1$, C_{out}^i is a code of block length N and rate R_i and defined over a field $\mathbb{F}_{Q_i}$. The inner code C_{in} is code of block length n and rate r that maps tuples from $\mathbb{F}_{Q_0} \times \mathbb{F}_{Q_1} \times \cdots \times \mathbb{F}_{Q_{s-1}}$ to symbols in $\mathbb{F}_q$. In other words,

$$C_{out}^i : (\mathbb{F}_{Q_i})^{R_i N} \to (\mathbb{F}_{Q_i})^N,$$

$$C_{in} : \mathbb{F}_{Q_0} \times \mathbb{F}_{Q_1} \times \cdots \times \mathbb{F}_{Q_{s-1}} \to (\mathbb{F}_q)^n.$$

The multilevel concatenated code, denoted by $(C^0_{out} \times C^1_{out} \times \ldots C^{s-1}_{out}) \circ C_{in}$ is a map of the following form:

$$(C^0_{out} \times C^1_{out} \times \ldots C^{s-1}_{out}) \circ C_{in} : (\mathbb{F}_{Q_0})^{R_0 N} \times (\mathbb{F}_{Q_1})^{R_1 N} \times \cdots \times (\mathbb{F}_{Q_{S-1}})^{R_{s-1} N} \to \mathbb{F}_q^{nN}.$$

We now describe the encoding scheme. Given a message $(m_0, m_1, \ldots, m_{s-1}) \in (\mathbb{F}_{Q_0})^{R_0 N} \times (\mathbb{F}_{Q_1})^{R_1 N} \times \cdots \times (\mathbb{F}_{Q_{S-1}})^{R_{s-1} N}$, we first construct an $s \times N$ matrix M, whose i^{th} row is the codeword $C^i_{out}(m_i)$. Note that every column of M is an element from the set $\mathbb{F}_{Q_0} \times \mathbb{F}_{Q_1} \times \cdots \times \mathbb{F}_{Q_{s-1}}$. Let the j^{th} column (for $1 \leqslant j \leqslant N$) be denoted by M_j. The codeword corresponding to the multilevel concatenated code ($C \stackrel{def}{=} (C^0_{out} \times C^1_{out} \times \ldots C^{s-1}_{out}) \circ C_{in}$) is defined as follows

$$C(m_0, m_1, \ldots, m_{s-1}) = (C_{in}(M_1), C_{in}(M_2), \cdots, C_{in}(M_N)) .$$

(The codeword can be naturally be thought of as an $n \times N$ matrix, whose i'th column corresponds to the inner codeword encoding the i'th symbols of the s outer codewords.)

For the rest of the paper, we will only consider outer codes over the same alphabet, that is, $Q_0 = Q_1 = \cdots = Q_{s-1} = Q$. Further, $Q = q^a$ for some integer $a \geqslant 1$. Note that if $C^0_{out}, \ldots, C^{s-1}_{out}$ and C_{in} are all $\mathbb{F}_q$ linear, then so is $(C^0_{out} \times C^1_{out} \times \cdots \times C^{s-1}_{out}) \circ C_{in}$.

The gain from using multilevel concatenated codes comes from looking at the inner code C_{in} along with its subcodes. For the rest of the section, we will consider the case when C_{in} is linear (though the ideas can easily be generalized for general codes). Let $G \in \mathbb{F}_q^{as \times n}$ be the generator matrix for C_{in}. Let $r_0 = as/n$ denote the rate of C_{in}. For $0 \leqslant j \leqslant s-1$, define $r_j = r_0(1 - j/s)$, and let G_j denote $r_j n \times n$ submatrix of G containing the last $r_j n$ rows of G. Denote the code generated by G_j by C^j_{in}; the rate of C^j_{in} is r_j. For our purposes we will actually look at the subcode of C_{in} where one fixes the first $0 \leqslant j \leqslant s-1$ message symbols. Note that for every j these are just cosets of C^j_{in}. We will be looking at C_{in}, which in addition to having good list decoding properties as a "whole," also has good list decoding properties for each of its subcode C^j_{in}.

The multilevel concatenated code C $(= (C^0_{out} \times \cdots \times C^{s-1}_{out}) \circ C_{in})$ has rate $R(C)$ that satisfies

$$R(C) = \frac{r_0}{s} \sum_{i=0}^{s-1} R_i \ . \tag{1}$$

The Blokh-Zyablov bound is the trade-off between rate and relative distance obtained when the outer codes meet the Singleton bound (i.e., C^j_{out} has relative distance $1 - R_j$), and the various subcodes C^j_{in} of the inner code, including the whole inner code $C_{in} = C^0_{in}$, lie on the Gilbert-Varshamov bound (i.e., have relative distance $\delta_j \geqslant H_q^{-1}(1 - r_j)$). The multilevel concatenated code then has relative distance at least $\min_{0 \leqslant j \leqslant s-1}(1 - R_j) H_q^{-1}(1 - r_j)$. Expressing the rate in terms of distance, the Blokh-Zyablov bound says that there exist multilevel concatenated C with relative distance at least δ with the following rate:

$$R^s_{BZ}(C) = \max_{0 < r < 1 - H_q(\delta)} r - \frac{r}{s} \sum_{i=0}^{s-1} \frac{\delta}{H_q^{-1}(1 - r + ri/s)}. \tag{2}$$

As s increases, the trade-off approaches the integral

$$R_{BZ}(C) = 1 - H_q(\delta) - \delta \int_0^{1 - H_q(\delta)} \frac{dx}{H_q^{-1}(1 - x)} \ . \tag{3}$$

The convergence of $R^s_{BZ}(C)$ to $R_{BZ}(C)$ happens quite quickly even for small s such as $s = 10$.

2.3 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 need to work with two different generalizations of list decoding. The first one is motivated by multilevel concatenation schemes. The definition looks more complicated than it really is.

Definition 2.2 (Nested linear list decodable code)**.** *Given a linear code C in terms of some generator matrix $G \in \mathbb{F}_q^{k \times n}$, an integer s that divides k, a vector $\mathbf{L} = \langle L_0, L_1, \ldots, L_{s-1} \rangle$ of integers L_j $(0 \leqslant j \leqslant s-1)$, a vector $\rho = \langle \rho_0, \rho_1 \ldots, \rho_{s-1} \rangle$ with $0 < \rho_j < 1$, and a vector $\mathbf{r} = \langle r_0, \ldots, r_{s-1} \rangle$ of reals where $r_0 = k/n$ and $0 \leqslant r_{s-1} < \cdots < r_i < r_0$, C is called an $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable if the following holds:*

For every $0 \leqslant j \leqslant s-1$, C^j is a rate r_j code that is (ρ_j, L_j)-list decodable, where C^j is the subcode of C generated by the the last $r_j n$ rows of the generator matrix G.

The second generalization of list decoding called list recovery, a term first coined in [6] even though the notion existed before, 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 lists (or more accurately sets, since the ordering of elements in the input lists does not matter).

Definition 2.3 (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 following simple folklore lemma shows how, for suitable parameters, a list recoverable outer code can be concatenated with a list decodable inner code to give a new list decodable code. The approach is simply to run the list decoding algorithm for each of the inner blocks, returning a list of possible symbols for each possible outer codeword symbol, which are then used as input to the list recovery algorithm for the outer code.

Lemma 2.1. *If C_{out} is a (ξ, ℓ, L)-list recoverable over an alphabet of size Q, and C_{in} is a (ρ, ℓ)-list decodable code with Q codewords, then the concatenated code $C_{out} \circ C_{in}$ is $(\xi \cdot \rho, L)$-list decodable.*

2.4 Known Result on List Recoverable Codes

We will use the following powerful result concerning good list recoverable codes from [8]; these codes will serve as the outer codes in our multilevel concatenation scheme.

Theorem 2.2. *For every integer $\ell \geqslant 1$, for all constants $\varepsilon > 0$, for all R, R'; $0 < R \leqslant R' < 1$, 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 $(N/\varepsilon^2)^{O(\varepsilon^{-2} \log \ell/(1-R'))}$.*

We remark that the above theorem was stated with $R' = R$ in [8], though the above follows immediately from the proof for $R' = R$ and properties of the folded Reed-Solomon codes [11]. The proof for $R' > R$ uses folded Reed-Solomon codes with a larger "folding" parameter. A larger folding parameter increases the fraction of errors that can be tolerated at the expense of a larger alphabet size.

3 Linear Codes with Good Nested List Decodability

In this section, we will prove the following result concerning the existence (and constructibility) of linear codes over any fixed alphabet with good nested list decodability properties.

Theorem 3.1. *For any integer $s \geqslant 1$ and reals $0 < r_{s-1} < r_{s-2} < \cdots < r_1 < r_0 < 1$, $\varepsilon > 0$, let $\rho_j = H_q^{-1}(1 - r_j - 2\varepsilon)$ for every $0 \leqslant j \leqslant s-1$. Let $\mathbf{r} = \langle r_0, \ldots, r_{s-1} \rangle$, $\rho = \langle \rho_0, \rho_1, \ldots, \rho_{s-1} \rangle$ and $\mathbf{L} = \langle L_0, L_1, \ldots, L_{s-1} \rangle$, where $L_j = q^{1/\varepsilon}$. For large enough n, there exists a linear code (over fixed alphabet $\mathbb{F}_q$) that is $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable. Further, such a code can be constructed in time $q^{O(n/\varepsilon)}$.*

Proof. We will show the existence of the required codes via a simple use of the probabilistic method (in fact, we will show that a random linear code has the required properties with high probability). We will then use the method of conditional expectation ([1]) to derandomize the construction with the claimed time complexity.

Define $k_j = \lfloor r_i n \rfloor$ for every $0 \leqslant j \leqslant s-1$. We will pick a random $k_0 \times n$ matrix G with entries picked independently from $\mathbb{F}_q$. We will show that the linear code C generated by G has good nested list decodable properties with high probability. Let C_j, for $0 \leqslant j \leqslant s-1$ be the code generated by the "bottom" k_j rows of G. Recall that we have to show that with high probability C_j is $(\rho_j, q^{1/\varepsilon})$ list decodable for every $0 \leqslant j \leqslant s-1$ (C_j obviously has rate r_j). Finally for integers $J, k \geqslant 1$, and a prime power q, let $\mathrm{Ind}(q, k, J)$ denote the collection of subsets $\{x_1, x_2, \ldots, x_J\} \subseteq \mathbb{F}_q^k$ such that all vectors $x_1, \ldots, x_J$ are linearly independent over $\mathbb{F}_q$.

We need the following two claims: (i) Given any L distinct vectors from $\mathbb{F}_q^k$, for some $k \geqslant 1$, at least $\lceil \log_q L \rceil$ of them are linearly independent; (ii) Any set of linearly independent vectors in $\mathbb{F}_q^k$ are mapped to independent random vectors in $\mathbb{F}_q^n$ by a random $k \times n$ matrix over $\mathbb{F}_q$. The first claim is obvious. For the second claim, first note that for any $\mathbf{v} \in \mathbb{F}_q^k$ and a random $k \times n$ matrix $\mathbf{G}$ (where each of the kn values are chosen uniformly and independently at random from $\mathbb{F}_q$) the values at the n different positions in $\mathbf{v} \cdot \mathbf{G}$ are independent. Further, the value at position $1 \leqslant i \leqslant n$, is given by $\mathbf{v} \cdot \mathbf{G}_i$, where $\mathbf{G}_i$ is the i^{th} column of $\mathbf{G}$. Now for fixed $\mathbf{v}$, $\mathbf{v} \cdot \mathbf{G}_i$ takes values from $\mathbb{F}_q$ uniformly at random (note that $\mathbf{G}_i$ is a random vector from $\mathbb{F}_q^k$). Finally, for linearly independent vectors $\mathbf{v}_1, \ldots, \mathbf{v}_m$ by a suitable linear invertible map can be mapped to the standard basis vectors $\mathbf{e}_1, \ldots, \mathbf{e}_m$. Obviously, the values $\mathbf{e}_1 \cdot \mathbf{G}_i \ \ldots, \mathbf{e}_m \cdot \mathbf{G}_i$ are independent.

We now move on to the proof of existence of linear codes with good nested list decodability. We will actually do the proof in a manner that will facilitate the derandomization of the proof. Define $J = \lceil \log_q(q^{1/\varepsilon} + 1) \rceil$. For any vector $\mathbf{r} \in \mathbb{F}_q^n$, integer $0 \leqslant j \leqslant s-1$, subset $T = \{x_1, \ldots, x_J\} \in \mathrm{Ind}(q, k_j, J)$ and any collection $\mathcal{S}$ of subsets $S_1, S_2, \ldots, S_J \subseteq \{1, \ldots, n\}$ of size at most $\rho_j n$, define an indicator variable $I(j, \mathbf{r}, T, \mathcal{S})$ in the following manner. $I(j, \mathbf{r}, T, \mathcal{S}) = 1$ if and only if for every $1 \leqslant i \leqslant J$, $C(x_i)$ differs from $\mathbf{r}$ in exactly the set S_i. Note that if for some $0 \leqslant j \leqslant s-1$, there are $q^{1/\varepsilon} + 1$ codewords in C_j all of which differ from some received word $\mathbf{r}$ in at most $\rho_j n$ places, then this set of codewords is a "counter-example" that shows that C is not $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable. Since the $q^{1/\varepsilon} + 1$ codewords will have some set T of J linearly independent codewords, the counter example will imply that $I(j, \mathbf{r}, T, \mathcal{S}) = 1$ for some collection of subsets $\mathcal{S}$. In other words, the indicator variable captures the set of bad events we would like to avoid. Finally define

the sum of all the indicator variables as follows:

$$S_C = \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} \sum_{\substack{\mathcal{S}=\{S_1,\ldots,S_J\},\\ S_i\subseteq\{1,\ldots,n\},|S_i|\leqslant\rho_j n}} I(j,\mathbf{r},T,\mathcal{S}).$$

Note that if $S_C = 0$, then C is $(\mathbf{r},\rho,\mathbf{L})$-nested list decodable as required. Thus, we can prove the existence of such a C if we can show that $\mathbf{E}_{\mathbf{C}}[S_C] < 1$. By linearity of expectation, we have

$$\mathbf{E}[S_C] = \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} \sum_{\substack{\mathcal{S}=\{S_1,\ldots,S_J\},\\ S_i\subseteq\{1,\ldots,n\},|S_i|\leqslant\rho_j n}} \mathbf{E}[I(j,\mathbf{r},T,\mathcal{S})]. \tag{4}$$

Fix some arbitrary $j,\mathbf{r},T=\{x_1,x_2,\ldots,x_J\},\mathcal{S}=\{S_1,S_2,\ldots,S_J\}$ (in their corresponding domains). Then we have

$$\begin{aligned}
\mathbf{E}[I(j,\mathbf{r},T,\mathcal{S})] &= \Pr[I(j,\mathbf{r},T,\mathcal{S})=1] \\
&= \prod_{x_i\in T} \Pr[C(x_i) \text{ differ from } \mathbf{r} \text{ in exactly the positions in } S_i] \\
&= \prod_{i=1}^{J} \left(\frac{q-1}{q}\right)^{|S_i|} \left(\frac{1}{q}\right)^{n-|S_i|} = \prod_{i=1}^{J} \frac{(q-1)^{|S_i|}}{q^n},
\end{aligned} \tag{5}$$

where the second and the third equality follow from the definition of the indicator variable, the fact that vectors in T are linearly independent and the fact that a random matrix maps linearly independent vectors to independent uniformly random vectors in $\mathbb{F}_q^n$. Using (5) in (4), we get

$$\begin{aligned}
\mathbf{E}[S_C] &= \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} \sum_{\substack{\mathcal{S}=\{S_1,\ldots,S_J\},\\ S_i\subseteq\{1,\ldots,n\},|S_i|\leqslant\rho_j n}} \prod_{i=1}^{J} \frac{(q-1)^{|S_i|}}{q^n} \\
&= \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} \sum_{(\ell_1,\ell_2,\ldots,\ell_J)\in\{0,1,\ldots,\rho_j n\}^J} \prod_{i=1}^{J} \binom{n}{\ell_i} \frac{(q-1)^{\ell_i}}{q^n} \\
&= \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} \left(\sum_{\ell=0}^{\rho_j n} \binom{n}{\ell} \frac{(q-1)^{\ell}}{q^n}\right)^J \\
&\leqslant \sum_{j=0}^{s-1} \sum_{\mathbf{r}\in\mathbb{F}_q^n} \sum_{T\in \mathrm{Ind}(q,k_j,J)} q^{nJ(H_q(\rho_j)-1)} \leqslant \sum_{j=0}^{s-1} q^n \cdot q^{Jk_j} \cdot q^{nJ(H_q(\rho_j)-1)} \\
&\leqslant \sum_{j=0}^{s-1} q^{nJ(1/J+r_j+1-r_j-2\varepsilon-1)} \leqslant s q^{-\varepsilon n J}.
\end{aligned} \tag{6}$$

The first inequality follows the following known inequality for $p < 1-1/q$ ([10]): $\sum_{i=0}^{pn} \binom{n}{i}(q-1)^i \leqslant q^{H_q(p)n}$. The second inequality follows by upper bounding the number of J linearly independent vectors in $\mathbb{F}_q^{k_j}$ by q^{Jk_j}. The third inequality follows from the fact that $k_j = \lfloor r_j n \rfloor$ and $\rho_j = H_q^{-1}(1-r_j-2\varepsilon)$, The final inequality follows from the fact that $J = \lceil \log_q(q^{1/\varepsilon}+1) \rceil$.

Thus, (6) shows that there exists a code C (in fact with high probability) that is $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable. In fact, this could have been proved using a simpler argument. However, the advantage of the argument above is that we can now apply the method of conditional expectations to derandomize the above proof.

The algorithm to deterministically generate a linear code C that is $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable is as follows. The algorithm consists of n steps. At any step $1 \leqslant i \leqslant n$, we choose the i^{th} column of the generator matrix to be the value $\mathbf{v}_i \in \mathbb{F}_q^{k_0}$ that minimizes the conditional expectation $\mathbf{E}[S_C | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_{i-1} = \mathbf{v}_{i-1}, \mathbf{G}_i = \mathbf{v}_i]$, where $\mathbf{G}_i$ denotes the i^{th} column of $\mathbf{G}$ and $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$ are the column vectors chosen in the previous $i-1$ steps. This algorithm would work only if for any $1 \leqslant i \leqslant n$ and vectors $\mathbf{v}_1, \ldots, \mathbf{v}_i$, we can exactly compute $\mathbf{E}[S_C | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_i = \mathbf{v}_i]$. Indeed from (4), we have $\mathbf{E}[S_C | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_i = \mathbf{v}_i]$ is

$$\sum_{j=0}^{s-1} \sum_{\mathbf{r} \in \mathbb{F}_q^n} \sum_{T \in \mathrm{Ind}(q, k_j, J)} \sum_{\substack{\mathcal{S} = \{S_1, \ldots, S_J\}, \\ S_i \subseteq \{1, \ldots, n\}, |S_i| \leqslant \rho_j n}} \mathbf{E}[I(j, \mathbf{r}, T, \mathcal{S}) | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_i = \mathbf{v}_i].$$

Thus, we would be done if we can compute the following for every value of $j, \mathbf{r}, T = \{x_1, \ldots, x_J\}, \mathcal{S} = \{S_1, \ldots, S_J\}$: $\mathbf{E}[I(j, \mathbf{r}, T, \mathcal{S}) = 1 | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_i = \mathbf{v}_i]$. Note that fixing the first i columns of G implies fixing the value of the codewords in the first i positions. Thus, the indicator variable is 0 (or in other words, the conditional expectation we need to compute is 0) if for some message, the corresponding codeword does not disagree with $\mathbf{r}$ exactly as dictated by $\mathcal{S}$ in the first i positions. More formally, $I(j, \mathbf{r}, T, \mathcal{S}) = 0$ if the following is true for some $1 \leqslant \ell \leqslant i$ and $0 \leqslant i' \leqslant J$: $x_{i'} \cdot \mathbf{G}_\ell \neq \mathbf{r}_\ell$, if $\ell \notin S_{i'}$ and $x_{i'} \cdot \mathbf{G}_\ell = \mathbf{r}_\ell$ otherwise. However, if none of these conditions hold, then using argument similar to the ones used to obtain (5), one can show that

$$\mathbf{E}[I(j, \mathbf{r}, T, \mathcal{S}) | \mathbf{G}_1 = \mathbf{v}_1, \ldots, \mathbf{G}_i = \mathbf{v}_i] = \prod_{\ell=1}^{J} \left(\frac{q-1}{q}\right)^{|S'_\ell|} \left(\frac{1}{q}\right)^{n-i-|S'_\ell|},$$

where $S'_\ell = S_\ell \setminus \{1, 2, \ldots, i\}$ for every $1 \leqslant \ell \leqslant J$.

To complete the proof, we need to estimate the time complexity of the above algorithm. There are n steps and at every step i, the algorithm has to consider $q^{k_0} \leqslant q^n$ different choices of $\mathbf{v}_i$. For every choice of $\mathbf{v}_i$, the algorithm has to compute the conditional expectation of the indicator variables for all possible values of $j, \mathbf{r}, T, \mathcal{S}$. It is easy to check that there are $\sum_{i=1}^{s} q^n \cdot q^{Jk_j} \cdot 2^{nJ} \leqslant s q^{n(1+2J)}$ possibilities. Finally, the computation of the conditional expected value of a fixed indicator variable takes time $O(snJ)$. Thus, in all the total time taken is $O(n \cdot q^n \cdot s q^{n(1+2J)} \cdot snJ) = q^{O(n/\varepsilon)}$, as required. □

4 List Decoding Multilevel Concatenated Codes

In this section, we will see how one can list decode multilevel concatenated codes, provided the outer codes have good list recoverability and the inner code has good nested list decodability. We have the following result, which generalizes Lemma 2.1 for regular concatenated codes (the case $s = 1$).

Theorem 4.1. *Let $s \geqslant 1$ and $\ell \geqslant 1$ be integers. Let $0 < R_0 < R_1 < \cdots < R_{s-1} < 1$, $0 < r_0 < 1$, $0 < \xi_0, \cdots, \xi_{s-1} < 1$, $0 < \rho_0, \cdots, \rho_{s-1} < 1$ and $\varepsilon > 0$ be reals. Let q be a prime power and let*

$Q = q^a$ for some integer $a > 1$. Further, let C^j_{out} ($0 \leqslant j \leqslant s-1$) be an $\mathbb{F}_q$-linear code over $\mathbb{F}_Q$ of rate R_j and block length N that is (ξ_j, ℓ, L)-list recoverable. Finally, let C_{in} be a linear $(\mathbf{r}, \rho, \mathbf{L})$-nested list decodable code over $\mathbb{F}_q$ of rate r_0 and block length $n = as/r_0$, where $\mathbf{r} = \langle r_0, \cdots, r_{s-1}\rangle$ with $r_i = (1 - i/s)r_0$, $\rho = \langle \rho_0, \cdots, \rho_{s-1}\rangle$ and $\mathbf{L} = \langle \ell, \ell, \cdots, \ell\rangle$. Then $C = (C^0_{out} \times \cdots \times C^{s-1}_{out}) \circ C_{in}$ is a linear $(\min_j \xi_j \cdot \rho_j, L^s)$-list decodable code. Further, if the outer code C^j_{out} can be list recovered in time $T_j(N)$ and the inner code C_{in} can be list decoded in time $t_j(n)$ (for the j^{th} level), then C can be list decoded in time $O\left(\sum_{j=0}^{s-1} L^j \left(T_j(N) + N \cdot t_j(n)\right)\right)$.

Proof. Given list recovery algorithms for C^j_{out} and list decoding algorithms for C_{in} (and its subcodes C^j_{in}), we will design a list decoding algorithm for C. Recall that the received word is an $n \times N$ matrix over $\mathbb{F}_q$. Each consecutive "chunk" of n/s rows should be decoded to a codeword in C^j_{out}. The details follow.

Before we describe the algorithm, we will need to fix some notation. Define $\delta = \min_j \xi_j \rho_j$. Let $\mathbf{R} \in \mathbb{F}_q^{nN}$ be the received word, which we will think of as an $n \times N$ matrix over $\mathbb{F}_q$ (note that s divides n). For any $n \times N$ matrix M and for any $1 \leqslant i \leqslant N$, let $M_i \in \mathbb{F}_q^n$ denote the i^{th} column of the matrix M. Finally, for every $0 \leqslant j \leqslant s-1$, let C^j_{in} denote the subcode of C_{in} generated by all but the first ja rows of the generator matrix of C_{in}. We are now ready to describe our algorithm.

Recall that the algorithm needs to output all codewords in C that differ from $\mathbf{R}$ in at most δ fraction of positions. For the ease of exposition, we will consider an algorithm that outputs matrices from $C^0_{out} \times \cdots \times C^{s-1}_{out}$. The algorithm has s phases. At the end of phase j ($0 \leqslant j \leqslant s-1$), the algorithm will have a list of matrices (called $\mathcal{L}_j$) from $C^0_{out} \times \cdots \times C^j_{out}$, where each matrix in $\mathcal{L}_j$ is a possible submatrix of some matrix that will be in the final list output by the algorithm. The following steps are performed in phase j (where we are assuming that the list decoding algorithm for C^j_{in} returns a list of messages while the list recovery algorithm for C^j_{out} returns a list of codewords).

1. Set $\mathcal{L}_j$ to be the empty set.
2. For every $\mathbf{c} = (c_0, \cdots, c_{j-1}) \in \mathcal{L}_{j-1}$ repeat the following steps (if this is the first phase, that is $j = 0$, then repeat the following steps once):
   (a) Let G_j be the first aj rows of the generator matrix of C_{in}. Let $\mathbf{X} = (G_j)^T \cdot \mathbf{c}$, where we think of $\mathbf{c}$ as an $ja \times N$ matrix over $\mathbb{F}_q$. Let $\mathbf{Y} = \mathbf{R} - \mathbf{X}$ (for $j = 0$ we use the convention that $\mathbf{X}$ is the all 0s matrix). For every $1 \leqslant i \leqslant N$, use the list decoding algorithm for C^j_{in} on column $\mathbf{Y}_i$ for up to ρ_j fraction of errors to obtain list $S^j_i \subseteq (\mathbb{F}_Q)^{s-j}$. Let $T^j_i \subseteq \mathbb{F}_Q$ be the projection of every vector in S^j_i on to its first component.
   (b) Run the list recovery algorithm for C^j_{out} on set of lists $\{T^j_i\}_i$ obtained from the previous step for up to ξ_j fraction of errors. Store the set of codewords returned in I_j.
   (c) Add $\{(\mathbf{c}, \mathbf{v}) | \mathbf{v} \in I_j\}$ to $\mathcal{L}_j$.

At the end, remove all the matrices $M \in \mathcal{L}_{s-1}$, for which the codeword $(C_{in}(M_1), C_{in}(M_2), \cdots, C_{in}(M_N))$ is at a distance more than δ from $\mathbf{R}$. Output the remaining matrices as the final answer.

We will first talk about the running time complexity of the algorithm. It is easy to check that each repetition of steps 2(a)-(c) takes time $O(T_j(N) + N \cdot t_j(n))$. To compute the final running time, we need to get a bound on number of times step 2 is repeated in phase j. It is easy to check that the number of repetitions is exactly $|\mathcal{L}_{j-1}|$. Thus, we need to bound $|\mathcal{L}_{j-1}|$. By the

list recoverability property of C^j_{out}, we can bound $|I_j|$ by L. This implies that $|\mathcal{L}_j| \leqslant L|\mathcal{L}_{j-1}|$, and therefore by induction we have

$$|\mathcal{L}_i| \leqslant L^{i+1} \quad \text{for } i = 0, 1, \ldots, s-1 \ . \tag{7}$$

Thus, the overall running time and the size of the list output by the algorithm are as claimed in the statement of the theorem.

We now argue the correctness of the algorithm. That is, we have to show that for every $M \in C^0_{out} \times \cdots \times C^{s-1}_{out}$, such that $(C_{in}(M_1), C_{in}(M_2) \cdots, C_{in}(M_N))$ is at a distance at most δ from $\mathbf{R}$ (call such an M a *good* matrix), $M \in \mathcal{L}_{s-1}$. In fact, we will prove a stronger claim: for every good matrix M and every $0 \leqslant j \leqslant s-1$, $M^j \in \mathcal{L}_j$, where M^j denotes the submatrix of M that lies in $C^0_{out} \times \cdots \times C^j_{out}$ (that is the first j "rows" of M). For the rest of the argument fix an arbitrary good matrix M. Now assume that the stronger claim above holds for $j' - 1$ ($< s-1$). In other words, $M^{j'-1} \in \mathcal{L}_{j'-1}$. Now, we need to show that $M^{j'} \in \mathcal{L}_{j'}$.

For concreteness, let $M = (m_0, \cdots, m_{s-1})^T$. As M is a good matrix and $\delta \leqslant \xi_{j'}\rho_{j'}$, $C_{in}(M_i)$ can disagree with $\mathbf{R}_i$ on at least a fraction $\rho_{j'}$ of positions for at most $\xi_{j'}$ fraction of column indices i. The next crucial observation is that for any column index i, $C_{in}(M_i) = (G_{j'})^T \cdot (m_{0,i}, \cdots, m_{j'-1,i}) + (G \setminus G_{j'})^T \cdot (m_{j',i}, \cdots, m_{s-1,i})$, where $G_{j'}$ is as defined in step 2(a), $G \setminus G_{j'}$ is the submatrix of G obtained by "removing" $G_{j'}$ and $m_{j',i}$ is the i^{th} component of the vector $m_{j'}$. The following might help the reader to visualize the different variables.

$$G^T \cdot M = \begin{pmatrix} (G_{j'})^T & (G \setminus G_{j'})^T \end{pmatrix} \cdot \begin{pmatrix} m_{0,1} & \cdots & m_{0,i} & \cdots & m_{0,N} \\ & & \vdots & & \\ m_{j'-1,1} & \cdots & m_{j'-1,i} & \cdots & m_{j'-1,N} \\ m_{j',1} & \cdots & m_{j',i} & \cdots & m_{j',N} \\ & & \vdots & & \\ m_{s-1,1} & \cdots & m_{s-1,i} & \cdots & m_{s-1,N} \end{pmatrix}$$

$$= \begin{pmatrix} \uparrow & & \uparrow & & \uparrow \\ C_{in}(M_1) & \cdots & C_{in}(M_i) & \cdots & C_{in}(M_N) \\ \downarrow & & \downarrow & & \downarrow \end{pmatrix}$$

Note that $G \setminus G_{j'}$ is the generator matrix of $C^{j'}_{in}$. Thus, for at most $\xi_{j'}$ fraction of column indices i, $(m_{j',i}, \cdots, m_{s-1,i}) \cdot (G \setminus G_{j'})$ disagrees with $\mathbf{R}_i - \mathbf{X}_i$ on at least $\rho_{j'}$ fraction of places, where $\mathbf{X}$ is as defined in Step 2(a), and $\mathbf{X}_i$ denotes the i'th column of $\mathbf{X}$. As $C^{j'}_{in}$ is $(\rho_{j'}, \ell)$-list decodable, for at least $1 - \xi_{j'}$ fraction of column index i, $M^{j'}_i$ will be in $S^{j'}_i$ (where $M^{j'}_i$ is M_i projected on it's last $s - j'$ co-ordinates and $S^{j'}_i$ is as defined in Step 2(a)). In other words, $m_{j',i}$ is in $T^{j'}_i$ for at least $1 - \xi_{j'}$ fraction of i's. Further, as $|S^{j'}_i| \leqslant \ell$, $|T^{j'}_i| \leqslant \ell$. This implies with the list recoverability property of $C^{j'}_{out}$ that $m_{j'} \in I_{j'}$, where $I_{j'}$ is as defined in step 2(b). Finally, step 2(c) implies that $M^{j'} \in \mathcal{L}_{j'}$ as required.

The proof of correctness of the algorithm along with (7) shows that C is (δ, L^s)-list decodable, which completes the proof. □

5 List Decoding up to the Blokh-Zyablov Bound

We combine the results we have proved in the last couple of sections to get our main result.

Theorem 5.1 (Main). *For every fixed field $\mathbb{F}_q$, reals $0 < \delta < 1, 0 < r \leqslant 1 - H_q(\delta), \varepsilon > 0$ and integer $s \geqslant 1$, there exists linear codes C over $\mathbb{F}_q$ of block length N that are $(\delta - \varepsilon, L(N))$-list decodable with rate R such that*

$$R = r - \frac{r}{s} \sum_{i=0}^{s-1} \frac{\delta}{H_q^{-1}(1 - r + ri/s)}, \tag{8}$$

and $L(N) = (N/\varepsilon^2)^{O\left(s\varepsilon^{-3}\delta/(H_q^{-1}(1-r)-\delta)\right)}$. Finally, C can be constructed in time $(N/\varepsilon^2)^{O(s/(\varepsilon^6 r\delta))}$ and list decoded in time polynomial in N.

Proof. Let $\gamma > 0$ (we will define its value later). For every $0 \leqslant j \leqslant s-1$ define $r_j = r(1 - j/s)$ and $R_j = 1 - \frac{\delta}{H_q^{-1}(1-r_j)}$. The code C is going to be a multilevel concatenated code $(C_{out}^0 \times \cdots \times C_{out}^{s-1}) \circ C_{in}$, where C_{out}^j is the code from Theorem 2.2 of rate R_j and block length N' (over $\mathbb{F}_{q^a}$) and C_{in} is an $(\langle r_0, \ldots, r_{s-1}\rangle, \rho, \mathbf{L})$-nested list decodable code as guaranteed by Theorem 3.1, where for $0 \leqslant j \leqslant s-1$, $\rho_j = H_q^{-1}(1 - r_j - 2\gamma^2)$ and $L_j = q^{1/\gamma^2}$. Finally, we will use the property of C_{out}^j that it is $(1 - R - \gamma, q^{1/\gamma^2}, (N'/\gamma^2)^{O(\gamma^{-3}\log(1/R_j))})$-list recoverable. Theorem 2.2 implies that such codes exist with (where we apply Theorem 2.2 with $R' = \max_j R_j = 1 - \delta/H_q^{-1}(1 - r/s)$)

$$q^a = (N'/\gamma^2)^{O(\gamma^{-4} H_q^{-1}(1-r/s)/\delta)}. \tag{9}$$

Further, as codes from Theorem 2.2 are $\mathbb{F}_q$-linear [8], C is a linear code.

The claims on the list decodability of C follow from the choices of R_j and r_j and Theorems 2.2, 3.1 and 4.1. In particular, note that we invoke Theorem 4.1 with the following parameters: $\xi_j = 1 - R_j - \gamma$ and $\rho_j = H_q^{-1}(1 - r_j - 2\gamma^2)$ (which implies[2] that $\xi_j \rho_j \geqslant \delta - \varepsilon$ as long as $\gamma = \Theta(\varepsilon)$), $\ell = q^{1/\gamma^2}$ and $L = (N'/\gamma^2)^{O(\gamma^{-1}\log(\ell/R_j))}$. The choices of ℓ and γ imply that $L = (N/\varepsilon^2)^{O(\varepsilon^{-3}\log(1/R_j))}$. Now $\log(1/R_j) \leqslant \log(1/R_{min})$, where $R_{min} = \min_j R_j = 1 - \delta/H_q^{-1}(1-r)$. Finally, we use the fact that for any $0 < y < 1$, $\ln(1/y) \leqslant 1/y - 1$ to get that $\log(1/R_j) \leqslant O(1/R_{min} - 1) = O(\delta/(H_q^{-1}(1-r) - \delta))$. The claimed upper bound of $L(N)$ follows as $L(N) \leqslant L^s$ (by Theorem 4.1).

By the choices of R_j and r_j and (1), the rate of C is as claimed. The construction time for C is the time required to construct C_{in}, which by Theorem 3.1 is $2^{O(n/\gamma^2)}$ where n is the block length of C_{in}. Note that $n = as/r$, which by (9) implies that the construction time is $(N/\varepsilon^2)^{O(\varepsilon^{-6} s H_q^{-1}(1-r/s)/(r\delta))}$. The claimed running time follows by using the bound $H_q^{-1}(1 - r/s) \leqslant 1$.

We finally consider the running time of the list decoding algorithm. We list decode the inner code(s) by brute force, which takes $2^{O(n)}$ time, that is, $t_j(n) = 2^{O(n)}$. Thus, Theorems 2.2, 4.1 and the bound on $L(N)$ implies the claimed running time complexity. □

Choosing the parameter r in the above theorem so as to maximize (8) gives us linear codes over any fixed field whose rate vs. list decoding radius trade-off meets the Blokh-Zyablov bound (2). As s grows, the trade-off approaches the integral form (3) of the Blokh-Zyablov bound.

6 Concluding Remarks

Code concatenation has been instrumental in all the progress in construction explicit binary codes that can be list decoded in polynomial time. However, the best known trade-off between rate and fraction of errors that can corrected via such codes (as we showed in this work) is the Blokh-Zyablov

[2] As for any $0 < x < 1$ and small enough $\alpha > 0$, $H_q^{-1}(x - \alpha^2) \geqslant H_q^{-1}(x) - \Theta(\alpha)$ [11].

bound, which is no where close to the list-decoding capacity. A natural question to ask is if whether concatenated codes can achieve list decoding capacity or is this work the best one can hope for? One reason to suspect the latter would be that the natural decoding algorithm for concatenated codes (which was used in this paper) and its analysis seem to bottom out at the Blokh-Zyablov bound. In recent work [9], we show that there *exist* q-ary linear concatenated codes that achieve list decoding capacity (in the sense that every Hamming ball of radius $H_q^{-1}(1 - R - \varepsilon)$ has polynomially many codewords, where R is the rate). In particular, this results holds when the outer code is a folded RS code. However, realizing the full potential of concatenated codes and achieving capacity (or even substantially improving upon the Blokh-Zyablov bound) with explicit codes and polynomial time decoding remains a huge challenge.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method*. John Wiley and Sons, Inc., 1992.

[2] E. L. Blokh and V. V. Zyablov. *Linear Concatenaed Codes*. Moscow: Nauka, 1982. (in Russian).

[3] I. 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.

[4] G. D. Forney. *Concatenated Codes*. MIT Press, Cambridge, MA, 1966.

[5] V. Guruswami. *List decoding of error-correcting codes*. Number 3282 in Lecture Notes in Computer Science. Springer, 2004. (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition).

[6] V. Guruswami and P. 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.

[7] V. Guruswami and A. 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.

[8] V. Guruswami and A. Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 2007. To Appear. Preliminary version appears in Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC 2006).

[9] V. Guruswami and A. Rudra. Concatenated codes can achieve list decoding capacity. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2008. To appear.

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

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

[12] V. A. Zinoviev. Generalized concatenated codes. *Prob. Peredachi Inform.*, 12(1):5–15, 1976.

[13] V. A. Zinoviev and V. V. Zyablov. Codes with unequal protection. *Prob. Peredachi Inform.*, 15(4):50–60, 1979.

[14] V. V. Zyablov. An estimate of the complexity of constructing binary linear cascade codes. *Problems of Information Transmission*, 7(1):3–10, 1971.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de/