Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets

[Extended Abstract]

Venkatesan Guruswami*

University of California at Berkeley
Computer Science Division
Berkeley, CA 94720

venkat@lcs.mit.edu

Piotr Indyk

MIT Laboratory for Computer Science
200 Technology Square
Cambridge, MA 02139

indyk@theory.lcs.mit.edu

ABSTRACT

We present an *explicit* construction of *linear-time encodable and decodable* codes of rate r which can correct a fraction $(1-r-\varepsilon)/2$ of errors over an alphabet of constant size depending only on ε, for every $0 < r < 1$ and arbitrarily small $\varepsilon > 0$. The error-correction performance of these codes is *optimal* as seen by the Singleton bound (these are "near-MDS" codes). Such near-MDS linear-time codes were known for the decoding from erasures [2]; our construction generalizes this to handle errors as well. Concatenating these codes with good, constant-sized binary codes gives a construction of linear-time binary codes which meet the so-called "Zyablov bound". In a nutshell, our results match the performance of the previously known explicit constructions of codes that had polynomial time encoding and decoding, but in addition have linear time encoding and decoding algorithms.

We also obtain some results for list decoding targeted at the situation when the fraction of errors is very large, namely $(1-\varepsilon)$ for an arbitrarily small constant $\varepsilon > 0$. The previously known constructions of such codes of good rate over constant-sized alphabets either used algebraic-geometric codes and thus suffered from complicated constructions and slow decoding, or as in the recent work of the authors [9], had fast encoding/decoding, but suffered from an alphabet size that was exponential in $1/\varepsilon$. We present two constructions of such codes with rate close to $\Omega(\varepsilon^2)$ over an alphabet of size quasi-polynomial in $1/\varepsilon$. One of the constructions, at the expense of a slight worsening of the rate, can achieve an alphabet size which is polynomial in $1/\varepsilon$. It also yields constructions of codes for list decoding from *erasures* which achieve new trade-offs. In particular, we construct codes of rate close to the optimal $\Omega(\varepsilon)$ rate which can be efficiently list decoded from a fraction $(1-\varepsilon)$ of erasures.

*Supported by a Miller Research Fellowship.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
STOC'02, May 19-21, 2002, Montreal, Quebec, Canada.
Copyright 2002 ACM 1-58113-495-9/02/0005 ...$5.00.

1. INTRODUCTION

Error-correcting codes are combinatorial objects used for the purpose of reliable transmission of information on a noisy communication channel. Two fundamental parameters of a code are its *rate* and its *minimum distance* (or simply, *distance*). The rate of a code is the ratio between the length of the message before encoding to its length after encoding; the larger the rate, the more efficient the code is. The distance of a code is the minimum Hamming distance between a pair of distinct codewords in the code; the larger the distance, the better the error-correction properties of the code. Moreover, in order to use the codes in practice, one also needs efficient algorithms for encoding and decoding the codes. Hence constructing efficiently encodable/decodable codes of large distance and rate is one of the most fundamental problems in coding theory.

In this paper, we present several construction of error-correcting codes with good distance/rate trade-offs and efficient encoding/decoding algorithms. We are interested in unique decoding (where the number of errors that occur is assumed to be less than half of the minimum distance of the code and thus we can uniquely identify the original codeword), as well as in list decoding (where the decoder is allowed to output a list of all codewords with significant match with the received word, thereby enabling decoding beyond the "half of the minimum distance" barrier). In the following, we describe our results in these two scenarios.

Unique decoding. Our first result is a construction of a code family with the best possible asymptotic rate vs. distance trade-off (namely, the *Singleton bound*), together with linear time encoding and linear time decoding algorithms up to half the (designed) distance. Specifically, for each distance, the rate achieved by our code matches the best achievable rate up to a multiplicative factor of $(1+\varepsilon)$, for an arbitrarily small $\varepsilon > 0$. In addition, the codes have constant alphabet size (the size is a function of ε). Thus, our constructions are essentially optimal in three respects: they can achieve any desired distance including a distance arbitrarily close to the codeword length; they achieve rate arbitrarily close to the optimum for each value of the distance; and they are encodable and decodable in linear time. The only criterion with respect to which our codes are sub-optimal is the alphabet size, which (although constant) depends exponentially on $1/\varepsilon$. A result similar to ours was known for the easier problem of decoding from erasures [2], and indeed we use techniques from that paper in our construction.

Concatenating these with good binary inner codes enables us to match the so-called *Zyablov* bound [23] with linear-time decoding. One can also use the multilevel concatenation technique of [4] to get linear-time codes that match the *Blokh-Zyablov* bound as well. We note that these bounds give essentially the best known rate vs. distance trade-off for explicit binary codes.[1] We are able to attain these bounds together with linear time encoding and decoding algorithms. In comparison, the first constructions of linear time encodable/decodable codes [20, 19] could not achieve arbitrarily large distance. While this aspect was fixed in the recent work [9] which obtained linear-time codes of any desired distance, the rate of these codes was sub-optimal.

To summarize, we obtain what was earlier possible with reasonable polynomial time construction complexity with linear time algorithms. Our result is a major step towards the resolution of a project proposed by Spielman [19], namely, to "obtain linear-time encodable and decodable error-correcting codes with rate and error tolerance as good as the best known codes, whatever their algorithmic efficiency".

The technique behind our construction gives a fairly general way of using highly expanding graphs to construct and decode codes from a large fraction of errors. In fact, later in this paper, we use it to construct a *list* decodable code with small alphabet size and good rate. In addition, the technique is also the key element in the recent construction of list-decodable codes that have a linear time encoding and decoding algorithms [11].

List decoding. The techniques developed for the unique decodable codes turned out to be also useful for list decoding. In this setting, we want to decode a corrupted codeword even if a fraction $(1-\varepsilon)$ fraction of the symbols are in error. Since unique decoding is impossible in this setting, we aim to output a small list of codewords which includes every codeword that agrees with the corrupted codeword on a fraction ε of positions.

Several constructions achieving this goal are known by now. These include certain Reed-Solomon (RS) and Algebraic-geometric (AG) codes, which achieve a rate of $\Omega(\varepsilon^2)$ [21, 17, 10]. In addition, AG-codes can achieve this property over an alphabet of constant size (specifically, $O(1/\varepsilon^4)$ size); in comparison, the alphabet size of RS codes is at least as large as their blocklength. With respect to those parameters, AG codes are superior to RS codes. Unfortunately, list decoding (or even construction) of AG codes is a task of high-complexity (see [9] for a detailed description). In comparison, RS codes are explicit and can be list decoded in near-quadratic time. Recently, new codes with constant alphabet size and quadratic decoding time have been developed in [9]. Those codes also achieve a rate of $\Omega(\varepsilon^2)$. However, the drawback of those codes is their alphabet size, which is exponential in $1/\varepsilon$.

By using the technique developed for uniquely decodable codes, we managed to construct a code with $\Omega(\varepsilon^2)$ rate, quadratic decoding time and alphabet size quasi-polynomial in $1/\varepsilon$. Thus, we have made progress towards constructing a single code family achieving optimal list decoding parameters. Unlike the previous codes, however, our codes only come with a *randomized Monte Carlo* construction algorithm. Nevertheless, the construction procedure succeeds with high probability, and the non-explicitness of the code does not hamper the encoding time (which is nearly linear) or the decoding time (which is quadratic).

In order to match (and even beat) the alphabet size guaranteed by AG codes, we developed another set of codes based on the technique of symbol *juxtaposition* (we call these constructions *juxtaposed codes*). Unlike the previous codes, they are explicit, while achieving alphabet size quasi-polynomial in $1/\varepsilon$ at the expense of a slightly worse rate (the rate is $\Omega(\varepsilon^2 \log^{-3}(1/\varepsilon))$). The codes can be decoded in quadratic time. With a different choice of parameters, this construction also yields codes over an alphabet of size $(1/\varepsilon)^t$ while maintaining almost quadratic rate of $\Omega(\varepsilon^{2+2/t})$, for every integer $t \geq 2$. Note that by using $t < 4$ we can achieve codes with sub-cubic rate with alphabet size better than $1/\varepsilon^4$ (which is achieved by the best known AG-codes).

We also consider list decodable erasure codes. Here, to decode a fraction $(1-\varepsilon)$ of erasures, certain AG-codes achieve a rate of $\Omega(\varepsilon)$ over an alphabet of size $O(1/\varepsilon^2)$; the construction is again quite complicated. For binary codes, the best construction is due to [8] and achieves a rate of $\Omega(\varepsilon^2/\log(1/\varepsilon))$. Existentially, the best possible rate is $\Theta(\varepsilon)$, and thus a big gap for binary codes still exists. In this paper, we essentially close this gap for slightly larger alphabets. Specifically, we achieve a rate of $\Omega(\varepsilon^{1+1/t})$ over an alphabet of size 2^t for every integer $t \geq 1$. Thus we approach the optimum bound of $\Omega(\varepsilon)$ by gradually increasing the alphabet size. This construction is also based on juxtaposed codes.

The latter erasure codes codes are non-linear. In fact, this is necessary due to the very good list size (polynomial in $\log(1/\varepsilon)$) that they achieve; it has been shown in [8] that linear codes with these parameters must output a list of size of $\Omega(1/\varepsilon)$. As a corollary we obtain a provable asymptotic separation between the power of linear and non-linear codes for erasure list-decodability, a situation which is quite rare in coding theory.

2. LINEAR-TIME CODES WITH NEAR-OPTIMAL RATE

We now present constructions of linear-time encodable and decodable codes over large alphabets (as well as the binary alphabet) which achieve excellent trade-offs between the rate and the fraction of errors corrected. The codes over the large alphabet get arbitrarily close to the Singleton bound and thus achieve the optimal rate vs. error-correction trade-off (not to forget the linear time encoding and decoding algorithms)! For binary codes, our codes meet the so-called Zyablov bound which is the best trade-off known with reasonable construction complexity (and the best known for concatenated codes).

2.1 High-level view of the construction

Once we construct codes over the large alphabet that match the Singleton bound, getting binary codes that meet the Zyablov bound is easy and follows by concatenation with a constant-sized inner code that meets the "Gilbert-Varshamov bound". So, let us focus on the construction of the large alphabet code.

[1]The only better bound known for constructive code is due to Katsman *et al* [12]—but this construction needs very good algebraic-geometric codes over a small alphabet and has very high construction and decoding complexity; see [5] for further discussion about this point.

Our code is constructed by combining three objects, a "left" code C, a constant-sized MDS (say, Reed-Solomon) code $\hat{C}$, and a suitable bipartite expander graph G (say, with n vertices on each side). The message will be first encoded by the left code C. The resulting codeword of C will then be broken into n blocks, each of constant size, and each of these blocks will be encoded by the Reed-Solomon code $\hat{C}$. The symbols of the resulting string will then be redistributed using the edges of the expander G, the symbols in the encoding of the i'th block being sent to the neighbors of the i'th node on the left side of G. Now, the final codeword (of length n) is obtained by "juxtaposing" or "concatenating" the symbols received at each of the n vertices on the right. The construction scheme is similar in spirit to earlier expander-based code constructions in [1, 2, 9], and specifically the construction of near-MDS erasure codes in [2].

We now elaborate a bit on how we pick each of these components. The left code C will be a linear-time code of rate very close to one, say, $(1-\gamma)$ for some small $\gamma > 0$, which can correct a fraction $\Theta(\gamma^2)$ of errors in linear time. The code $\hat{C}$ will be a Reed-Solomon code of rate (very close to) r. Its blocklength will be equal to the degree D of the expander. For the graph G, we can take any expander whose second eigenvalue λ is much smaller than its degree D; in order to get the best parameters (specifically, alphabet size), we use a Ramanujan graph which satisfies $\lambda = O(\sqrt{D})$.

The code $\hat{C}$ and the expander are standard and we just use them "off-the-shelf". For the left code C, the existing construction of linear-time encodable/decodable codes due to Spielman [20, 19] do not give this directly, as even to correct a very small fraction of errors, the rate has to be an absolute constant bounded away from 1. However, as Spielman [19] remarks it is possible to pick parameters differently in his construction and achieve any rate, though we suspect the dependence between the rate and fraction of errors corrected will not be great. Here, we present a new construction which has the property necessary to us and achieves a reasonable trade-off; our construction is obtained by combining ideas from [2] and [22] in a novel way. The construction of this left code is in fact our main technical contribution in this section, as the rest of the construction pretty much follows the scheme used in [2] for near-MDS erasure codes. We discuss this construction next, before moving on to the construction of the final near-MDS code.

2.2 Linear-time codes with rates close to 1

LEMMA 1. *For every $\gamma > 0$, there is an explicitly specified code of rate $1/(1+\gamma)$ over an alphabet of size $q = \Omega(1/\gamma^2)$ such that a code of blocklength N in the family can be encoded in $O(N/\gamma)$ time and can be decoded from a fraction $\beta = O(\gamma^2)$ of errors in $O(N/\gamma^2)$ time.*

Proof: For infinitely many values of m and for some fixed $q = O(1/\gamma^2)$, we will construct a code over $\mathrm{GF}(q)$ of dimension m and blocklength $(1+\gamma)m$ which can be encoded in linear time and can be decoded from βm errors in linear time for $\beta = \Theta(\gamma^2)$. The encoding will work in two steps. In the first step, the message is encoded by a code C_1 into a string of length $(1+\gamma')m$ comprising of the m message symbols and $\gamma' m$ check symbols (we take $\gamma' = \gamma/4$). This code has the property that given the correct values to all of the check symbols, an arbitrary set of βm errors in the message symbols can be corrected. In the second step, the check symbols are further encoded by a linear-time code C_2 which can correct up to βm errors. The combined code thus maps m symbols into $(1+4\gamma')m = (1+\gamma)m$ symbols and can correct up to βm errors. The decoding algorithm for the combined code from βm errors is the obvious one: first decode C_2 to correct any errors in the check bits, and then decode C_1 to correct, using the correct values of the check bits, the up to βm errors that could exist in the message bits.

For the code C_2, we can use the codes due to Spielman which have some constant rate. Specifically, as stated in [2], there is an explicit such code C_2 over $\mathrm{GF}(q)$ of rate $1/4$ which can correct a b fraction of errors for some absolute constant $b > 0$.

It remains to describe the code C_1. The code C_1 must encode m symbols into $(1+\gamma')m$ symbols such that the encoding can be performed in linear time and moreover C_1 can be decoded from up to βm errors in the message bits, where $\beta = O(\gamma'^2)$, in linear time.

Let H be a d-regular bipartite "Ramanujan" expander with m edges and $n' = m/d$ vertices on each side, such that the second largest eigenvalue λ of its adjacency matrix satisfies $\lambda \le 2\sqrt{d}$. Here d is a constant that is independent of n', i.e., we use a family of constant-degree expanders (jumping ahead $d = \Omega(1/\gamma^2)$ will suffice). The m positions of the message to be encoded are identified with the edges of H. For each vertex v of H, we compute $\gamma' d$ check symbols corresponding to the message symbols on edges incident upon v. These are computed using some systematic MDS code $\tilde{C}$ of dimension d, blocklength $(1+\gamma')d$, and which can correct fewer than $\gamma' d/2$ errors; for example we can use a Reed-Solomon code over a field of size $O(d)$. In all, this gives $\gamma' nd = \gamma' m$ check symbols, as required.

It is clear that C_1 can be encoded in linear time, since each of the n MDS codes is of constant-size. We now discuss the linear-time decoding algorithm for C_1 that corrects up to βm errors in the message symbols, given the correct values of all check symbols. This algorithm and its analysis follows along the lines of Zemor's recent improvement [22] of the analysis of Sipser and Spielman [18]. For completeness sake, we next present the details of this analysis.

Let the two sides of the bipartition of H be A and B. For each $v \in A \cup B$ denote by E_v the set of edges of H incident on v. Let $x \in \mathrm{GF}(q)^m$ be the portion of the received word corresponding to the m message symbols — by hypothesis, x is the message vector corrupted by at most βm errors. Let $y \in \mathrm{GF}(m)^{\gamma' m}$ be the vector of the check symbols. Denote by x_{E_v} the projection of x on the d edges in E_v, and by y_{E_v} the projection of y to to the $\gamma' d$ check symbols that correspond to the encoding by the MDS code $\tilde{C}$ of the symbols on the edges in E_v. The decoding algorithm proceeds in rounds, and in each round does the following:

(a) (Left wing decoding) For each $v \in A$ in parallel, check if there exists a vector $z \in \mathrm{GF}(q)^d$ within distance $\gamma' d/2$ of x_{E_v} and whose check bits agree with y_{E_v}; if so, set x_{E_v} to z.

(b) (Right wing decoding) For each $v \in B$ in parallel, check if there exists a vector $z \in \mathrm{GF}(q)^d$ within distance $\gamma' d/2$ of x_{E_v} and whose check bits agree with y_{E_v}; if so, set x_{E_v} to z.

To analyze the algorithm, by linearity it suffices to consider the case when the correct message is the all-zeroes string (which also implies that all check symbols equal 0). Let $X = \{e : x_e \neq 0\}$ be the set of edges whose symbols are in error in the original received word x. For $i \geq 1$, let $Y^{(i)}$ (resp. $Z^{(i)}$) be the set of edges in error, i.e. edges e so that $x_e \neq 0$, after the left wing (resp. right wing) of the i'th round of decoding (we use the convention $Y^{(0)} = Z^{(0)} = X$). Define the set $A^{(i)}$ and $B^{(i)}$ for $i \geq 1$ as follows:

- $A^{(i)} = \{v \in A : E_v \cap Y^{(i)} \neq \emptyset\}$
- $B^{(i)} = \{v \in B : E_v \cap Z^{(i)} \neq \emptyset\}$

Now comes the crucial part of the analysis. Let $i \geq 1$ be fixed. For each $v \in A^{(i)}$ (i.e., vertices on the left which are incident to some uncorrected edge after the left wing decoding of the i'th round), we have $|E_v \cap Z^{(i-1)}| \geq \gamma' d/2$, as otherwise the left wing decoding of the i'th round would have corrected the fewer than $\gamma' d/2$ errors that remained in the edges of E_v. We also have, for the same reason, $|E_v \cap Y^{(i)}| \geq \gamma' d/2$ for every $v \in B^{(i)}$.

Our goal is to now prove that the size of the $A^{(i)}$'s and $B^{(i)}$'s decreases geometrically, which will imply that the algorithm converges in $O(\log n)$ rounds. Note that this does not immediately imply a linear upper bound on the decoding time, since each round itself appears to require linear runtime. However, there is a linear-time implementation of the algorithm by carefully considering only geometrically decreasing subsets of A, B when implementing the successive decoding rounds. We omit the details and point the reader to, for example, [19, Sec. 2.5] or [3], where explicit details on this aspect appear.

Now, consider the subgraph of H induced by the edges in $Y^{(i)}$. By definition, each such edge must be incident upon a vertex in $A^{(i)}$. Furthermore, every vertex in $B^{(i)}$ is incident upon at least $\gamma' d/2$ edges of $Y^{(i)}$. Applying Lemma 2 stated at the end of this section to this situation (with the choice $S = A^{(i)}$, $T = B^{(i)}$ and $Y = Y^{(i)}$), the expansion property of the graph H implies that $B^{(i)}$ has to be small provided $A^{(i)}$ is small. Specifically, $|B^{(i)}| \leq \zeta |A^{(i)}|$ for some $\zeta < 1$, provided $|A^{(i)}| \leq \rho n \big(\frac{\gamma'}{4} - \frac{2}{\sqrt{d}}\big)$ for some $\rho < 1$. This condition will be satisfied provided $d \geq 64/\gamma'^2$ and $|A^{(i)}| \leq \gamma' n/16$. By the same argument, we will also have $|A^{(i+1)}| \leq \zeta |B^{(i)}|$ for $i \geq 1$. Hence, we would have proved the geometrically decreasing property, provided we can get an upper bound of $\gamma' n/16$ on $|A^{(1)}|$ to start with.

By definition each vertex of $A^{(1)}$ is adjacent to at least $\gamma' d/2$ erroneous edges, and hence we have $|X| \geq |A^{(1)}| \gamma' d/2$. Also, by hypothesis there are at most $\beta m = \beta n d$ errors, and so $|A^{(1)}| \leq \frac{2\beta n}{\gamma'}$. Therefore, if $\beta \leq \gamma'^2/32$, then $|A^{(1)}| \leq \gamma' n/16$ as desired. $\square$ *(Lemma 1)*

LEMMA 2 ([22]). *Let $\rho < 1$ be arbitrary. Let $H = (A, B, E)$ be a d-regular bipartite expander with n vertices on each side and whose adjacency matrix has second largest eigenvalue $\lambda \leq d/3$. Let S be a subset of vertices of A such that $|S| \leq \rho n \Big(\frac{\alpha}{2} - \frac{\lambda}{d}\Big)$. Let T be a subset of vertices of B and suppose that there exists a set $Y \subseteq E$ of edges such that: (a) every edge in Y has one of its endpoints in S, and (b) every vertex in T is incident to at least αd edges of Y. Then, $|T| \leq \frac{1}{2-\rho}|S|$.*

2.3 Linear-time error-correcting codes meeting the Singleton bound

We now use the codes from the previous section as the "left code" in our general construction scheme to obtain linear time encodable/decodable codes whose rate vs. error-correcting trade-off approaches the Singleton bound (we call such codes *near-MDS* codes). Below we state a more general result that handles both errors and erasures. This will help us deduce the result for binary codes in the next section very easily, since the GMD algorithm for concatenated codes that we will employ requires an errors-and-erasures decoding algorithm for the outer code.

THEOREM 3. *For every r, $0 < r < 1$, and all sufficiently small $\varepsilon > 0$, there exists an* explicitly *specified family of* GF(2)*-linear (also called additive)*[2] *codes of rate r over an alphabet of size $2^{O(\varepsilon^{-4} r^{-1} \log(1/\varepsilon))}$ such that codes from the family can be encoded in linear time and can also be (unique) decoded in linear time from a fraction e of errors and s of erasures provided $2e + s \leq (1 - r - \varepsilon)$.*

Proof: We will use the construction outlined in Section 2.1 with left code being the code from Lemma 1 for the choice $\gamma = \varepsilon/4$. Let x be a message of length m over GF(q) for some constant q (jumping ahead, q will be a power of two large enough for the left code and the Reed-Solomon code $\hat{C}$ to exist). The message is first encoded by C to give a string $y = C(x)$ of length $n' = (1 + \varepsilon/4)m$ over GF(q). We assume, by Lemma 1, that C can correct $\beta n'$ errors in linear time for $\beta = O(\varepsilon^2)$. The symbols of y will be broken up into $n = n'/b$ blocks consisting of b symbols each for a block size $b = \Theta(1/\varepsilon^4)$. Each of these n blocks will undergo encoding by a Reed-Solomon code over GF(q) of dimension b and rate $r' = r(1 + \varepsilon/4)$, to give n blocks $B_1, \ldots, B_n$ each consisting of $\Delta = b/r'$ symbols over GF(q) (if we pick $q = \Omega(r^{-1}\varepsilon^{-4}) \geq \Delta$, both the left code as well as the Reed-Solomon code will exist over an alphabet of size q).

Let $G = (A, B, E)$ be a Δ-regular bipartite expander with n vertices on each side with the following property:

(*) For every subset $X \subset A$ with $|X| \geq \beta n/2$ and every $Y \subseteq B$, we have $\left| \frac{|E(X:Y)|}{|X|\Delta} - \frac{|Y|}{|B|} \right| \leq \varepsilon/4$.

One can show that Ramanujan graphs, namely graphs whose second largest eigenvalue satisfies $\lambda = O(\sqrt{\Delta})$, of degree $\Delta = O(1/\beta\varepsilon^2) = O(1/\varepsilon^4)$, give bipartite graphs with the above property (see for example [9] or [7, Chap. 11] for a proof). Explicit constructions of Ramanujan graphs are known [13] and since $C, \hat{C}$ are explicitly specified as well, our overall construction is explicit.[3] The symbols of the i'th block will be redistributed to the neighbors of the i'th

[2] A code C over a field of characteristic 2 is said to GF(2)-linear or additive if $x + y \in C$ whenever both $x \in C$ and $y \in C$.

[3] Here we assume that parameters have been so picked that there is an explicit Ramanujan graph, eg. the construction of [13], with exactly n vertices. Since there is a lot of flexibility in the choice of parameters of the left code C and the Reed-Solomon code $\hat{C}$, and since the sequences of vertex sizes of known explicit constructions of Ramanujan graphs form a dense sequence, this can be easily ensured. For sake of simplicity, we ignore this issue and simply assume that expanders with exactly the required number of vertices exist.

vertex on the left side of G (the j'th symbol going to the j'th neighbor of the vertex, for $1 \le j \le \Delta$, as per some arbitrary ordering of the neighbors of each vertex). This gives, for each vertex on the right side, a collection of Δ $\mathrm{GF}(q)$-symbols obtained from its Δ neighbors on the left, which, equivalently, can be viewed as a single symbol over $\mathrm{GF}(q^\Delta)$. The string (of length n) consisting of these symbols forms the encoding of x by our final code over $\mathrm{GF}(q^\Delta)$, call it C^*.

RATE AND ALPHABET SIZE. This gives a code over an alphabet of size $q^\Delta = 2^{O(b \lg q/r')} = 2^{O(\varepsilon^{-4} r^{-1} \log(1/\varepsilon))}$ and which has rate $\frac{m/\Delta}{n} = \frac{mr'}{bn} = \frac{mr'}{n'} = r$ (since $n' = (1+\varepsilon/4)m$ and $r' = r(1+\varepsilon/4)$). It is also clear that C^* has a linear time encoding algorithm.

DECODING COMPLEXITY. Using the Property (*) of G, it is also easy to show that the relative distance of C^* is at least $(1 - r - \varepsilon/2)$. In fact, we next prove that C^* can be unique decoded from a fraction e of errors and s of erasures provided $2e + s \le (1 - r - \varepsilon)$.

Let z be a received word for C^* with a fraction s of erasures and a fraction e of errors, where $2e + s \le (1 - r - \varepsilon)$. Since the relative distance of C^* is greater than $(1 - r - \varepsilon)$, there is a unique message x that is solution to the decoding problem. Let S be the set of erasures in the received word z, and let F be the set of errors (i.e., the positions where $C^*(x)$ and z differ). We have $|S| = sn$ and $|F| = en$.

Given the received word z, the decoding algorithm proceeds as follows. In the first step, the word z is used to compute certain "received words" z_i, $1 \le i \le n$, for the n encodings by $\hat{C}$ (corresponding to the n blocks into which a codeword of C is broken into). This is done as follows. For each i, j, $1 \le i \le n$ and $1 \le j \le \Delta$, if the j'th neighbor of the i'th node of A has an unerased symbol, say $\zeta \in \mathrm{GF}(q^\Delta)$, then the j'th symbol of z_i is set to the symbol in the appropriate coordinate of ζ (namely, the coordinate which received that symbol through the expander). If the j'th neighbor of the i'th node of A has an erased symbol, then we declare an erasure at the j'th position of z_i.

For each i, $1 \le i \le n$, let z_i be the received word thus obtained for the encoding of i'th block. Let s_i be the fraction of positions in z_i which are erased, and let e_i be the fraction of positions in z_i which are set to a wrong symbol. With the z_i's computed, the algorithm continues as follows. For each i, we run a unique error-erasure decoding algorithm for the Reed-Solomon code $\hat{C}$ with received word z_i. If it succeeds in decoding, we let $y_i \in \mathrm{GF}(q)^b$ be the message it outputs, otherwise we let y_i be an arbitrary string in $\mathrm{GF}(q)^b$. Finally, the decoding is completed by running the linear time unique decoding algorithm for C on the received word $y = \langle y_1, y_2, \ldots, y_n \rangle$, and outputting whatever message x it outputs.

It is clear that the algorithm runs in linear time. We now prove the correctness of this procedure. We claim that it suffices to prove that the received words z_i (obtained from the first stage of the decoding that uses the expander) satisfy $2e_i + s_i < (1 - r - \varepsilon/4)$ for at least $(1-\beta)n$ values of i. Indeed, for any such i, the Reed-Solomon decoder will succeed in finding the correct block y_i (as the relative distance of each Reed-Solomon code is at least $(1 - r(1+\varepsilon/4)) \ge 1 - r - \varepsilon/4$). Hence the received word y passed to the decoding algorithm for C will agree with $C(x)$ entirely on a fraction $(1-\beta)$ of the blocks, or in other words y and $C(x)$ will differ in at most $\beta n'$ positions. Since the assumed decoding algorithm for C can correct up to a fraction β of errors, we will correctly find and output the message x.

It remains to prove that $2e_i + s_i < (1 - r - \varepsilon/4)$ for all but βn values of i. Define $X' \subset A$ to be the set of nodes which have at least a fraction $(s + \varepsilon/4)$ of neighbors in the set S (the set of erasures in the received word z). Also define $X'' \subset A$ to the nodes which have at least a fraction $(e + \varepsilon/4)$ of neighbors in F (the set of erroneous positions in z). It easily follows from the Property (*) of the expander G that $|X'|, |X''| \le \beta n/2$.

Now consider any node $i \in A \setminus (X' \cup X'')$. It has less than a fraction $(e + \varepsilon/4)$ of neighbors in F. These correspond to the errors in the received word z_i, and hence we have

$$e_i < e + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') \ . \tag{1}$$

A node $i \in A \setminus (X' \cup X'')$ also has less than a fraction $(s + \varepsilon/4)$ of neighbors in S. These correspond to the erasures in the received word z_i, and hence we have

$$s_i < s + \varepsilon/4 \quad \text{for every } i \in A \setminus (X' \cup X'') \ . \tag{2}$$

Since $2e + s \le (1 - r - \varepsilon)$ by hypothesis, we have, combining (1) and (2) that $2e_i + s_i < (1 - r - \varepsilon/4)$, for each $i \in A \setminus (X' \cup X'')$. Since $|X'|, |X''| \le \beta n/2$, we have proved that the condition $2e_i + s_i < (1 - r - \varepsilon/4)$ holds for all but a fraction β of i's in the range $1 \le i \le n$. This completes the proof of correctness of the decoding algorithm. □

2.4 Binary linear-time codes meeting the Zyablov bound

We now state our result for binary codes. The result is obtained by concatenating the near-MDS codes from Theorem 3 with a binary inner code which meets the Gilbert-Varshamov bound. Such a code can be constructed by picking a linear code at random and checking that it has the necessary distance property, or a deterministic construction can be obtained by searching for the inner code (since it is of constant size, this takes only $O(1)$ time). Linear time encoding is clear, and for decoding we use Generalized Minimum Distance (GMD) decoding [6], which decodes a concatenated code up to the "product bound" (i.e., half the product of the designed distances of the outer and inner codes) by running several instances of the errors-and-erasures algorithm for the outer near-MDS code. The number of such runs needed is bounded by a constant since the inner code is of constant size, and since each run takes linear time by Theorem 3, the overall decoding time is linear. We omit the full details and state the final result below. The reader may find all the details about how GMD decoding together with Theorem 3 will give the result below, by referring to similar results in [9] or [7, Chap. 11].

THEOREM 4. *For every $\varepsilon > 0$ and for any code rate $0 < R < 1$, there exists a family of* binary linear *concatenated codes of rate R which can be encoded in linear time and can be decoded in linear time from up to a fraction e of errors, where*

$$e \ge \max_{R<r<1} \frac{(1 - r - \varepsilon) H^{-1}(1 - R/r)}{2} \tag{3}$$

($H^{-1}(y)$ is defined to be the unique x in the range $0 \le x \le 1/2$ that satisfies $H(x) = y$). Every code in the family is explicitly specified given a constant sized binary linear code

which can be constructed in probabilistic $O(\varepsilon^{-4}\log(1/\varepsilon))$ or deterministic $2^{O(\varepsilon^{-4}\log(1/\varepsilon))}$ time.

The bound of Equation (3) is half the Zyablov bound [23], and thus these codes match the best error-correction performance known for constructive binary concatenated codes. We remark that the first explicit construction of codes meeting the Zyablov bound for all rates was due to Shen [16].

2.5 Beyond Zyablov bound with linear-time codes using multilevel concatenation

By using the multilevel generalizations of concatenated codes (see [5] for a detailed discussion) we can construct linear-time binary codes which get arbitrarily close to the so-called *Blokh-Zyablov* bound [4] and thus lie above the Zyablov bound. Specifically, for any $\varepsilon > 0$, we can correct a fraction e of errors with rate

$$R \geq 1 - \varepsilon - H(2e) - 2e \int_0^{1-H(2e)} \frac{dx}{H^{-1}(1-x)} .$$

Further details will appear in the full version of the paper.

3. CONSTRUCTIONS FOR LIST DECODING FROM ERRORS

So far, we have been concerned with unique decoding codes up to a number of errors which is less than the half-the-distance bound. We now turn to constructions of codes that can be efficiently *list decoded* even when most of the received symbols are corrupted, specifically when a fraction $(1-\varepsilon)$ of the symbols are in error. Our goal here is to achieve the previously best known rate of $\Omega(\varepsilon^2)$ over a constant-sized alphabet together with efficient decoding algorithms. This was achieved in the recent work of the authors [9], but the drawback of this scheme was the high (exponential in $1/\varepsilon$) alphabet size. In this section, we present two new constructions which enable us to achieve a similar rate and decoding complexity with a much smaller (quasi-polynomial or polynomial in $1/\varepsilon$, depending upon the actual construction) alphabet size.

Definition 1. *A q-ary code C of blocklength n is said to be (e, L)*-list decodable *if every Hamming ball of radius e has at most L codewords, i.e., $\forall\ \mathbf{x} \in [q]^n$, $|B(\mathbf{x}, e) \cap C| \leq L$.*

3.1 A probabilistic construction with quasi-polynomial alphabet size

We begin with two definitions that will be necessary for this section.

Definition 2. *A bipartite graph $G = (A, B, E)$ with $|A| = n$, $|B| = m$, left degree d and right degree nd/m is called an $(n, m, d, \alpha, \varepsilon, \varepsilon')$-expander, if for any $Y \subset B$, $|Y| \geq \varepsilon|B|$, the fraction of nodes in A which have less than $\varepsilon' d$ neighbors in Y, is at most α. We do not require G to be simple and it might have multiple edges between some pairs of vertices.*

Definition 3. *For α, $0 < \alpha < 1$, and integers $L \geq \ell \geq 2$, a q-ary code C of blocklength n is said to be (α, ℓ, L)*-list recoverable *if given arbitrary "lists" $L_i \subseteq [q]$ of size at most ℓ for each i, $1 \leq i \leq n$, the number of codewords $\mathbf{c} = \langle c_1, \ldots, c_n\rangle \in C$ such that $c_i \in L_i$ for at least αn values of i, is at most L.*

The main result of this section is the following:

Theorem 5. *For every sufficiently small $\varepsilon > 0$, there exists a code family with the following properties:*

(i) It has rate $\Omega(\varepsilon^2)$ and is defined over an alphabet of size $2^{O(\log^2(1/\varepsilon))}$.

(ii) A Monte Carlo construction of a code of blocklength N in the family that has the list decodability property claimed in (iii) with high probability can be obtained in $O(N \log N \log^{O(1)}(1/\varepsilon))$ time.

(iii) A code of blocklength N is the family can be list decoded from up to $(1-\varepsilon)N$ errors in $O(N^2(\log N/\varepsilon)^{O(1)})$ time using lists of size $O(1/\varepsilon)$.

The above code construction is obtained by plugging in specific components to the following general way of constructing codes:

Lemma 6. *Assume we are given:*

- *a q_1-ary code C_1, with rate r_1 and blocklength n, which is $(1/2, \ell, L)$-recoverable in time $T(n)$*
- *a q_2-ary code C_2, with rate r_2 and blocklength d s.t. $\log q_1 \leq dr_2 \log q_2$, which is $((1-\varepsilon/2)d, \ell)$-list decodable*
- *an $(n, m, d, 1/2, \varepsilon, \varepsilon/2)$-expander G*

Then we can construct a code C which is explicitly specified given C_1, C_2, G and which

- *has blocklength m, rate $r_1 \cdot r_2$ and alphabet size $q_2^{nd/m}$*
- *is $((1-\varepsilon)m, L)$-list decodable in $O(T(n) + n2^{dr_2 \log q_2})$ time*

Proof: We construct C in the same way as we constructed the near-MDS code in the previous section, namely the message is first encoded using C_1 and then each of the resulting symbols is encoded using C_2. The symbols of the resulting string are re-distributed according to the graph G.

For simplicity, assume that the number of bits needed to represent a symbol of C_1 (i.e., $\log q_1$) is equal to the message length of code C_2 (i.e., $dr_2 \log q_2$); otherwise, we merge several symbols in codewords from C_2 into one block, which is later encoded by C_2. For any codeword $c' \in C_1$, we encode each of symbols of c' using the code C_2. Then the symbols of the resulting vectors are distributed according to the graph G. Specifically, the j-th symbol of the codeword is a concatenation, over all $(i, j) \in E$, of the k-th symbol of $C_2(c'_i)$, such that j is the k-th neighbor of i as per some fixed order.

The decoding procedure is straightforward. Let y be such that $C(y)$ matches the corrupted codeword in at least ε fraction of positions. For each node $i \in A$, we retrieve a sequence x_i of d symbols by inverting the above procedure. By the properties of G, at least $n/2$ nodes in A retrieve at least $\varepsilon d/2$ correct symbols (given that ε fraction of the symbols of the retrieved message are correct). Each node i then decodes its sequence x_i using the decoding algorithm for C_2, producing a length-ℓ list of candidate symbols for $C_1(y)_i$. Clearly, at least $1/2$ fraction of lists contains the right symbol of $C_1(y)$. Thus we can recover a list of L vectors (containing y) by applying the recovering algorithm for C_1. $\square$ *(Lemma 6)*

We will use the above lemma to obtain a code C from the following components:

(i) A Reed-Solomon code C_1 with rate $\Theta(\varepsilon)$ over an alphabet of size $q_1 = O(n)$ which is $(1/2, O(1/\varepsilon), O(1/\varepsilon))$-list recoverable in $O(n^2(\log n/\varepsilon)^{O(1)})$ time. The existence of a code with these properties follows from the

Reed-Solomon list decoding algorithms in [21, 10] and their fast implementations from [15, 14].

(ii) A *pseudolinear* code C_2 with rate $\Theta(\varepsilon)$, blocklength d, and alphabet size $q_2 = O(1/\varepsilon^2)$ which is $((1-\varepsilon/2)d, O(1/\varepsilon))$-list decodable. Such a code exists and has an efficient $O(d^2\varepsilon^{-O(1)})$ time probabilistic construction by the results of [9] (see also Lemma 9 for a formal statement of this claim).

(iii) An $(n, nd/s, d, 1/2, \varepsilon, \varepsilon/2)$-expander G where $s = O(\log(1/\varepsilon))$, and $d = \Theta(\log n/(\varepsilon\log(1/\varepsilon)))$ is picked so that $\log q_1 = \log n + O(1)$ is equal to $dr_2 \log q_2 = \Theta(d\varepsilon\log(1/\varepsilon))$. A randomized Monte Carlo construction of such a graph G follows from Lemma 7 below.

The claim of Theorem 5 now follows by applying Lemma 6 to the above choice of C_1, C_2 and G. The rate will be $\Omega(\varepsilon^2)$, and alphabet size equals $q_2^s = (1/\varepsilon)^{O(\log(1/\varepsilon))}$. The construction time is dominated by that to construct the graph G which takes probabilistic $O(nd\log(nd)) = O(N\log(1/\varepsilon)\log N)$ time (since the blocklength of C equals $N = nd/s$). The claimed list decodability property also follows easily from Lemma 6. Finally, we state the lemma which guarantees the necessary expander; the proof is based on an application of the probabilistic method and is omitted.

LEMMA 7. *There exists a constant $c > 0$ such that for every small enough $\varepsilon > 0$, if $d > c/\varepsilon$ and $s = \Theta(\log(1/\varepsilon))$, then for all large enough n, there exists a bipartite graph G which is an $(n, nd/s, d, 1/2, \varepsilon, \varepsilon/2)$-expander. Furthermore, a graph which has the expansion property with high probability can be obtained in probabilistic $O(dn\log(dn))$ time.*

3.2 Constructions using juxtaposed codes

In this section, we indicate how one can improve the alphabet size significantly at the expense of a moderate worsening of the rate, by using an entirely different technique (the technique is also somewhat simpler, as it avoids the use of expanders). The basic idea is use to several concatenated codes, each one of which is "good" for some error distribution, and then "juxtapose" symbols from these codes together to obtain a new code over a larger alphabet which has nice list decodability properties. More concretely, in this approach, multiple Reed-Solomon codes (of varying rates) are concatenated with *several* different inner codes (of varying rate and list decodability). Corresponding to each Reed-Solomon and inner code pair, we get one concatenated codeword, and the final encoding of a message is obtained by "juxtaposing together" the symbols from the various individual concatenated codewords.

The purpose of using multiple concatenated codes is that depending on the distribution of errors in the received word, the portions of it corresponding to a significant fraction of a *certain* inner encoding (that depends on the level of non-uniformity in the distribution of errors) will have relatively few errors. These can then be decoded to provide useful information about a large fraction of symbols to the decoder of the *corresponding* outer Reed-Solomon code. Essentially, depending on how (non)-uniformly the errors are distributed, a certain concatenated code "kicks in" and enables recovery of the message. The use of multiple concatenated codes reduces the rate compared to the expander based constructions, but we gain in the alphabet size. For example, for a near-quadratic (namely, $\Omega(\varepsilon^2\log^{-O(1)}(1/\varepsilon))$ rate, the alphabet size can be quasi-polynomial as opposed to exponential in $1/\varepsilon$.

We first discuss the basic code construction scheme, and then formally state the results we obtain for appropriate settings of parameters. Let $\varepsilon > 0$ be given; the goal being to construct a code family of good rate (as close to $\Omega(\varepsilon^2)$ as possible) that can be efficiently decoded from up to a fraction $(1-\varepsilon)$ of errors. Let $t \geq 1$ be an integer parameter (t will be the number of codes that will be juxtaposed together).

Let $\delta_0, \delta_1, \ldots, \delta_t$ be a sequence in geometric progression with $\delta_0 = \varepsilon/2$, $\delta_t = 1$, and $\delta_i/\delta_{i-1} = \Delta$ for $1 \leq i \leq t$. Note that these parameters must satisfy $\Delta^t = 2/\varepsilon$.

Fix $c > 1$ and let $q_0 = O(1/\varepsilon^c)$ be a prime power. Let m be a large enough integer. The juxtaposed code construction, say C^*, that we now give, will be parameterized by $(q_0, m, \varepsilon, t, \Delta)$.

For each i, $1 \leq i \leq t$, we will have one q_0-ary concatenated code $\mathbf{C}_i$ with outer code a Reed-Solomon code C_i^{RS} and inner code an appropriate q_0-ary *pseudolinear code* C_i^{in}. Pseudolinear codes were introduced in [9] and have the very useful property that a pseudolinear code drawn from a suitable family have the property of general random codes for purposes of list decoding, and yet pseudolinear codes have succinct representations and efficient encoding. We refer the reader to [9] for details on pseudolinear codes; for our purpose here, the exact definition of pseudolinear codes is not important, and the codes we need are formally detailed in Lemma 9. The parameters of the outer and inner codes will be as follows.

THE REED-SOLOMON CODES. The blocklength of each of the Reed-Solomon codes will be the same, $n_0 = q_0^m$. The code C_i^{RS} will be defined over the alphabet $\mathrm{GF}(q^{m_i})$ where $m_i = m\delta_i/\delta_0$. The rate of C_i^{RS} will be $R_i = \Theta(\varepsilon^2/(t^2\delta_i\Delta))$ and its dimension will be $k_i = R_i n_0$ (the reason for this choice of rate will become clear once we specify the decoding algorithm in the proof of Theorem 8 below). Note that each message that is encoded by C_i^{RS} consists of k_i symbols over $\mathrm{GF}(q_0^{m_i})$, or equivalently, $k_i m_i = R_i n_0 m_i = \Theta(\frac{\varepsilon m n_0}{t^2\Delta})$ symbols over $\mathrm{GF}(q_0)$. This quantity is independent of i, and hence the number of q_0-ary symbols in the message of each C_i^{RS} can be made equal. This is very useful for juxtaposing the codes together, as it makes sure that the dimension of each of the concatenated codes $\mathbf{C}_i$ will be the same.

THE INNER CODES. The blocklength of each inner code C_i^{in} will be the same, say n_1. Note that this ensures that each one of the concatenated codes $\mathbf{C}_i$ has identical blocklength $N \stackrel{\mathrm{def}}{=} n_0 n_1$. The dimension of C_i^{in} will be m_i, so that it can be concatenated with the Reed-Solomon code C_i^{RS} (that was defined over $\mathrm{GF}(q_0^{m_i})$). The code C_i^{in} will be a pseudolinear code of rate $r_i = m_i/n_1 = \Omega(\delta_{i-1})$ which is $((1-\delta_{i-1})n_1, O(1/\delta_{i-1}))$-list decodable (the exact codes are those guaranteed by Lemma 9). The choice of parameters implies that the blocklength n_1 equals $n_1 = O\left(\frac{m_i}{\delta_{i-1}}\right) = O\left(\frac{m\delta_i}{\delta_0\delta_{i-1}}\right) = O\left(\frac{m\Delta}{\varepsilon}\right)$, which is independent of i and can thus be made identical for each of the inner codes C_i^{in}.

THE JUXTAPOSED CODE. The final code, call it C^*, will be the juxtaposition of the codes $\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_t$. Formally, by this we mean that to encode a message according to C^*, we will encode it according to each $\mathbf{C}_i$ to give a codeword, say $\langle c_{i1}, \ldots, c_{iN}\rangle \in \mathrm{GF}(q_0)^N$. The associated codeword of C^* will then be $\langle d_1, \ldots, d_N \rangle \in \mathrm{GF}(q_0^t)^N$ where $d_j = \langle c_{1j}, c_{2j}, \ldots, c_{tj} \rangle$ is interpreted as an element of $\mathrm{GF}(q_0^t)$.

We now pick parameters δ_i's appropriately in the above scheme and obtain the following theorem.

THEOREM 8. *For every $\varepsilon > 0$, every integer $t \geq 1$ and each $b > t$, there exists a code family with the following properties:*

(i) It has rate $\Omega(t^{-3}\varepsilon^{2+2/t})$ and is defined over an alphabet of size $O(1/\varepsilon^b)$.

(ii) A code of blocklength N in the family can be constructed in $N^{O(1/\varepsilon^{1+1/t})}$ time deterministically, and a construction that has the list decodability property (iii) below with high probability can be obtained in probabilistic $O(\varepsilon^{-(2+2/t)} \log^2 N)$ time.

(iii) A code of blocklength N belonging to the family can be list decoded from up to a fraction $(1-\varepsilon)$ of errors in $N^{O(1/\varepsilon)}$ time using lists of size $O(t^2/\varepsilon^{1+1/t})$.

In the proof of the theorem, we will need the following lemma from [9] concerning the existence of good "pseudolinear" codes to correct a $(1-\varepsilon)$ fraction of errors. The proof uses an application of the probabilistic method and its derandomization using the method of conditional expectations.

LEMMA 9. *Let $a > 1$ be an arbitrary constant. Then there exist constants $b_a, c_a > 1$ such that for every $\varepsilon > 0$ the following holds. For all prime powers $q = \Omega(1/\varepsilon^a)$, there exist $L = b_a/\varepsilon$ and a family of (L, q)-pseudolinear codes of rate at least ε/c_a, such that a code of blocklength n in the family has the following properties:*

(a) it is $((1-\varepsilon)n, L)$-list decodable.

(b) it is constructible in deterministic time $q^{O(n)} = 2^{O(n \log q)}$, or with high probability in randomized $O(n^2 \log q)$ time, and

(c) it can can be represented in $O(n^2\varepsilon^{-1} \log q)$ space, and encoded using $O(n^2\varepsilon^{-2})$ field operations over $\mathbb{F}_q$.

Proof of Theorem 8: Let $\varepsilon > 0$ be given. Pick $q_0 = O(1/\varepsilon^c)$ to be a power of 2 where $c = b/t > 1$. Let us pick the δ_i's in geometric progression with $\delta_0 = \varepsilon/2$, $\delta_t = 1$, and $\delta_i/\delta_{i-1} = \Delta$ for $1 \leq i \leq t$. Note that this implies $\Delta = (2/\varepsilon)^{1/t}$.

For every large enough integer m, we now apply the construction C^* discussed in Section 3.2 with the parameters $(q_0, m, \varepsilon, t, \Delta)$.

The code C^* is then clearly defined over an alphabet of size $q_0^t = O((1/\varepsilon)^{ct}) = O(1/\varepsilon^b)$ Recall that C^* is the juxtaposition of t codes $\mathbf{C}_i$, $1 \leq i \leq t$, each of which is obtained by the concatenation of a rate R_i Reed-Solomon code C_i^{RS} with a rate r_i inner code C_i^{in}, where $R_i = \Theta(\frac{\varepsilon^2}{t^2\delta_i\Delta})$ and $r_i = \Theta(\delta_{i-1})$. Therefore the rate of each $\mathbf{C}_i$ equals

$$R_i r_i = \Omega\Big(\frac{\varepsilon^2}{t^2\delta_i\Delta} \cdot \delta_{i-1}\Big) = \Omega\Big(\frac{\varepsilon^2}{t^2\Delta^2}\Big) \ . \tag{4}$$

Let K, N be the common dimension and blocklength respectively of the $\mathbf{C}_i$'s. The rate of the juxtaposed code C^* is $1/t$ times the rate of each $\mathbf{C}_i$ because of the juxtaposition operator and hence

$$R(C^*) = \Omega\Big(\frac{\varepsilon^2}{t^3\Delta^2}\Big) = \Omega\big(t^{-3}\varepsilon^{2+2/t}\big) \ ,$$

as claimed in Part (i) of the theorem.

The dominant component in the construction of C^* is the construction of the inner codes C_i^{in} used in the concatenated codes $\mathbf{C}_i$. By Lemma 9, we have that each C_i^{in} can be constructed in $q_0^{O(m\Delta/\varepsilon)}$ time deterministically, and $O(m^2\Delta^2\varepsilon^{-2}\log(1/\varepsilon))$ time probabilistically. Since the overall blocklength $N = n_0 n_1 = q_0^m n_1$, we have $m \leq \log N/\log q_0$. Therefore the constructions times are $N^{O(\Delta/\varepsilon)} = N^{O(1/\varepsilon^{1+1/t})}$ for a deterministic construction, and $O(\Delta^2\varepsilon^{-2}\log^2 N) = O(\varepsilon^{-(2+2/t)}\log^2 N)$ for a probabilistic construction (that works with high probability). This proves Property (ii) claimed in the theorem.

It remains to prove the list decodability property of C^*. Specifically, we wish to prove that given a received word $\mathbf{r} \in \mathrm{GF}(q_0^t)^N$, we can output a list of all codewords of C^* that differ from $\mathbf{r}$ in at most $(1-\varepsilon)N$ positions, in $N^{O(1/\varepsilon)}$ time. Indeed let $\mathbf{c} = C^*(\mathbf{x})$ be a codeword of C^* that differs from $\mathbf{r}$ in at most a fraction $(1-\varepsilon)$ of places. Note that both $\mathbf{r}$ and $\mathbf{c}$ can be broken up into n_0 blocks of n_1 symbols each, corresponding to the n_0 inner encodings at the n_0 positions of the outer Reed-Solomon codes. (Here we are using the fact that all the Reed-Solomon codes C_i^{RS} and the inner codes C_i^{in} have the same blocklength, namely n_0 and n_1, respectively.)

Now comes the crucial part. Since the overall agreement between $\mathbf{c}$ and $\mathbf{r}$ is at least a fraction ε of symbols, a standard averaging argument implies that there exists a set B consisting of at least $\varepsilon n_0/2$ inner blocks, such that $\mathbf{c}$ and $\mathbf{r}$ agree on more than a fraction $\varepsilon/2 = \delta_0$ of symbols within each block in B. Now imagine partitioning the blocks in B into t parts P_i, $1 \leq i \leq t$, in the following way. The part P_i consists of all blocks in B for which the fractional agreement between the portions of $\mathbf{c}$ and $\mathbf{r}$ corresponding to that block lies in the interval $(\delta_{i-1}, \delta_i]$. One of these parts must have at least $|B|/t$ blocks. Let this part be P_{i^*}. Hence we conclude that there exists *some* i^*, $1 \leq i^* \leq t$, such that for at least a fraction $\frac{\varepsilon}{2t\delta_{i^*}}$ of the n_0 blocks, $\mathbf{c}$ and $\mathbf{r}$ agree on at least a fraction δ_{i^*-1} of positions within that block.

Now consider decoding the n_0 inner codes $C_{i^*}^{\mathrm{in}}$ corresponding to this i^* up to a radius of $(1-\delta_{i^*-1})$ errors (here we focus attention on and use only the i^*'th symbol from each of the N "juxtaposed" symbols from the received word $\mathbf{r}$). This can be accomplished by brute-force in a total of $n_0 q_0^{m\delta_{i^*}/\delta_0} = q_0^{O(m/\varepsilon)} = n_0^{O(1/\varepsilon)}$ time. By the $((1-\delta_{i^*})n_1, O(1/\delta_{i^*-1}))$-list decodability property of $C_{i^*}^{\mathrm{in}}$, this decoding only outputs a list $L_j^{(i^*)}$ of $O(1/\delta_{i^*-1})$ codewords (or in other words Reed-Solomon symbols for the code $C_{i^*}^{\mathrm{RS}}$) for each of the blocks j, $1 \leq j \leq n_0$. By our choice of i^*, at least $\frac{\varepsilon n_0}{2t\delta_{i^*}}$ of these lists have the "correct" symbol of $C_{i^*}^{\mathrm{RS}}(\mathbf{x})$.

To finish the decoding, it suffices to be able to list decode $C_{i^*}^{\mathrm{RS}}$ with these lists $L_j^{(i^*)}$, $1 \leq j \leq n_0$, as input, and find *all* messages $\mathbf{x}$ such that $L_j^{(i^*)}$ contains the j'th symbol of $C_{i^*}^{\mathrm{RS}}(\mathbf{x})$ for at least $\frac{\varepsilon n_0}{2t\delta_{i^*}}$ values of j. We can now apply the list decoding algorithm for Reed-Solomon codes from [21] to accomplish this decoding task. (Since the fraction of correct lists $\alpha = \varepsilon/(2t\delta_{i^*})$ is $\Omega\big((R_{i^*}/\delta_{i^*-1})^{1/2}\big)$, the decoding algorithm of [21] will succeed.) The running time will be near-quadratic if we use the fast implementations of this algorithm from [15, 14]. Moreover, the result of [10] also implies that the number of codewords output by the decoding algorithm will be $O(\sqrt{n\ell/k})$, which is $O(t\Delta/\varepsilon)$ for our choice of parameters.

Of course, the algorithm cannot know the value of i^* in the above description, but running the above decoding procedure for each $\mathbf{C}_i$, $1 \leq i \leq t$, will output a list of size at most $O(t^2\Delta/\varepsilon) = O(t^2\varepsilon^{-(1+1/t)})$ that includes all codewords that differ from the received word $\mathbf{r}$ in at most a fraction $(1-\varepsilon)$ of the positions. The decoding time is dominated by the time to decode the inner codes, which, as discussed earlier, takes $n_0^{O(1/\varepsilon)} = N^{O(1/\varepsilon)}$ time. This completes the proof of Property (iii) of the theorem as well. □ *(Theorem 8)*

Comparison with Algebraic-geometric codes: Note that for $t \leq 3$, the result of Theorem 8 is incomparable to AG-codes, since it gets a better alphabet size than AG-codes (which work over alphabet size of $O(1/\varepsilon^4)$), but the rate is worse than ε^2. Thus the above codes give some new, interesting trade-offs for codes that can be list decoded in polynomial time from a fraction $(1-\varepsilon)$ of errors.

By picking a fine "bucketing" with $\Delta = 2$ and $t = \lceil \lg(2/\varepsilon) \rceil$ in the above theorem, we can achieve a rate very close to ε^2 though the alphabet size becomes quasi-polynomial in $1/\varepsilon$. Specifically, we can construct codes of rate $\Omega(\varepsilon^2 \log^{-3}(1/\varepsilon))$ over an alphabet of size $2^{O(\log^2(1/\varepsilon))}$ and which can be list decoded in polynomial time from a fraction $(1-\varepsilon)$ of errors using lists of size $O(\varepsilon^{-1}\log^2(1/\varepsilon))$.

4. GOOD ERASURE LIST-DECODABLE CODES

We now proceed to constructions of erasure list-decodable codes over alphabets which are slightly larger than binary. There are two main tools used in our construction. The first one is the use of appropriate pseudolinear codes as inner codes instead of linear codes as in the recent erasure codes construction from [8]. Pseudolinear codes were introduced in [9] for the case of errors and we need an analogue of their results for the case of erasures — the exact codes we require are those guaranteed by Lemma 10 below. The provably better bound on list size for pseudolinear codes compared to linear codes (see Theorem 11 below) translates into some quantitative advantage for the erasure list-decodability of the concatenated code. The second tool is the use of symbol juxtaposition, exactly as in Section 3.2, to combine together several binary codes into a single code over a larger alphabet.

The following lemma guarantees the existence of good "pseudolinear" binary erasure codes. This is the analog of Lemma 9 for the case of erasures. The proof uses a straightforward application of the probabilistic method. We omit the details here, but the interested reader can find them in [7, Chap. 10] or in the full version of this paper.

LEMMA 10. *For every σ, $0 < \sigma < 1/2$, there exist $L = O(\log(1/\sigma))$ and a family of $(L,2)$-pseudolinear codes with the following properties:*

(i) It has rate $\Theta(\sigma)$.

(ii) A code of blocklength n in the family is $((1-\sigma)n, L)$-erasure list-decodable.

(iii) A code of blocklength n in the family can be constructed in $O(n^2\sigma\log(1/\sigma))$ time probabilistically and in $2^{O(\sigma\log(1/\sigma)n)}$ time deterministically.

4.1 Separation between linear and non-linear codes

The following shows that non-linearity gives strictly more power for erasure list-decodable codes. Such a provable asymptotic separation between linear and non-linear codes is extremely rare in coding theory. The result is obtained by a combination of the upper bound on list size for pseudolinear codes from Lemma 10 and the lower bound of $\Omega(1/\varepsilon)$ from [8] on the list size for list decoding linear codes of positive rate from a fraction $(1-\varepsilon)$ of erasures.

THEOREM 11. *Let $\varepsilon > 0$ be sufficiently small and suppose there exists a binary* linear *code family of positive rate that can be list decoded from up to a fraction $(1-\varepsilon)$ of erasures using lists of size L. Then $L = \Omega(1/\varepsilon)$. On the other hand, there exists a binary code family of positive rate (in fact, rate $\Omega(\varepsilon)$) which can be list decoded using lists of size $O(\log(1/\varepsilon))$ from a fraction $(1-\varepsilon)$ of erasures.*

4.2 Main theorem on erasure list-decodability

The following gives our main result on list decoding from erasures. We omit the proof from this extended abstract and defer it to the full version of the paper. We just mention that the proof follows exactly along the lines of Theorem 8; that is, the final code will be the juxtaposition of several concatenated codes with outer codes being Reed-Solomon codes varying rates. For the inner codes, instead of the pseudolinear codes from Lemma 9, we will use the erasure list-decodable pseudolinear codes guaranteed by Lemma 10.

THEOREM 12. *For every $\varepsilon > 0$ and every integer $t \geq 1$, there exists a code family with the following properties:*

(i) It has rate $\Omega(\varepsilon^{1+1/t}/(t^2\log(1/\varepsilon)))$ and is defined over an alphabet of size 2^t.

(ii) A code of blocklength N can be constructed in $O(\log^2 N \log(1/\varepsilon)/\varepsilon^{2+1/t})$ time probabilistically and in $N^{O(\varepsilon^{-1}\log(1/\varepsilon))}$ time deterministically.

(iii) A code of blocklength N in the family can be list decoded from up to a fraction $(1-\varepsilon)$ of erasures using lists of size $O(t\log(1/\varepsilon))$ in $N^{O(1/\varepsilon)}$ time.

The above theorem with the choice $t = 1$ gives binary codes with the same rate as the main result of [8], except that the construction is no longer linear, and the list size is much better (namely, $O(\log(1/\varepsilon))$ instead of the $O(1/\varepsilon)$ obtained in [8]). Applying Theorem 12 with $t = \lceil \lg(1/\varepsilon) \rceil$, we get the following corollary:

COROLLARY 13. *For every $\varepsilon > 0$, there exists a polynomial time constructible code family of of rate $\Omega(\varepsilon\log^{-3}(1/\varepsilon))$ over an alphabet of size $O(1/\varepsilon)$ such that each code in the family can be list decoded from up to a fraction $(1-\varepsilon)$ of erasures in polynomial time using lists of size $O(\log^2(1/\varepsilon))$.*

Note that the rate is off only by a $\log^{O(1)}(1/\varepsilon)$ factor from the *optimal* $\Omega(\varepsilon)$ rate which is achieved by random codes. However, while the random construction works even over the binary case, we need larger and larger alphabet sizes in order to approach the optimal $\Omega(\varepsilon)$ rate.

AG-codes which achieve the best known trade-offs between rate and minimum distance provide constructions of highly erasure list decodable codes of the optimal rate $\Omega(\varepsilon)$, but they require an alphabet size of $O(1/\varepsilon^2)$. The constructions implied by Theorem 12 are incomparable to AG-codes since they achieve a better alphabet size at the expense of a moderate worsening of the rate. Thus, the results of this section give polynomial time constructions of codes that achieve some new and non-trivial trade-offs between rate and erasure list-decodability.

An interesting question that remains open is to construct codes over the binary or ternary alphabet with rate better than $\Omega(\varepsilon^2)$ for list decoding from a fraction $(1-\varepsilon)$ of erasures.

4.3 Improving the decoding time in Theorem 12

One of the drawbacks of the result of Theorem 12 is that the decoding time is quite high (namely, $N^{O(1/\varepsilon)}$). It turns out that juxtaposing the pseudolinear codes used at the inner level further with a suitable *linear* code, we can improve the decoding time to near-quadratic in N at the expense of a squaring the alphabet size. Specifically, we can get codes of rate $\Omega(\varepsilon^{1+1/t}/(t^2 \log(1/\varepsilon)))$ over an alphabet of size 2^{2t}. We omit the details.

5. REFERENCES

[1] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ronny Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38:509–516, 1992.

[2] Noga Alon, Jeff Edmonds and Michael Luby. Linear time erasure codes with nearly optimal recovery. *Proceedings of FOCS'95*.

[3] Alexander Barg and Gillés Zémor. Error exponents of expander codes. *IEEE Transactions on Information Theory*, to appear, 2001.

[4] E. L. Blokh and V. V. Zyablov. Linear concatenated codes. *Nauka, Moscow*, 1982 (in Russian).

[5] 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.

[6] G. David Forney. Generalized Minimum Distance decoding. *IEEE Transactions on Information Theory*, 12:125–131, 1966.

[7] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes*. Ph.D thesis, Massachusetts Institute of Technology, August 2001.

[8] Venkatesan Guruswami. List decoding from erasures: Bounds and code constructions. *Proceedings of the 21st Foundations of Software Technology and Theoretical Computer Science*, Bangalore, India, pages 195-206, December 2001.

[9] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science*, Las Vegas, NV, pages 658-667, October 2001.

[10] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45:1757–1767, 1999.

[11] Piotr Indyk. List decoding in linear-time. *Manuscript*, 2002.

[12] G. L. Katsman, Michael A. Tsfasman, and Serge G. Vlădut. Modular curves and codes with a polynomial construction. *IEEE Transactions on Information Theory*, 30:353–355, 1984.

[13] Alex Lubotzky, R. Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[14] Rasmus R. Nielsen and Tom Hϕholdt. Decoding Reed-Solomon codes beyond half the minimum distance. *Coding Theory, Cryptography and Related areas, (eds. Buchmann, Hoeholdt, Stichtenoth and H. tapia-Recillas)*, pages 221–236, 1999.

[15] Ronny Roth and Gitit Ruckenstein. Efficient decoding of Reed-Solomon codes beyond half the minimum distance. *IEEE Transactions on Information Theory*, 46(1):246–257, January 2000.

[16] Ba-Zhong Shen. A Justesen construction of binary concatenated codes that asymptotically meet the Zyablov bound for low rate. *IEEE Transactions on Information Theory*, 39:239–242, 1993.

[17] M. Amin Shokrollahi and Hal Wasserman. List decoding of algebraic-geometric codes. *IEEE Transactions on Information Theory*, 45(2):432–437, 1999.

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

[19] Daniel Spielman. *Computationally Efficient Error-Correcting Codes and Holographic Proofs*. Ph.D thesis, Massachusetts Institute of Technology, June 1995.

[20] Daniel Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1732, 1996.

[21] Madhu Sudan. Decoding of Reed-Solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

[22] Gillés Zémor. On expander codes. *IEEE Transactions on Information Theory*, 47(2):835–837, 2001.

[23] Victor V. Zyablov. An estimate of the complexity of constructing binary linear cascaded codes. *Problemy Peridachi Informatsii*, 15(2):58-70, 1971.