Error Correcting Codes for Uncompressed Messages

Ofer Grossman
MIT, Cambridge, MA, USA
ogrossma@mit.edu

Justin Holmgren
NTT Research, Palo Alto, CA, USA
justin.holmgren@ntt-research.com

Abstract

Most types of messages we transmit (e.g., video, audio, images, text) are not fully compressed, since they do not have known efficient and information theoretically optimal compression algorithms. When transmitting such messages, standard error correcting codes fail to take advantage of the fact that messages are not fully compressed.

We show that in this setting, it is sub-optimal to use standard error correction. We consider a model where there is a set of "valid messages" which the sender may send that may not be efficiently compressible, but where it is possible for the receiver to recognize valid messages. In this model, we construct a (probabilistic) encoding procedure that achieves better tradeoffs between data rates and error-resilience (compared to just applying a standard error correcting code).

Additionally, our techniques yield improved efficiently decodable (probabilistic) codes for fully compressed messages (the standard setting where the set of valid messages is all binary strings) in the high-rate regime.

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Coding Theory, List Decoding

Digital Object Identifier 10.4230/LIPIcs.ITCS.2021.43

Related Version The full version of this paper is available at [11], `https://eccc.weizmann.ac.il/report/2020/038/`.

Supplementary Material The code used to generate the figures and tables in this paper can be found at `https://github.com/jnhnum1/contextual_codes`.

Funding Work done in part while generously hosted by the Simons Institute.
Ofer Grossman: Supported by a Fannie and John Hertz Foundation Fellowship and a NSF GRFP fellowship.
Justin Holmgren: Work done in part while at Princeton University, supported by the Simons Collaboration on Algorithms and Geometry and National Science Foundation grant No. CCF-1714779.

Acknowledgements We thank Elad Haramaty and Madhu Sudan for many very fruitful discussions in the early stages of this work. We also thank Venkat Guruswami and Lijie Chen for helpful discussions, and Shafi Goldwasser for her encouragement and her feedback on an earlier version of this work. We also thank anonymous reviewers for many useful comments on earlier versions of this work.

1 Introduction

If Alice wishes to send a message m to Bob, she might first compress it as well as she can. In this work, we focus on *lossless* compression, meaning that Bob must recover m *exactly*. There are many types of data (e.g. images, audio, video, text) that we do not know how to compress and decompress efficiently and information-theoretically optimally. For such messages, this compression step will result in a longer-than-optimal message. For example, the best efficient compression scheme may result in a b-bit long compressed message, whereas an information theoretically optimal compression scheme might be able to obtain $0.5b$ bits.

After compressing, Alice can apply an error correcting code to the compressed message, ensuring that Bob can recover the message in the presence of corruptions. Suppose Alice wishes to have her message be resilient against up to 5% worst-case errors. Then, the best known construction of a code with efficient unique decoding and public randomness will result in a total of approximately $3.64b$ bits sent to Bob (using bounds implicit in [22, 17]).

We show that in this setting it is sub-optimal to treat compression and error correction as two orthogonal concerns. We instead address both compression and error correction simultaneously, constructing an error correcting code that exploits the fact that messages are not fully compressed. This allows Alice to send only fewer bits to Bob with the same error resilience. For example, with the above parameters Alice can send $2.24b$ bits.

1.1 Contextually Unique Decoding

We define a new notion, *contextually unique decoding*, that formalizes the idea of encoding a message that is not fully compressed. Roughly speaking, we let $S \subseteq \{0,1\}^k$ denote a set of "valid messages" that Alice may send. Suppose, for example, that Alice is sending English text of a certain size to Bob. Then we think of S as the set of all "reasonable" texts Alice can send. For example, "meet me at 5pm" (when translated to binary) is in S. However, "wef ojip447oll" is not in S. We assume that Bob has oracle access to S – he has the power to determine whether a message is reasonable or not. Because most strings do not look like reasonable texts, we see that S is pretty small[1]. We now wish to say that whenever Alice encodes an element m of S, Bob will be able to recover m.

Motivated by this, we say that a family of codes $\{C_i : \{0,1\}^k \to \{0,1\}^n\}$ is *contextually uniquely decodable* if there is a decoding algorithm D such that for any sufficiently small set of messages $S \subseteq \{0,1\}^k$, it holds w.h.p. for a random i that for all $m \in S$, the algorithm D (with oracle access to S) can recover m given i and an adversarially corrupted $C_i(m)$ (where the adversary may depend on i).

Alice may have partially compressed the text she wishes to send (we assume she cannot fully compress it, since we don't know any efficient practical information theoretically optimal compression schemes for text). In this case, a message is in S if it looks like reasonable text once it is decompressed.

A code with contextually unique decoding would (assuming public randomness) allow Alice to send a message to Bob, so that he can recover Alice's message even in the presence of corruptions.

Formally, we define:

▶ **Definition 1** (δ-Hamming Adversary). *A δ-*Hamming adversary *is a function $\mathcal{A} : \{0,1\}^n \to \{0,1\}^n$ such that for all $c \in \{0,1\}^n$, the Hamming distance between c and $\mathcal{A}(c)$ is at most δn.*

[1] Notice that if S if of size $2^{k'}$, then information theoretically it would be possible for Alice to compress the message to k' bits, and then apply an error correcting code.

▶ **Definition 2** (Contextually Unique Decoder). *An oracle algorithm D is an $(r, \delta, \epsilon, \tau)$-*contextually unique decoder *for a family of probabilistic codes $\{C_i : \{0,1\}^k \xrightarrow{\$} \{0,1\}^n\}_{i \in \mathcal{I}}$ if for all sets $S \subseteq \{0,1\}^k$ with $|S| \leq 2^{rn}$, it holds with probability at least $1 - \epsilon$ over the choice of $i \leftarrow \mathcal{I}$ that for all messages $m \in S$ and all δ-Hamming adversaries $\mathcal{A}$ (that may depend on i),*

$$\Pr_{c \leftarrow C_i(m)}[D^S(i, \mathcal{A}(c)) \neq m] \leq \tau.$$

We emphasize the order of quantifiers in the definition. We use randomness in two different ways. First, randomness is used to pick a code from the family $\{C_i\}$. This choice of randomness is agreed upon by all parties ahead of time and is publicly known (to the sender, receiver, and the adversary), and must work for all messages. Randomness is then also used by the sender (Alice) when encoding. That is, even after fixing the message m and the choice of C_i, the encoding of a message m using C_i depends on the encoder's randomness (we use the notation $C_i : \{0,1\}^k \xrightarrow{\$} \{0,1\}^n$ to denote that C_i is a function taking an input from $\{0,1\}^k$, along with some randomness, and outputs an element of $\{0,1\}^n$, which may depend on the randomness). The decoder only needs to know the randomness used in picking C_i, and not the randomness used by the encoder in *evaluating* C_i.

1.2 Main Result and Construction Overview

We first overview our construction of contextually unique decodable codes, and then we formally describe our main result (Theorem 3).

1.2.1 Construction Overview

The Main Idea

In the standard model of error correcting codes, we have a code C, which we use to encode a message m as $C(m)$. Then, even when an adversary may corrupt a bounded number of entries of $C(m)$, it is still possible to recover m. This is called unique decoding. List decoding [6, 28] is a generalization of unique decoding where instead of recovering m, the decoding algorithm outputs a short (polynomial sized) list $m_1, m_2, \ldots, m_\ell$ such that the real message m is in the list. This relaxation makes it possible to handle more errors.

The key in our construction is to have Alice send a coded version of the message m with good list decoding properties. Then, the goal will be that when Bob list-decodes Alice's message, only one of the elements in Bob's list will be a "valid message" (that is, only one element of the list will be in S). Then, the hope is that Bob can correct errors up to the list decoding radius, instead of the unique decoding radius.

So for example, Alice might encode the message "call me at 4pm", and after an adversary adds some errors, and Bob decodes, he will have a list of messages. Ideally, the list will look something like "kwjlewf 6oahzm", "aowi2ifmlpzo", "wef ojip447oll", "call me at 4pm", and "5ncbzmap89pqq". From this list, it will be easy for Bob to infer that the message Alice sent was "call me at 4pm" (formally, he will use his oracle access to S to check which of the strings are in S, and we hope that only one will be in the set S of valid messages). Note, however, that if Bob's list contains more than one valid message – for example, if the list of messages is "call me at 7pm", "my phone broke", "call me at 4pm", and "come to my office" – then it will not be possible for Bob to determine which was the intended message (formally, this situation corresponds to Bob's list containing more than one element in S).

The technical focus of our constructions is ensuring that within the set of candidate messages provided by a list-decoding algorithm, with high probability only one message will be valid. Our main theorem (Theorem 3) can indeed be viewed as a transformation from a list-decodable code to a contextually-uniquely decodable code.

Randomizing the message space

Let S be the set of valid messages. To ensure that only one elements of Bob's list is in S, the idea is to randomize the message space. If the messages in Bob's list are more or less random (other than Alice's intended message), it is unlikely that more than one of the messages will be in S (we assume that $|S|$ is small relative to the entire message space $\{0,1\}^k$). So ideally, what we would want to do is pick a random permutation π of $\{0,1\}^k$ (which is the set of all possible messages, including those not in S) using public randomness, and then use an error correcting code C with good list decoding parameters on $\pi(m)$. Then, the set of messages that the adversary can cause to be in Bob's list will be a random small subset of $\{0,1\}^k$, which, because S is small, will likely not intersect S.

There are some issues with the approach described above. One issue is that picking a random permutation of $\{0,1\}^k$ requires a number of bits exponential in k, but we want Alice and Bob to be efficient. This issue can be solved by using pairwise independence. Roughly speaking, one can see why pairwise independence is enough as follows. The choice of π is bad if there are two messages m_1 and m_2 in S such that $C(\pi(m_1))$ is close to $C(\pi(m_2))$. This causes the adversary to be able to corrupt few entries of an encoding of m_1 and cause it to be close to an encoding of m_2. One can see that the probability $C(\pi(m_1))$ is close to $C(\pi(m_2))$ is the same for a random π and a π chosen from a pairwise independent family, since it depends on only two evaluations of π.

Another issue with the construction as described above is that the probability that there exist $m_1, m_2 \in S$ with $C(\pi(m_1))$ close to $C(\pi(m_2))$ is not that low (it is 2^{-cn}, for some c. Ideally, we would like it to be $2^{-\omega(n)}$, so we can apply a union bound over all of S and not worry about the value of c). We alter the construction by instead of picking a single π, picking a collection $\pi_1, \pi_2, \ldots, \pi_N$ which will be agreed on using public randomness. Then, the encoder (Alice) will pick a random $j \in [N]$, and use $C(\pi_j(m))$ as the message sent to Bob. To decode, Bob will decode C to get a list L, and then for each $j \in [n]$ and for each $x \in L$ he will check if $\pi_j^{-1}(x) \in S$, and with high probability only one such pair (x, j) will satisfy $\pi_j^{-1}(x) \in S$. Then Bob will know that the message m that Alice sent is $\pi_j^{-1}(x)$.

We now outline how we show that Bob's list with high probability indeed contains only one element in S. Consider the probability that for a certain m in S, we have $C(\pi(m))$ close to some $C(\pi(m'))$. Call this probability p. Then, the probability that for most j, we have $C(\pi_j(m))$ close to some $C(\pi_{j'}(m'))$ will be approximately $p^{\Omega(N)}$ (by a Chernoff bound), which is much smaller than p. This allows us to apply a union bound over all messages in S without losing anything significant.

Amplifying the success probability (Section 4)

The construction described above works, but it has a downside that there is inverse polynomial probability of error. That is, with probability inversely polynomial in the message lengths, Bob may be unable to recover the message Alice sent, since with probability approximately $\frac{1}{N}$ Alice may pick a bad choice of i.

Ideally, we would want to succeed with all but negligible probability. One approach is to set N to be superpolynomial. The problem with this is that now Bob will not be able to efficiently decode, since his decoding algorithm requires trying every one of the N

permutations. To fix this, instead of Alice sending Bob $C(\pi_i(m))$, she will send him $C(\pi_i(m), i)$ (we apply C to the string which is the concatenation of $\pi_i(m)$ and i). Now, when Bob list decodes C, he will get a list of the form $(x_1, i_1), (x_2, i_2), \dots (x_\ell, i_\ell)$. Now, he can check if $\pi_{i_1}^{-1}(x_1) \in S$, or if $\pi_{i_2}^{-1}(x_2) \in S$, and so on. Crucially, we see that for each element (x_j, i_j) in the list, Bob needs to try only a single permutation (namely i_j), instead of all permutations. This allows Bob to remain efficient even when there are superpolynomially many permutations for Alice to pick from.

This leads to a new issue, since one needs to agree on a superpolynomially sized family of permutations sampled from a family of pairwise independent permutations. Also, we want this family to be efficiently sampleable, and for each element to have a succinct description. It turns out that this can be solved using k-wise independence (and k-wise ϵ-dependence). This part is more technical, and we refer the reader to the body of the paper for details.

1.2.2 The main theorem

Here we formally describe the main theorem. We say that an n-bit code is combinatorially (ρ, λ)-list decodable if for any $y \in \{0,1\}^n$, there are at most $\approx 2^{\lambda n}$ codewords within relative Hamming distance ρ of y. We are also interested in the asymptotic computational efficiency of encoding and decoding procedures, so we consider ensembles of codes $\{C_n : \{0,1\}^{k_n} \to \{0,1\}^n\}_{n \in \mathbb{Z}^+}$. We will restrict our attention to ensembles where $r = \lim \frac{k_n}{n}$ exists, and we call r the rate of the ensemble. We say that $\{C_n\}$ is efficiently ρ-list-decodable if there is a polynomial-time algorithm that on input $y \in \{0,1\}^n$, outputs all codewords of C_n that are within relative Hamming distance ρ of y.

▶ **Theorem 3** (Simplified Main Theorem). *Suppose that $\{C'_n : \{0,1\}^{k'_n} \to \{0,1\}^n\}_{n \in \mathbb{Z}^+}$ is a rate-r' ensemble of (deterministic) codes that is efficiently ρ-list-decodable. Suppose also that $\{C'_n\}$ is* combinatorially *$(2\rho, \lambda)$-list decodable.*

Then for some negligible function $\epsilon(n)$, there is a rate-r' ensemble of probabilistic codes $\{C_n\}$ such that C_n has a polynomial-time $\big(r' - \lambda - o(1), \rho, \epsilon, \epsilon\big)$-contextually unique decoder.

So, suppose we wish to construct contextually unique codes where the message can be recovered when there are up to 0.1 fraction of corruptions. So we have $\rho = 0.1$. We now wish to find an r' and λ which maximize $r' - \lambda$ such that there are deterministic codes which are of rate r', and are combinatorially $(2\rho, \lambda)$-list decodable (we also have make sure the codes are *efficiently* ρ-list decodable).

Once fixing ρ, the tradeoff here is between r' and λ. The best contextually unique codes will have high rates $r' - \lambda - o(1)$, so we want λ to be small, and r' to be large. However, the codes $\{C'_n\}$ must be combinatorially $(2\rho, \lambda)$-list decodable. So, as we increase r', the lowest possible value of λ decreases.

We give some examples of parameter settings to Theorem 3 in Table 1.

1.3 Improvements for standard randomized setting

An important special case of contextually unique decoding is obtained by fixing $S = \{0,1\}^{k_n}$, viewed as a subset of $\{0,1\}^{k'_n}$ for $k'_n > k_n$ by zero-padding. In this case, a contextually uniquely decodable code is quite similar to a standard error-correcting code – the main difference is in the use of randomness both in generating the code and encoding messages. Perhaps surprisingly, we obtain better parameters in the high-rate regime than any other known efficiently decodable code (including probabilistic constructions with public randomness).

Table 1 This table shows some example values of what rates can be achieved with our main theorem (Theorem 3) together with the best of the Blokh-Zyablov (Fact 6) and Thommesen-Rudra bounds. If $|S| = 2^{s \cdot k}$, and $S \subseteq \{0,1\}^k$, we say that the *sparsity* of S is s. So, for example, if S is of size $2^{.5k}$, and there are 3% fraction of errors, we see that we can achieve rate .574 (and so Alice's message size would be $k/.574$, or approximately $1.74k$). The best Alice would be able to do without contextually unique decoding would be rate .396, which corresponds to over $2.52k$ bits sent (this can be seen by looking at the sparsity 1 row, which corresponds to using standard unique decoding, since in this case S is the whole message space $\{0,1\}^k$).

Errors Sparsity	0.01	0.02	0.03	0.05	0.1	0.2
1	0.661	0.504	0.396	0.275	0.142	0.028
0.9	0.778	0.591	0.451	0.277	0.142	0.030
0.75	0.778	0.661	0.574	0.332	0.142	0.034
0.5	0.778	0.661	0.574	0.446	0.142	0.038
0.25	0.778	0.661	0.574	0.446	0.142	0.040
0.1	0.778	0.661	0.574	0.446	0.178	0.041
0.05	0.778	0.661	0.574	0.446	0.202	0.041
0.01	0.778	0.661	0.574	0.446	0.235	0.041

Discussion

If there is a (deterministic) code that can be *efficiently* list decoded up to ρ errors, and *combinatorially* uniquely decoded up to ρ' errors, then it is possible to efficiently uniquely decode up to $\min(\rho, \rho')$ errors. This can be done by simply list decoding, and then going over every element in the list to determine which of the elements, when encoded, is closest to the received message. So, in short, it is not hard to see that good efficient list decoding and good combinatorial unique decoding implies good efficient unique decoding (this idea is, roughly speaking, what gives the green line (TR bound) in Figure 1).

Our corollary can be viewed as a strengthening of this result. We show that rather than requiring good efficient list decoding and good combinatorial unique decoding, we can use codes with good efficient list decoding and good combinatorial *list* decoding. The idea, roughly speaking, is to use our efficient list decoding algorithm to obtain a list of candidate messages, and we use our main theorem on contextually unique decoding to ensure that only one of these messages will be a "valid" message. Then, we can go through each candidate in the list, and pick the one which is a valid message.

▶ **Corollary 4** (Simplified Main Corollary). *Suppose that* $\{\tilde{C}^{(n)} : \{0,1\}^{k'_n} \to \{0,1\}^n\}_{n \in \mathbb{Z}^+}$ *is a rate-*r' *ensemble of (deterministic) codes that is efficiently* ρ*-list-decodable. Suppose also that* $\{\tilde{C}^{(n)}\}$ *is* combinatorially $(2\rho, \lambda)$*-list decodable.*

Then there exists an ensemble $\{\mathcal{C}^{(n)}\}_{n \in \mathbb{Z}^+}$*, where* $\mathcal{C}^{(n)} = \{C_i : \{0,1\}^{k_n} \xrightarrow{\$} \{0,1\}^n\}_{i \in \mathcal{I}^{(n)}}$ *is a family of probabilistic codes, such that:*

1. $\lim \frac{k_n}{n} = r' - \lambda$*, and*
2. *There are* $\mathrm{poly}(n)$*-time algorithms to:*
   - *Sample from* $\mathcal{I}^{(n)}$ *given* 1^n*.*
   - *Probabilistically encode* $C_i(m)$ *given* i *and* m*,*
   - *Decode* ρ*-corrupted codewords of* C_i*. That is, there is a deterministic* $\mathrm{poly}(n)$*-time algorithm* D *and a negligible function* $\epsilon(n)$ *such that with probability at least* $1 - \epsilon(n)$ *over the choice of* $i \leftarrow \mathcal{I}^{(n)}$*, it holds for all messages* $m \in \{0,1\}^{k_n}$ *and all* ρ*-Hamming adversaries* $\mathcal{A}$*, that*

$$\Pr_{c \leftarrow C_i(m)}[D(i, \mathcal{A}(c)) \neq m] \leq \epsilon(n).$$

▶ Remark 5. An interesting weakening of the conclusion of Corollary 4 is that for sufficiently large n, there exists $i \in \mathcal{I}^{(n)}$ such that for all messages $m \in \{0,1\}^{k_n}$, there exists randomness s such that for all $c' \approx_\delta C_i(m; s)$, it holds that $D(i, c') = m$. In other words, $D(i, \cdot)$ is a polynomial-size error-correcting circuit for an (inefficiently computable and non-explicit) *deterministic* code.

We compare the conclusion of Corollary 4 to what is known for standard (deterministic) binary codes.

To our knowledge, the best known rate vs. error tolerance tradeoff for efficiently decodable and deterministic binary codes is given by the Blokh-Zyablov (BZ) bound [2], and is attained by multi-level concatenated codes (see Fact 6). With probabilistically constructed codes, it is possible to do better for sufficiently low rates. It is known that for rates below about 0.02, the concatenation of a folded Reed-Solomon code with random linear codes simultaneously achieves high distance (matching the GV bound) [26] and efficient list-decodability for a larger number of errors [22]. This implies an efficient unique decoding procedure (by list decoding and then taking the candidate that is closest to the received word). While not made explicit in previous work, the same ideas achieve performance that is intermediate between the BZ and GV bounds for rates up to about 0.3. We will refer to the resulting rate-distance tradeoff as the Thommesen-Rudra (TR) bound. In [17], the authors implicitly show that that one can achieve *near linear* time decoders up to the TR bound.

In the high rate regime there were no codes, even probabilistic constructions, that were efficiently decodable beyond the BZ bound.

▶ **Fact 6** (Blokh-Zyablov bound [2, 14]). *For any $\rho \in (0, \frac{1}{2})$ and any*

$$0 < R < R_{\mathsf{BZ}}(\rho) \stackrel{\mathsf{def}}{=} 1 - H(\rho) - \rho \cdot \int_0^{1-H(\rho)} \frac{dx}{H^{-1}(1-x)},$$

there exists a rate-R ensemble of codes $\{C_n\}_{n \in \mathbb{Z}^+}$ that is efficiently ρ-list decodable and efficiently uniquely-decodable against up to a $\frac{\rho}{2}$ fraction of errors.

Note that efficient ρ-list decodability generically implies combinatorial $\big(\rho', H(\rho') - H(\rho) + o(1)\big)$-list decodability for any $\frac{1}{2} \geq \rho' \geq \rho$. By combining this with Corollary 4, we obtain bounds that improve over the BZ (and TR) bound for rates above roughly 0.3. This is illustrated in Figure 1.

1.4 Related Work

Our work is an application of list decoding, a notion that was introduced by Elias in the 50's [6]. The notion was then (implicitly) revisited with a focus on algorithmic efficiency by Goldreich and Levin [10], who showed how to efficiently list-decode the Hadamard code, and later by Sudan [24], who showed the same for Reed-Solomon codes. List decoding has proven to be a useful notion in computational complexity theory, and has recently been the focus of extensive research (see e.g. the surveys of Sudan [25] and Guruswami [13]).

Several works have studied variants of the error correction problem in which it is possible to obtain improved results on worst-case unique decoding. Guruswami [12] considered a model in which a sender is able send a small amount of information over a noise-free channel, and showed that this enables unique decoding up to the list-decoding radius. Langberg [20] considered the different notion of "private codes", in which the sender and receiver share some secret randomness, and showed that it is possible to achieve better parameters in this model. Our constructions in contrast use only public randomness, and does not require any noise-free channel.

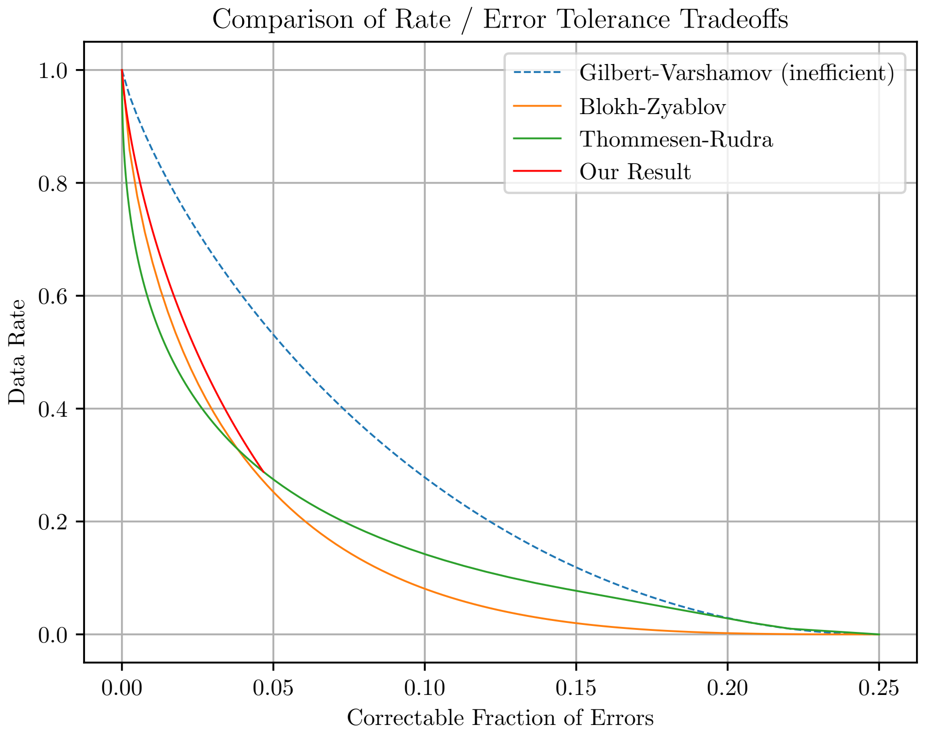

Figure 1 We improve over previous efficiently decodable binary codes (even probabilistic constructions) [2, 22] for rates above about 0.3. Although it appears in this plot as if the TR bound slightly beats the GV bound for very low rates, this is an artifact of our plotting software that disappears upon zooming in.

Perhaps a more relevant line of work to us is one that studies, loosely speaking, whether imperfectly shared context can improve the efficiency of interactive protocols. This question has been articulated and studied in the settings of interactive communication complexity [3, 7, 8], simultaneous message passing [1], and message compression [18, 16], and in the general setting of "goal-oriented communication" [9].

Our work can be viewed through a similar lens. We seek to improve the efficiency of communication, leveraging context (which implies that only a small number of messages "make sense"). Like in prior works, the context is not fully known to both parties. In fact, we go further: the sender may know *nothing* about the context, other than that the message he is sending makes sense. At the same time, the receiver may know very little about the context – only enough to answer a polynomial number of questions on whether a given message makes sense. Moreover, in contrast to prior works, we do not assume error-free communication channels, and we emphasize the importance of efficient algorithms, while prior works have focused primarily on minimizing communication.

A main idea in this paper is to use list decodable codes, and to permute the message space in such a way as to achieve unique decoding instead of just list decoding. Similar ideas have been used for example in [15] and [4]. However, in those works the adversary is not fully general like in this work, but is restricted (either computationally, or by having to corrupt the codeword in an online fashion).

2 Preliminaries

2.1 Codes

A deterministic code of dimension k and block length n over an alphabet Σ is a (multi-)subset $\mathcal{C} \subseteq \Sigma^n$ of size $|\Sigma|^k$, whose elements are called codewords. The rate of such a code is the quantity $\frac{k}{n}$. Throughout this paper, we focus on the case when Σ is a finite field $\mathbb{F}_q$ and when the dimension k is integral. In such cases, we associate the codewords of $\mathcal{C}$ with Σ^k, and we abuse notation by writing $\mathcal{C}$ to refer both to the multiset of codewords and the corresponding mapping from Σ^k to Σ^n. A code $\mathcal{C}$ as above is said to be linear if it is a subspace of Σ^n, and in this case the associated mapping can be taken to be a linear function.

For any alphabet Σ, any n, and any $u, v \in \Sigma^n$, the Hamming distance between u and v, denoted $\Delta(u, v)$, is

$$\Delta(u, v) \overset{\text{def}}{=} \left|\left\{i \in [n] : u_i \neq v_i\right\}\right|.$$

When $\Delta(u, v) \leq \delta n$, we write $u \approx_\delta v$. If S is a set, we write $\Delta(u, S)$ to denote $\min_{v \in S} \Delta(u, v)$. The distance of a code $\mathcal{C}$ is $\min_{c \neq c' \in \mathcal{C}} \Delta(c, c')$.

We also consider probabilistic codes, focusing on codes over binary alphabets.

▶ **Definition 7.** *A* probabilistic binary code *of* block length *n and* dimension *k is a randomized function $\mathcal{C} : \{0,1\}^k \xrightarrow{\$} \{0,1\}^n$.*

When discussing the asymptotic performance of (deterministic or probabilistic) codes, it makes sense to consider ensembles of codes $\{\mathcal{C}_n : \{0,1\}^{k_n} \to \{0,1\}^{\ell_n}\}$ with varying message lengths and block lengths. We will always assume several restrictions on k_n and ℓ_n to rule out pathological examples. Specifically, we will assume that:

- The limit $r = \lim_{n\to\infty} \frac{k_n}{\ell_n}$ exists with $r \in (0,1)$. We call r the rate of the ensemble.
- $\lim_{n\to\infty} \frac{\ell_n}{n} = 1$. This is important so that for a large message of length k, the cost of padding to length k_n is not too large.

Given these two assumptions, it is possible without loss of generality to assume $\ell_n = n$ (we can always take a code from the ensemble with larger ℓ_n, and truncate it; asymptotically, this affects neither its rate nor its error tolerance).

▶ **Definition 8.** *We say that an ensemble of codes $\{C_n : \{0,1\}^{k_n} \to \{0,1\}^n\}_{n\in\mathbb{Z}^+}$ is* combinatorially (ρ, λ)-list decodable *if there is some $L(n) \leq 2^{(\lambda+o(1))\cdot n}$ and $\rho'(n) \geq \rho - o(1)$ such that for any $y \in \{0,1\}^n$, there are at most $L(n)$ values of $m \in \{0,1\}^{k_n}$ for which $C_n(m) \approx_{\rho'(n)} y$. If there is a polynomial-time algorithm that outputs all such m (in which case we can assume $\lambda = 0$), then we say that $\{C_n\}$ is* efficiently ρ-list decodable*.*

We will also say that $\{C_n\}$ is combinatorially ρ-list decodable *if it is combinatorially $(\rho, 0)$-list decodable.*

2.2 Binomial Coefficients

We will use the following approximations of binomial coefficients.

▶ **Fact 9.** *For any $n, k \in \mathbb{Z}^{\geq 0}$, it holds that*

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k.$$

For all constant $0 \leq \delta \leq 1$, as n goes to infinity

$$\binom{n}{\delta n} = \tilde{\Theta}\big(2^{H(\delta)n}\big),$$

where $H(p) = -p \log_2 p - (1-p)\log_2(1-p)$ is the binary entropy function.

We will also use the standard notion of q-ary entropy.

▶ **Definition 10.** *The q-ary entropy function is*

$$H_q(x) = x\log_q(q-1) - x\log_q x - (1-x)\log_q(1-x).$$

We define the inverse function H_q^{-1} to map any $y \in [0,1]$ to the unique value $x \in [0, 1-1/q]$ for which $H_q(x) = y$.

2.3 Covering Numbers for Hamming Balls

For $x \in \{0,1\}^n$, we will denote by $B_r(x)$ the Hamming ball of radius r centered at x, i.e. the set $\{x' \in \{0,1\}^n : \Delta(x,x') \leq r\}$.

▶ **Definition 11.** *Let S be a subset of $\{0,1\}^n$, and let r be a positive real number. An r-covering of S is a subset C of $\{0,1\}^n$ such that $S \subseteq \cup_{x \in C} B_r(x)$. The r-covering number of S, denoted $N_r(S)$, is the minimum cardinality of any r-covering of S.*

A volume argument with Fact 9 shows, for any $0 < \delta_0 < \delta_1 \leq \frac{1}{2}$, that $N_{\delta_0 n}(B_{\delta_1 n}) \geq \tilde{\Omega}\big(2^{(H(\delta_1)-H(\delta_0))\cdot n}\big)$. In fact, a simple application of the probabilistic method (due to Dumer et al.) also shows that $N_{\delta_0 n}(B_{\delta_1 n}) \leq \tilde{O}\big(2^{(H(\delta_1)-H(\delta_0))\cdot n}\big)$. These two statements are combined in the following fact.

▶ **Fact 12** ([5, Eq. 2.4])**.** *For any $0 \leq \delta_0 < \delta_1 \leq \frac{1}{2}$ and any $x \in \{0,1\}^n$, it holds that*

$$N_{\delta_0 n}\big(B_{\delta_1 n}(x)\big) = \tilde{\Theta}\big(2^{(H(\delta_1)-H(\delta_0))\cdot n}\big).$$

2.4 *t*-wise Independence

▶ **Definition 13.** *A family of hash functions $\{h_i : X \to Y\}_{i \in \mathcal{I}}$ is said to be t-wise independent if for all distinct $x_1, \ldots, x_t \in X$, the distribution of $(h_i(x_1), \ldots, h_i(x_t))$ for a uniformly random $i \in \mathcal{I}$ is uniformly random over Y^t.*

▶ **Imported Theorem 14** ([27])**.** *For any $n, m, t \in \mathbb{Z}^+$, there exists a t-wise independent family of hash functions mapping $\{0,1\}^n$ to $\{0,1\}^m$ such that it takes $\mathrm{poly}(n,m,t)$ time to sample or evaluate a hash function.*

▶ **Definition 15.** *A family of permutations $\{\pi_i : X \to X\}_{i \in \mathcal{I}}$ is said to be t-wise ϵ-dependent if for all distinct $x_1, \ldots, x_t \in X$ it holds for uniformly random $i \in \mathcal{I}$ that the distribution of $\big(\pi_i(x_1), \ldots, \pi_i(x_t)\big)$ is ϵ-close in statistical distance to uniformly random over tuples of* distinct *$y_1, \ldots, y_t \in X$.*

▶ **Imported Theorem 16** ([19, Theorem 5.9])**.** *For any $\epsilon > 0$ and any $t \in \mathbb{Z}^+$, there exists a t-wise ϵ-dependent family of permutations on $\{0,1\}^n$ with description length $O\big(nt + \log(\frac{1}{\epsilon})\big)$ such that it takes time $\mathrm{poly}\big(n, t, \log(\frac{1}{\epsilon})\big)$ to sample, evaluate, or invert a permutation.*

3 Contextually Unique Decoding

In this section we present the notion of contextually-unique decoding, and we give some simple constructions of contextually-unique decoders with qualitatively worse parameters than our main result.

▶ **Definition 17.** *An oracle algorithm D is an $(r, \delta, \epsilon, \tau)$-*contextually unique decoder *for a family of probabilistic codes $\{C_i : \{0,1\}^k \xrightarrow{\$} \{0,1\}^n\}_{i \in \mathcal{I}}$ if for all sets $S \subseteq \{0,1\}^k$ with $|S| \leq 2^{rn}$, it holds with probability at least $1 - \epsilon$ over the choice of $i \leftarrow \mathcal{I}$ that for all messages $m \in S$ and all δ-Hamming adversaries $\mathcal{A}$,*

$$\Pr_{c \leftarrow C_i(m)}[D^S(i, \mathcal{A}(c)) \neq m] \leq \tau.$$

On the order of quantifiers

In the definitions above, we fix a family of codes, then say that for all small enough S, a random code from the family is good for S. In particular, we do not allow an adversary to choose S after the code is sampled. This may be problematic in some cases, but Definition 17 suffices in the common case of languages that are already established (but not perfectly compressible). This includes languages like "English sentences" or "images of dogs". In this case, since the set S is already in principle determined (albeit not fully understood), it suffices to pick and agree upon a random code from the family ahead of time, and always use that code in the future.

3.1 Inefficient Decoding

It is possible to show that for fixed deterministic codes, contextually unique decoding is no easier than unique decoding for the entire ambient message space. In Theorems 18 and 20, we show that randomly sampled codes can do better (albeit with an inefficient decoder).

A family of codes $\{C_i : \{0,1\}^k \to \{0,1\}^n\}_i$ is said to be pairwise independent if for all distinct $x, x' \in \{0,1\}^k$, the distribution of $\big(C_i(x), C_i(x')\big)$ for random i is uniform over $\{0,1\}^n \times \{0,1\}^n$. For instance, a random linear code is pairwise independent.

▶ **Theorem 18.** *Let $\{\mathcal{C}_n\}_{n \in \mathbb{Z}^+}$ be an ensemble of pairwise independent code families*[2]*, where each code in the family $\mathcal{C}_n$ has n-bit codewords. Then for all $r, \delta \in (0,1)$ with $H(2\delta) + 2r < 1$, there is a $(r, \delta, \exp(-\Omega(n)), 0)$-contextually unique decoder for $\mathcal{C}_n$.*

Proof. Let S be a message space with $|S| \leq 2^{rn}$. We will show that with all but $\exp(-\Omega(n))$ probability over the choice of code $C \leftarrow \mathcal{C}_n$, the restriction $C|_S$ of C to S has relative distance 2δ.

For any distinct $m, m' \in S$, it follows from pairwise independence and Fact 9 that

$$\Pr_C[C(m) \approx_{2\delta} C(m')] \leq \frac{\tilde{O}(2^{H(2\delta)n})}{2^n} \leq \frac{1}{\tilde{\Omega}(2^{(1-H(2\delta))\cdot n})}.$$

Union bounding over all pairs of m, m',

$$\Pr_C[\exists m, m' \in S \text{ s.t. } m \neq m' \text{ and } C(m) \approx_{2\delta} C(m')] \leq \frac{1}{\tilde{\Omega}(2^{(1-2r-H(2\delta))\cdot n})} \leq \exp(-\Omega(n)). \blacktriangleleft$$

[2] Note that we do not explicitly make any assumption on the rate vs. distance tradeoff of $\mathcal{C}_n$; instead, we implicitly use the fact that any code drawn from a pairwise independent family has relatively good distance with high probability.

Discussion

One interesting aspect of Theorem 18 is that it demonstrates a family of codes with an "apparent rate" that is independent of the number of tolerable errors, as long as the "true" message space is sufficiently sparse. For example, each $\mathcal{C}_n$ might map $\{0,1\}^{2n} \to \{0,1\}^n$, yet as long as the $2n$-bit messages have some structure that is known to the receiver (not necessarily to the sender!), it can be guaranteed that the receiver will reconstruct the sender's message.

However, the parameters achieved by Theorem 18 are not optimal. In particular, its error-tolerance is not competitive with the alternative approach of first compressing messages in S into rn-bit representations, and then applying a good error-correcting code to this representation. The GV bound for binary codes states that there exist codes with rate r and relative distance 2δ whenever $H(2\delta) + r < 1$. It is consistent with current knowledge that this bound is tight.

Our next result closes this gap by sampling a probabilistic code rather than a deterministic code.

▶ **Construction 19.** *Let $C_1, \ldots, C_N : \{0,1\}^k \to \{0,1\}^n$ be deterministic binary codes. We define the probabilistic code $C_{\mathsf{mix}}[C_1, \ldots, C_N] : \{0,1\}^k \overset{\$}{\to} \{0,1\}^n$ so that $C_{\mathsf{mix}}[C_1, \ldots, C_N](m)$ is $C_i(m)$ for a uniformly random $i \leftarrow [N]$.*

▶ **Theorem 20.** *Let $\{\mathcal{C}_n\}_{n\in\mathbb{Z}^+}$ be an ensemble, where $\mathcal{C}_n$ is a pairwise independent family of codes with n-bit codewords.*

For all $r, \delta \in (0,1)$ with $H(2\delta) + r < 1$ and any $\tau \geq n^{-O(1)}$, there exists $N \leq n^{O(1)}$ such that there is an (inefficient) $(r, \delta, \exp(-\Omega(n)), \tau)$-contextually unique decoder for $\{C_{\mathsf{mix}}[C_1, \ldots, C_N]\}_{C_i \in \mathcal{C}_n}$

In other words, $\mathcal{C}_n$ pairwise independent family of codes with n-bit codewords, and the code we use is $\{C_{\mathsf{mix}}[C_1, \ldots, C_N]\}_{C_i \in \mathcal{C}_n}$, where the $C_1, \ldots, C_N$ are randomly chosen elements of $\mathcal{C}_n$.

Proof Overview. Suppose that a sender encodes a message m, and the receiver gets an adversarially perturbed codeword c'. We define the (inefficient) decoder so that it finds all i' and all $m' \in S$ for which $C_{i'}(m')$ is within distance δn of c'. We claim that with high probability, the only such (i', m') is in fact (i, m).

To see this, we first fix m, and consider two different ways in which an encoding of m can be confused for an encoding of a different message. Using Fact 9 one can show that, for each i:

1. The probability over the choice of C_i that there exists $m' \in S \setminus \{m\}$ such that $C_i(m')$ and $C_i(m)$ are within Hamming distance $2\delta n$ is at most $2^{(r+H(2\delta)-1)n}$.
2. The probability over $C_1, \ldots, C_{i-1}, C_{i+1}, \ldots, C_N$ that there exists $m' \in S \setminus \{m\}$ and $i' \neq i$ such that $C_{i'}(m')$ and $C_i(m)$ are within Hamming distance δn is at most $N \cdot 2^{(r+H(2\delta)-1)n}$.

At this point, unless $H(2\delta) + 2r < 1$, we cannot simply apply a union bound to argue that with high probability $C_i(m)$ and $C_{i'}(m')$ are $2\delta n$-far for all $m \neq m'$.

To rely only on the weaker condition that $H(2\delta) + r < 1$, the key insight is that for any fixed m, "most" (all but a τ fraction) of C_i's will be good in the above sense with all but $2^{-(r+\Omega(1))\cdot n}$ probability. To show this, we must set N to be a sufficiently large polynomial and use Azuma's inequality (rather than Chernoff) because the events $\{(2)$ holds for $i\}_i$ are not mutually independent. After this, the probability $2^{-(r+\Omega(1))\cdot n}$ is sufficiently small that we can union bound over all 2^{rn} choices of m. ◀

Rather than elaborating on the details here, we instead defer to our full proof of Theorem 21, which uses the same approach.

Discussion

Unlike Theorem 18, Theorem 20 matches (other than the arbitrarily small inverse polynomial probability of decoding error) the rate vs. error tolerance tradeoff that is known to be achievable with inefficient decoding for *known, efficiently compressible* sets S (the GV bound).

3.2 Efficient Decoding with Noticeable Error

To obtain an efficient contextually-unique decoder, we adapt the ideas of Theorem 20. Instead of using a pairwise independent family of codes (which is not efficiently decodable), we use a "random" efficiently list-decodable code. Specifically, we use a fixed efficiently list-decodable deterministic code, composed with a random (efficiently evaluable and invertible) permutation π.

Recall the definition of C_{mix} from Construction 19.

▶ **Theorem 21.** *Let $\{C'_n : \{0,1\}^{k_n} \to \{0,1\}^n\}_{n\in\mathbb{Z}^+}$ be a rate-r' ensemble of (deterministic) binary codes that is efficiently ρ-list decodable and combinatorially $(2\delta, \lambda)$-list decodable.*

Then, for any $r < r' - \lambda$ and any $\tau(n) \geq n^{-O(1)}$, there exists $N(n) \leq n^{O(1)}$ such that for any pairwise independent family Π_n of permutations of $\{0,1\}^{k_n}$, the family of codes $\{C_{\mathsf{mix}}[C'_n \circ \pi_1, \ldots, C'_n \circ \pi_N]\}_{\pi_1,\ldots,\pi_N\in\Pi_n}$ has a $(r, \delta, \exp(-\Omega(n)), \tau)$-contextually unique decoder.

Discussion

The main advantage of Theorem 21 over Theorem 20 is that the decoder can run in $\mathrm{poly}(n)$ time. The main disadvantage compared to Theorem 18 is that the probability of incorrectly decoding is relatively high; in particular, the length of the description of a code (and therefore also the encoder's and decoder's running times) are inversely proportional to the error probability.

Our proof of Theorem 21 relies on the following version of the Azuma-Hoeffding inequality, which can be found as Equation (3) in [23]:

▶ **Imported Theorem 22** (Azuma-Hoeffding). *Let $\{X_k\}_{k=0}^{\infty}$ be a real-valued martingale with $a_k \leq X_k - X_{k-1} \leq b_k$. Then for every $r \geq 0$,*

$$\Pr[|X_n - X_0| \geq t] \leq 2 \cdot \exp\left(-\frac{2t^2}{\sum_{k=1}^{n}(b_k - a_k)^2}\right).$$

We now commence the proof of Theorem 21.

Proof. We first describe the decoding algorithm. We are given as input a corrupted codeword $y \in \{0,1\}^n$, and given oracle access to a set S of "valid messages". We run the list-decoding algorithm for C'_n on y to obtain a list of codewords $c_i = C'_n(m_i)$ for $i = 1, \ldots, L$. We find $i \in [L], j \in [N]$ that $\pi_j^{-1}(m_i)$ is in S. If no such (i,j) exists, or if multiple such (i,j) exists, we reject (output $\bot$). Otherwise, we output $\pi_j^{-1}(m_i)$.

Let p_0 denote the quantity $2^{rn} \cdot \max_{c\in\{0,1\}^n} \Pr_{m\leftarrow\{0,1\}^{r'n}}[\Delta(C_n(m), c) \leq 2\delta n]$. Using a union bound, we can see that p_0 bounds the probability, for any fixed c, that $C'_n(y)$ is $2\delta n$-close to c for *any* of 2^{rn} different uniformly random y. The combinatorial list-decodability of $\{C'_n\}$ implies that $p_0 \leq 1/\tilde{\Omega}\big(2^{(r'-r-\lambda)\cdot n}\big)$. By assumption on r, this decreases exponentially with n, and in particular for any $N \leq n^{O(1)}$, it holds that

$$\tau \geq \omega(p_0 \cdot N^2). \tag{1}$$

Let $N(n)$ be a sufficiently large polynomial such that $2\tau^2 N - \ln(2) \cdot rn \geq \Omega(n)$.

▷ Claim 23. For any permutations $\pi_1, \ldots, \pi_N$, let $C_{\pi_1,\pi_2,\ldots,\pi_N}$ denote $C_{\mathsf{mix}}[C'_n \circ \pi_1, \ldots, C'_n \circ \pi_N]$. For every $m \in S$, it holds that

$$\Pr_{\substack{\pi_1,\ldots,\pi_N \\ C:=C_{\pi_1,\pi_2,\ldots,\pi_N}}}\left[\begin{array}{l} \exists\, \delta\text{-Hamming adversary } \mathcal{A} \text{ s.t.} \\ \Pr_{c \leftarrow C(m)}[D^S(\mathcal{A}(c)) \neq m] \geq 3\tau \end{array}\right] \leq e^{-(2-o(1))\tau^2 N}$$
$$\leq 2^{-(r+\Omega(1))n}.$$

Proof. Consider the probability space defined by sampling $\pi_1, \ldots, \pi_N \leftarrow \Pi$. For each $i \in [N]$, define random variables

$$X_i^{(<)} = \begin{cases} 1 & \text{if } \exists j < i \text{ and } \exists m' \in S \text{ s.t. } C_n(\pi_j(m')) \approx_{2\delta} C_n(\pi_i(m)) \\ 0 & \text{otherwise.} \end{cases}$$

Define random variables $\{X_i^{(=)}\}_{i\in[N]}$ and $\{X_i^{(>)}\}_{i\in[N]}$ analogously – that is, replace the condition "$j < i$" by "$j = i$" or "$j > i$" respectively.

Note that $X_1^{(=)}, \ldots, X_N^{(=)}$ are mutually independent because $X_i^{(=)}$ depends only on π_i. The pairwise independence of Π and a union bound over all m' implies that for each i, $\Pr[X_i^{(=)} = 1] \leq p_0 \leq N \cdot p_0$.

The random variables $X_1^{(<)}, \ldots, X_N^{(<)}$ are not independent. However, conditioned on $X_1^{(<)}, \ldots, X_{i-1}^{(<)}$ (indeed on any value of $\pi_1, \ldots, \pi_{i-1}$) the pairwise independence of π_i and a union bound over $j < i$ and over m' implies that

$$\Pr[X_i^{(<)} = 1 | X_1^{(<)}, \ldots, X_{i-1}^{(<)}] \leq i \cdot p_0 \leq N \cdot p_0.$$

Similarly,

$$\Pr[X_i^{(>)} = 1 | X_{i+1}^{(>)}, \ldots, X_N^{(>)}] \leq (N - i) \cdot p_0 \leq N \cdot p_0.$$

Azuma's inequality (Imported Theorem 22) implies that

$$\Pr[\sum_i X_i^{(<)} \geq \tau N] \leq 2e^{-2(\tau - p_0 N^2)^2 N}$$
$$\leq e^{-(2-o(1))\tau^2 N} \qquad \text{by (1),}$$

and we obtain the same bound on $\Pr[\sum_i X_i^{(>)} \geq \tau N]$ and $\Pr[\sum_i X_i^{(<)} \geq \tau N]$. So

$$\Pr\Big[\sum_i \big(X_i^{(<)} + X_i^{(=)} + X_i^{(>)}\big) \geq 3\tau N\Big] \leq 3 \cdot e^{-(2-o(1))\tau^2 N} \leq e^{-(2-o(1))\tau^2 N},$$

which is equivalent to the statement of the claim. ◁

Theorem 21 follows from union bounding over all 2^{rn} values of $m \in S$. ◀

4 Main Theorem: Efficient Decoding with Negligible Error

In our previous constructions, we always had some inverse polynomial probability (over the choice of encoding randomness) of incorrectly decoding. We now show how to reduce this error probability to negligible by using a super-polynomial number of permutations, but preserving the polynomial-time efficiency of encoding and decoding. This is Theorem 25 below, from which Theorem 3 follows immediately (after using Theorem 14 and Theorem 16).

▶ **Construction 24.** *Let $C' : \{0,1\}^{r'n} \to \{0,1\}^n$ be a deterministic code, let $\Pi = \{\pi_k : \{0,1\}^{r'n-s} \to \{0,1\}^{r'n-s}\}_{k\in\mathcal{K}}$ be a family of efficiently evaluable and invertible permutations, and let $h : \{0,1\}^s \to \mathcal{K}$ be a hash function.*

We define a probabilistic code $C_{C',\Pi,h} : \{0,1\}^{r'n-s} \xrightarrow{\$} \{0,1\}^n$ that encodes a message $m \in \{0,1\}^{r'n-s}$ by picking $x \leftarrow \{0,1\}^s$ at random, and outputting $C'(\pi_{h(x)}(m), x)$.

▶ **Theorem 25.** *Suppose that:*

- *$C' : \{0,1\}^{r'n} \to \{0,1\}^n$ is a (deterministic) binary code that is efficiently ρ_e-list-decodable*
- *$\delta \le \rho_e$ and λ are such that C' is combinatorially $(2\delta, \lambda)$-list decodable.*
- *For some $t = t(n)$ and $s = s(n)$ satisfying $\Omega(n) \le t(n) \le n^{O(1)}$ and $\omega(\log n) \le s(n) \le o(n)$:*
  - *$\Pi = \{\pi_k : \{0,1\}^{r'n-s} \to \{0,1\}^{r'n-s}\}_{k\in\mathcal{K}}$ is a $(t+1)$-wise ϵ-dependent family of permutations with $\epsilon \le 2^{-n}$, and*
  - *$\mathcal{H} = \{h_i : \{0,1\}^s \to \mathcal{K}\}_{i\in\mathcal{I}}$ is a $2t$-wise independent hash family.*

Then the family $\{C_{C',\Pi,h}\}_{h\in\mathcal{H}}$ has an $(r, \delta, \exp(-\omega(n)), \frac{t}{2^s})$-contextually unique decoder for any $r < r' - \lambda$.

Proof. Let $S \subseteq \{0,1\}^{r'n-s}$ be any set of messages with $|S| \le 2^{rn}$. We describe the contextually unique decoding algorithm on input $y \in \{0,1\}^n$. First, the algorithm applies the efficient list-decoding algorithm to obtain all codewords $y'_1, \dots, y'_L$ of C' that are within relative Hamming distance ρ_e of y. Then each y'_i is parsed as $(\pi_{h(x)}(m_i), x)$. The decoding algorithm outputs any m_i that is in S. It is immediate from efficient list-decodability that there is at least one such m_i. We need to show that with high probability there is at most one such m_i.

We will rely on the following variant of the Chernoff bound for binary random variables, which does not require the random variables to be fully independent. Instead, it only requires bounding the probability that relatively small subsets of variables are simultaneously 1.

▶ **Imported Theorem 26** ([21]). *Let $X_1, \dots, X_N$ be $\{0,1\}$-valued random variables, let $0 < \beta < 1$, and let $0 < t < \beta N$. Then*

$$\Pr\left[\sum_{i=1}^{N} X_i \ge \beta N\right] \le \frac{1}{\binom{\beta N}{t}} \cdot \sum_{A \in \binom{[N]}{t}} \mathbb{E}\left[\prod_{i\in A} X_i\right].$$

We will write N to denote 2^s, and for brevity of notation we will view any hash function $h \in \mathcal{H}$ directly as the corresponding tuple of permutations $(\pi_{k_0}, \dots, \pi_{k_{N-1}})$, where $k_i = h(i)$.

It is sufficient to show that for $\tau(n) = \frac{t}{N}$, it holds for every $m \in S$ that

$$\Pr_{(\pi_1,\dots,\pi_N)\leftarrow\mathcal{H}}\left[\left|\{i : \exists j \in [N], m' \in S \setminus \{m\} \text{ s.t. } C'(\pi_i(m), i) \approx_{2\delta} C'(\pi_j(m'), j)\}\right| \ge \tau \cdot N\right]$$

is at most $2^{-rn} \cdot \exp(-\omega(n))$.

We will use the Chernoff variant to prove the above inequality. Let X_i denote the indicator random variable for the event

$$\exists j \in [N], m' \in S \setminus \{m\} \text{ s.t. } C'(\pi_i(m), i) \approx_{2\delta} C'(\pi_j(m'), j),$$

so what we want to bound is $\Pr\left[\sum_{i=1}^{N} X_i \geq \tau \cdot N\right]$.

For $i, j \in [N]$ and $m' \in S \setminus \{m\}$, let $Y_{i,j,m'}$ denote the indicator random variable for the event

$$C'(\pi_i(m), i) \approx_{2\delta} C'(\pi_j(m'), j).$$

Let $A \subseteq [N]$ be a subset of size $|A| = t$. Say $A = \{a_1, \ldots, a_t\}$. We have

$$\begin{aligned}\mathbb{E}\left[\prod_{a \in A} X_a\right] &\leq \mathbb{E}\left[\prod_{a \in A} \sum_{\substack{j \in [N] \\ m' \in S \setminus \{m\}}} Y_{a,j,m'}\right] \\ &= \sum_{\substack{j_1, \ldots, j_t \in [N] \\ m'_1, \ldots, m'_t \in S \setminus \{m\}}} \mathbb{E}\left[\prod_{i=1}^{t} Y_{a_i, j_i, m'_i}\right]. \qquad (2)\end{aligned}$$

We now would like to use the independence of $\mathcal{H}$ and of Π to equate $\mathbb{E}[\prod_i Y_{a_i,j_i,m'_i}]$ with $\prod_i \mathbb{E}[Y_{a_i,j_i,m'_i}]$. However this is not quite true, for two reasons. First, Π is only *approximately* $(t+1)$-wise independent. Second, Π is a family of $(t+1)$-wise (almost) independent *permutations*, rather than unstructured functions.

Still, an only slightly worse bound holds for $\mathbb{E}\left[\prod_{i=1}^{t} Y_{a_i,j_i,m'_i}\right]$. Conditioned on $\pi_{j_1}(m'_1)$, …, $\pi_{j_t}(m'_t)$ and $\pi_{a_1}(m), \ldots, \pi_{a_{i-1}}(m)$, the $2t$-wise independence of $\mathcal{H}$ and the $(t+1)$-wise ϵ-dependence of Π imply that the distribution of of $\pi_{a_i}(m)$ is $(\epsilon + \frac{t}{2^{r'n-s}})$-close to uniform over $\{0,1\}^{r'n-s}$. The combinatorial list-decodability of C' asserts that the number of y for which $C'(y) \approx_{2\delta} C'(\pi_{j_i}(m'), j_i)$ is at most $2^{\lambda \cdot n}$.

We can therefore continue bounding (2) as follows:

$$\begin{aligned}&\leq \sum_{\substack{j_1, \ldots, j_t \in [N] \\ m'_1, \ldots, m'_t \in S \setminus \{m\}}} \prod_{i=1}^{t} \left(\frac{2^{\lambda \cdot n}}{2^{r'n-s}} + \epsilon + \frac{t}{2^{r'n-s}}\right) \\ &\leq (N \cdot 2^{rn})^t \cdot 2^{(\lambda - r' + o(1)) \cdot nt} \\ &\leq \alpha(n)^t,\end{aligned}$$

where we define $\alpha(n) = N(n) \cdot \tilde{O}(2^{(\lambda + r - r' + o(1)) \cdot n})$, which is $\exp(-\Omega(n))$ by assumption on δ and r and because $N \leq 2^{o(n)}$. Thus for $\tau(n) = \frac{t}{N} \geq \omega(\alpha(n))$, it holds by Imported Theorem 26 that

$$\begin{aligned}\Pr\left[\sum_{i=1}^{N} X_i \geq \tau \cdot N\right] &\leq \frac{\binom{N}{t}}{\binom{\tau \cdot N}{t}} \cdot \alpha(n)^t \\ &\leq \left(\frac{e\alpha}{\tau}\right)^t \\ &\leq \exp(-\omega(t)) \\ &\leq \exp(-\omega(n)) \\ &\leq 2^{rn} \cdot \exp(-\omega(n)).\end{aligned}$$

Theorem 25 follows by union bounding over all 2^{rn} choices of $m \in S$. ◀

5 Future Directions

There are several interesting directions that we have not yet explored. We highlight a few below:

- How well is it possible to perform contextually unique decoding in different error models? For example, one might consider adversarial erasures, insertions, deletions, random errors, and so on.
- What are the optimal achievable parameters for contextually unique decoding?
- Is it possible to have a single probabilistic code that simultaneously works well for all message sets S of bounded size? If so, with what parameters?
- When $S = \{0,1\}^k$ padded with zeroes, can our construction be made explicit?

References

1 Mohammad Bavarian, Dmitry Gavinsky, and Tsuyoshi Ito. On the role of shared randomness in simultaneous communication. In *International Colloquium on Automata, Languages, and Programming*, pages 150–162. Springer, 2014.

2 E. L Blokh and Victor Zyablov. Linear concatenated codes. *Nauka*, 1982.

3 Clément L Canonne, Venkatesan Guruswami, Raghu Meka, and Madhu Sudan. Communication with imperfectly shared randomness. *IEEE Transactions on Information Theory*, 63(10):6799–6818, 2017.

4 Zitan Chen, Sidharth Jaggi, and Michael Langberg. The capacity of online (causal) q-ary error-erasure channels. *IEEE Transactions on Information Theory*, 65(6):3384–3411, 2019.

5 Ilya Dumer, Mark S. Pinsker, and Vyacheslav V. Prelov. Epsilon-entropy of an ellipsoid in a hamming space. *Probl. Inf. Transm.*, 38(1):1–15, 2002.

6 Peter Elias. List decoding for noisy channels. Technical Report 335, Research Laboratory of Electronics, MIT, 1957.

7 Badih Ghazi, Ilan Komargodski, Pravesh Kothari, and Madhu Sudan. Communication with contextual uncertainty. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 2072–2085. SIAM, 2016.

8 Badih Ghazi and Madhu Sudan. The power of shared randomness in uncertain communication. In *ICALP*, volume 80 of *LIPIcs*, pages 49:1–49:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

9 Oded Goldreich, Brendan Juba, and Madhu Sudan. A theory of goal-oriented communication. *Journal of the ACM (JACM)*, 59(2):8, 2012.

10 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.

11 Ofer Grossman and Justin Holmgren. Error correcting codes for uncompressed messages. *Electron. Colloquium Comput. Complex.*, 27:38, 2020.

12 Venkatesan Guruswami. List decoding with side information. In *IEEE Conference on Computational Complexity*, page 300. IEEE Computer Society, 2003.

13 Venkatesan Guruswami. Algorithmic results in list decoding. *Theoretical Computer Science*, 2(2):107–195, 2006.

14 Venkatesan Guruswami and Atri Rudra. Better binary list decodable codes via multilevel concatenation. *IEEE Transactions on Information Theory*, 55(1):19–26, 2009.

15 Venkatesan Guruswami and Adam Smith. Optimal rate code constructions for computationally simple channels. *Journal of the ACM (JACM)*, 63(4):1–37, 2016.

16 Elad Haramaty and Madhu Sudan. Deterministic compression with uncertain priors. *Algorithmica*, 76(3):630–653, 2016.

17 Brett Hemenway, Noga Ron-Zewi, and Mary Wootters. Local list recovery of high-rate tensor codes & applications. In *FOCS*, pages 204–215. IEEE Computer Society, 2017.

18 Brendan Juba, Adam Tauman Kalai, Sanjeev Khanna, and Madhu Sudan. Compression without a common prior: an information-theoretic justification for ambiguity in language. In *ICS*, pages 79–86. Tsinghua University Press, 2011.

19 Eyal Kaplan, Moni Naor, and Omer Reingold. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009.

20 Michael Langberg. Private codes or succinct random codes that are (almost) perfect. In *FOCS*, pages 325–334. IEEE Computer Society, 2004.

21 Nathan Linial and Zur Luria. Chernoff's inequality - a very elementary proof, 2014. `arXiv:1403.7739`.

22 Atri Rudra. *List decoding and property testing of error correcting codes*. PhD thesis, University of Washington, 2007.

23 Igal Sason. On refined versions of the Azuma-Hoeffding inequality with applications in information theory. *arXiv preprint*, 2011. `arXiv:1111.1977`.

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

25 Madhu Sudan. List decoding: Algorithms and applications. In *IFIP International Conference on Theoretical Computer Science*, pages 25–41. Springer, 2000.

26 C. Thommesen. The existence of binary linear concatenated codes with Reed-Solomon outer codes which asymptotically meet the Gilbert-Varshamov bound. *IEEE Transactions on Information Theory*, 29(6):850–853, November 1983. `doi:10.1109/tit.1983.1056765`.

27 Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *J. Comput. Syst. Sci.*, 22(3):265–279, 1981.

28 John M Wozencraft. List decoding. *Quarterly Progress Report*, 48:90–95, 1958.