





Interactive Error Correcting Codes: New Constructions and Impossibility Bounds

Meghal Gupta   

University of California Berkeley, CA, USA

Rachel Yun Zhang   

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

An *interactive error correcting code* (iECC) is an interactive protocol with the guarantee that the receiver can correctly determine the sender’s message, even in the presence of noise. It was shown in works by Gupta, Kalai, and Zhang (STOC 2022) and by Efremenko, Kol, Saxena, and Zhang (FOCS 2022) that there exist iECC’s that are resilient to a larger fraction of errors than is possible in standard error-correcting codes without interaction. In this work, we improve upon these existing works in two ways:

- First, we improve upon the erasure iECC of Kalai, Gupta, and Zhang, which has communication complexity quadratic in the message size. In our work, we construct the first iECC resilient to $> \frac{1}{2}$ adversarial erasures that is also positive rate. For any $\epsilon > 0$, our iECC is resilient to $\frac{6}{11} - \epsilon$ adversarial erasures and has size $O_\epsilon(k)$.
- Second, we prove a better upper bound on the maximal possible error resilience of any iECC in the case of bit flip errors. It is known that an iECC can achieve $\frac{1}{4} + 10^{-5}$ error resilience (Efremenko, Kol, Saxena, and Zhang), while the best known upper bound was $\frac{2}{7} \approx 0.2857$ (Gupta, Kalai, and Zhang). We improve upon the upper bound, showing that no iECC can be resilient to more than $\frac{13}{47} \approx 0.2766$ fraction of errors.

2012 ACM Subject Classification Mathematics of computing → Information theory

Keywords and phrases Code, Interactive Protocol, Error Resilience

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.32

Category RANDOM

Related Version This conference publication is the merge of two works.

Previous Version: <https://arxiv.org/abs/2201.11929> [10]

Previous Version: <https://arxiv.org/abs/2305.04376> [11]

Funding *Meghal Gupta:* U.C. Berkeley Chancellor’s Fellowship

Rachel Yun Zhang: This research was supported in part by DARPA under Agreement No. HR00112020023, an NSF grant CNS-2154149, an Akamai Presidential Fellowship, and NSF Graduate Research Fellowship 2141064.

1 Introduction

Consider the following task: Alice wishes to communicate a message to Bob such that even if a constant fraction of the communicated bits are adversarially tampered with, Bob is still guaranteed to be able to determine her message. This task motivated the prolific study of *error correcting codes*, starting with the seminal works of [16, 15]. An error correcting code encodes a message x into a longer codeword $\text{ECC}(x)$, such that the Hamming distance between any two distinct codewords is a constant fraction of the length of the codewords.

An important question in the study of error correcting codes is determining the maximal possible error resilience. It is known that in the adversarial bit-flip model, any ECC can be resilient to at most $\frac{1}{4}$ corruptions, and in the adversarial erasure error model any ECC can be resilient to at most $\frac{1}{2}$ corruptions.



© Meghal Gupta and Rachel Yun Zhang;

licensed under Creative Commons License CC-BY 4.0

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2023).

Editors: Nicole Megow and Adam D. Smith; Article No. 32; pp. 32:1–32:14



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

This prompts the following natural question: *Can we achieve better error resilience if we use interaction?*

In [9], Gupta, Kalai and Zhang introduce the notion of an *interactive error correcting code* (iECC), which is an interactive protocol with a fixed length and speaking order, such that Bob can correctly learn Alice’s input x as long as not too large a fraction of the *total communication* is erased. They demonstrate that iECC’s can in fact achieve a higher erasure resilience than standard error correcting codes. In particular, they design an iECC that is resilient to adversarial erasure of $\frac{3}{5} - \epsilon$ of the total communication.

Note that a classical error correcting code is an iECC in which Alice speaks in every round. Their result essentially shows that Bob talking occasionally *instead of Alice* actually improves the error resilience. It is not obvious that this should be the case – since Bob can only send feedback, while Alice can actually send new information, Bob’s messages a priori seem a lot less valuable than Alice’s. Nevertheless, they are able to leverage this to improve the erasure resilience past $\frac{1}{2}$. Later, [7] present an iECC that achieves an error (bit flip) resilience greater than $\frac{1}{4}$.

In this paper, we present two new results about iECC’s. We mention that this conference version of the paper is a combination of two separate works on the arXiv: [10] and [11]. We also note the relevant context that the result of [7], which constructed a binary iECC with $> \frac{1}{4}$ error resilience, was published after the positive rate result in this paper, but before the impossibility bound in this paper.

1.1 Positive Rate iECC

One weakness of the erasure-resilient iECC presented by [9] is that the size of their protocol is quadratic in the length of Alice’s original message x . This leaves open the question of whether there exists an iECC achieving $> \frac{1}{2}$ erasure resilience with size *linear* in the length of the original message. In this paper, we answer this question in the affirmative.

Specifically, we show a positive rate iECC that achieves an erasure resilience of $\frac{6}{11} - \epsilon$ over the binary erasure channel, which is larger than $\frac{1}{2}$.

► **Theorem 1.** *For any $\epsilon > 0$, there exists an iECC over the binary erasure channel resilient to $\frac{6}{11} - \epsilon$ erasures, such that the communication complexity for inputs of size n is $O_\epsilon(n)$ and the time complexity is $\text{poly}_\epsilon(n)$.*

We remark that our iECC achieves a lower erasure resilience than the quadratic sized iECC of [9], which is resilient to $\frac{3}{5} - \epsilon$ erasures. However, we believe that an iECC achieving both positive rate and $\frac{3}{5} - \epsilon$ erasure resilience can likely be constructed by combining ideas from this paper and [9]. Nevertheless, we leave open the existence of such an iECC.

► **Remark 2.** Since the original paper [10] was posted to arXiv, the work of [7] constructed a positive rate iECC resilient to $\frac{1}{4} + 10^{-5}$ bit flip errors, thereby resolving whether iECC’s can achieve better error resilience in the case of bit flip errors as well! Since bit flip errors are stronger than erasures, their iECC is also a positive rate iECC resilient to $> \frac{1}{2}$ erasures. Comparing this work with theirs, their works in the case of bit flips errors while ours does not, but our work achieves the higher erasure resilience.

1.2 Upper Bound on Maximal Error Resilience of iECC

The current best known upper (impossibility) bound for the error resilience of an iECC (in the bit flip setting, rather than erasure setting) was given in [9], who showed that no iECC can be resilient to more than $\frac{2}{7}$ adversarial errors. This upper bound came from the combination of two natural attacks, one of which is guaranteed to work no matter how the rounds in which Alice and Bob speak are distributed.

1. Corrupt *none* of Bob's bits. Then, Bob's messages provide perfect reliable feedback, in which case works about *error-correcting codes with feedback* beginning with [2] tell us that it suffices to corrupt $\frac{1}{3}$ of Alice's bits.
2. Corrupt *half* of Bob's bits so that his messages appear random and thus essentially are useless: then Alice's communication essentially reduces to the case of a standard error-correcting code, in which case an adversary can corrupt $\frac{1}{4}$ of her bits to confuse Bob between two possible values of x .

The largest error resilience known so far is achieved by [7], who construct an iECC resilient to $\frac{1}{4} + 10^{-5}$ bit flip errors. This leaves a large gap between the best known achievable error resilience and the best known upper bound. *What is the largest possible error resilience of an iECC? Is it possible to achieve error resilience equal to this natural upper bound of $\frac{2}{7}$?*

In this work, we answer the latter question in the negative, providing a new upper bound of $\frac{13}{47} \approx 0.2766$, improving upon the previous best upper bound of $\frac{2}{7} \approx 0.2857$.

► **Theorem 3.** *For sufficiently small $\epsilon > 0$, there exists $k_0 = k_0(\epsilon) \in \mathbb{N}$ such that for any $k > k_0$, no iECC over the binary bit flip channel where Alice is trying to communicate $x \in \{0, 1\}^k$ is resilient to $\frac{13}{47} + \epsilon$ fraction of adversarial bit flips.*

2 Related Works

In this section, we discuss previous work on interactive error-correcting codes, as well as prior work on error-correcting codes with feedback.

2.1 Interactive Error-Correcting Codes

The notion of an *interactive error-correcting code* (iECC) was first introduced in [9], who demonstrated an iECC resilient to $\frac{3}{5}$ fraction of adversarial erasures, surpassing the best possible erasure resilience of standard ECC's of $\frac{1}{2}$. They also gave an upper bound of $\frac{2}{3}$ on the erasure resilience of any iECC. In the case of bit flip errors, they proved an upper bound of $\frac{2}{7}$ on the error resilience achievable by any iECC, leaving open the problem of constructing an iECC resilient to greater than $\frac{1}{4}$ adversarial errors.

The followup work of [10] improved upon the erasure iECC of [9], giving a construction of an iECC with *positive rate* but resilient to only $\frac{6}{11}$ adversarial erasures.

In the bit flip error model, [7] answered [9]'s question in the affirmative, constructing an iECC with error resilience $\frac{1}{4} + 10^{-5}$. This narrowed the optimal error resilience of any iECC to the range $[\frac{1}{4} + 10^{-5}, \frac{2}{7}]$.

2.2 Error-Correcting Codes with Feedback

The use of interaction in the noise resilient communication of a message has been studied previously in the form of *error-correcting codes with feedback*. In an error-correcting code with feedback, Alice wishes to communicate a message to Bob in an error-resilient fashion, provided that after every message she sends she receives some feedback from Bob about what he has received. She can then use this noiseless feedback to choose the next bit that she sends. Error-correcting codes with feedback were first introduced in the Ph.D. thesis of Berlekamp [2] and have been studied in a number of followup works, including [3, 19, 17, 14, 8, 1]. Originally, this feedback was considered in the *noiseless* setting, meaning that none of Bob's messages are allowed to be corrupted, and error rate is calculated solely as a function of the number of messages Alice sends. That is, Bob's feedback is free and always correct, so that Alice can tailor her next message to specifically the bit of information Bob most needs to hear.

In the bit flip error model, [3, 19, 17, 14] showed that the maximal error resilience of an error-correcting code with noiseless feedback is $\frac{1}{3}$. [8] show this is achievable even by protocols that only send logarithmically many bits of feedback over a constant number of rounds. For explicit constant number of rounds of feedback, [4] initiated the study of the noise resilience vs. round complexity tradeoff for both erasures and errors. For larger alphabets, the maximal error resilience was studied in [1].

When the feedback is *noisy*, i.e. the feedback may be corrupted as well, much less is known. Several works such as [5, 6] considered ECC's with noisy feedback over the binary symmetric channel. [18] considers adversarial corruption, under a model which places separate corruption budgets on the forward and feedback rounds. They construct a scheme that is resilient to $\frac{1}{2}$ of the forward communication and 1 of the feedback being erased. We note that their scheme's forward erasure resilience is equal to that achievable by standard error-correcting codes.

3 Preliminaries and Definitions

Before we dive into the technical part of our paper, we present important preliminaries on classical error correcting codes, and define an iECC formally and what it means for one to be resilient to α -fraction of erasures.

Notation

In this work, we use the following notations.

- The function $\Delta(x, y)$ represents the Hamming distance between x and y .
- The interval $[a, b]$ for $a, b \in \mathbb{Z}_{\geq 0}$ denotes the integers from a to b inclusive. The interval $[n]$ denotes the integers $1, \dots, n$.
- The symbol \perp in a message represents the erasure symbol that a party might receive in the erasure model.
- When we say Bob k -decodes a message, we mean that he list decodes it to exactly k possible messages Alice could have sent in the valid message space.
- The output of an ECC is 0-indexed. All other strings are 1-indexed.

3.1 Interactive Error Correcting Codes

We formally define our notion of an *interactive error correcting code* (iECC). The two types of corruptions we will be interested in are erasures and bit flips. We first start by defining a non-adaptive interactive protocol.

► **Definition 4** (Non-Adaptive Interactive Protocol). *A non-adaptive interactive protocol $\pi = \{\pi_n\}_{n \in \mathbb{N}}$ is an interactive protocol between Alice and Bob, where in each round a single party sends a single bit to the other party. The order of speaking, as well as the number of rounds in the protocol, is fixed beforehand. The number of rounds is denoted $|\pi|$.*

► **Definition 5** (Interactive Error Correcting Code). *An interactive error correcting code (iECC) is a non-adaptive interactive protocol $\pi = \{\pi_n\}_{n \in \mathbb{N}}$, with the following syntax:*

- *At the beginning of the protocol, Alice receives as private input some $x \in \{0, 1\}^n$.*
- *At the end of the protocol, Bob outputs some $\hat{x} \in \{0, 1\}^n$.*

We say that π is α -resilient to adversarial bit flips (resp. erasures) if there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$ and $x \in \{0, 1\}^n$, and for all online adversarial attacks consisting of flipping (resp. erasing) at most $\alpha \cdot |\pi|$ of the total communication, Bob outputs x at the end of the protocol with probability 1.

3.2 Classical Error Correcting Codes

► **Definition 6** (Error Correcting Code). *An error correcting code (ECC) is a family of maps $\text{ECC} = \{\text{ECC}_n : \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)}\}_{n \in \mathbb{N}}$. An ECC has relative distance $\alpha > 0$ if for all $n \in \mathbb{N}$ and any $x \neq y \in \{0, 1\}^n$,*

$$\Delta(\text{ECC}_n(x), \text{ECC}_n(y)) \geq \alpha m(n).$$

Binary error correcting codes with relative distance $\approx \frac{1}{2}$ are well known to exist with linear blowup in communication complexity.

► **Theorem 7** ([13]). *For all $\epsilon > 0$, there exists an explicit linear error correcting code $\text{ECC}_\epsilon = \{\text{ECC}_{\epsilon, n} : \{0, 1\}^n \rightarrow \{0, 1\}^m\}_{n \in \mathbb{N}}$ with relative distance $\frac{1}{2} - \epsilon$ and with $m = m(n) = O_\epsilon(n)$. Furthermore, all codewords other than $\text{ECC}_{\epsilon, n}(0^n)$ are relative distance $\frac{1}{2} - \epsilon$ from 0^m and 1^m as well.*

A relative distance of $\frac{1}{2}$ is in fact optimal in the sense that as the number of codewords N approaches ∞ , the maximal possible relative distance between N codewords approaches $\frac{1}{2}$. We remark, however, that for small values of N , the distance can be much larger: for $N = 2$, the relative distance between codewords can be as large as 1, e.g. the codewords 0^M and 1^M , and for $N = 4$, the relative distance can be as large as $\frac{2}{3}$, e.g. the codewords $(000)^M, (110)^M, (101)^M, (011)^M$. Our constructions leverage this fact that codes with higher relative distance exist for a small constant number of codewords.

We will also need the following important lemma about the number of shared bits between any three codewords in an error correcting code scheme that has distance $\frac{1}{2}$.

► **Lemma 8.** *For any error correcting code $\text{ECC}_\epsilon = \{\text{ECC}_{\epsilon, n} : \{0, 1\}^n \rightarrow \{0, 1\}^m\}_{n \in \mathbb{N}}$ with relative distance $\frac{1}{2} - \epsilon$, and any large enough $n \in \mathbb{N}$, any three codewords in $\text{ECC}_{\epsilon, n}$ overlap on at most $(\frac{1}{4} + \frac{3}{2}\epsilon) \cdot m$ locations.*

Lemma 8 means that assuming that $< \frac{3}{4}$ of a codeword is erased, the resulting message is list-decodable to a set of size ≤ 2 , at least in theory. The following theorem says such a code exists with list-decoding being polynomial time, while also satisfying a couple other properties necessary in the protocol construction in Section 4.2.

► **Theorem 9** ([12]). *For all $\epsilon > 0$, any explicit (given with its encoding matrix) linear code $\text{ECC}_\epsilon = \{\text{ECC}_{\epsilon, n} : \{0, 1\}^n \rightarrow \{0, 1\}^m\}_{n \in \mathbb{N}}$ with relative distance $(\frac{1}{2} - \epsilon)$, can be efficiently decoded and list-decoded. That is, there exists a $\text{poly}_\epsilon(n)$ -time decoding algorithm $\text{DEC}_\epsilon = \{\text{DEC}_{\epsilon, n} : \{0, 1\}^m \rightarrow \mathcal{P}(\{0, 1\}^n)\}_{n \in \mathbb{N}}$, such that for any $n \in \mathbb{N}$, $x \in \{0, 1\}^n$, and corruption σ consisting of fewer than $(\frac{1}{2} - \epsilon) \cdot m$ erasures,*

$$x = \text{DEC}_{\epsilon, n}(\sigma \circ \text{ECC}_{\epsilon, n}(x)).$$

Moreover, for any corruption σ consisting of fewer than $(\frac{3}{4} - \frac{3}{2}\epsilon) \cdot m$ erasures,

$$|\text{DEC}_{\epsilon, n}(\sigma \circ \text{ECC}_{\epsilon, n}(x))| \leq 2, \quad x \in \text{DEC}_{\epsilon, n}(\sigma \circ \text{ECC}_{\epsilon, n}(x)).$$

Our following theorem gives an ECC such that any two codewords differ on most segments of length α .

► **Theorem 10.** For all $n \in \mathbb{N}, \epsilon > 0$, there exists $\alpha = \Theta_\epsilon(\log n)$ such that, there exists an explicit linear code $\text{ECC}_\epsilon = \{\text{ECC}_{\epsilon,n} : \{0, 1\}^n \rightarrow \{0, 1\}^m\}_{n \in \mathbb{N}}$ with $m = m(n) = O_\epsilon(n)$, satisfying the following property: For all $n \in \mathbb{N}, \epsilon > 0$, it holds that $\alpha | m$ and for any $x \neq x' \in \{0, 1\}^n$ and $j \in \{0 \dots \frac{m}{\alpha} - 1\}$,

$$\text{ECC}_{\epsilon,n}(x)[j\alpha : (j+1)\alpha - 1] = \text{ECC}_{\epsilon,n}(x')[j\alpha : (j+1)\alpha - 1]$$

for at most $\frac{\epsilon m}{\alpha}$ values of j .

► **Lemma 11.** Let $\text{ECC}'_\epsilon = \{\text{ECC}'_{n,\epsilon} : \{0, 1\}^n \rightarrow \{0, 1\}^{m(n)}\}$ be a explicit linear code satisfying the properties of Theorem 10 with $\alpha = \alpha_n = \Theta_\epsilon(\log n)$. For all linear $\text{ECC}_\epsilon = \{\text{ECC}_{n,\epsilon} : \{0, 1\}^\alpha \times \{0, 1\}^\beta \rightarrow \{0, 1\}^{p(n)}\}$ with relative distance $\frac{1}{2} - \epsilon$ and for all β , the code defined by

$$C(x) = \text{ECC}_\epsilon(\text{ECC}'_\epsilon(x)[0, \alpha - 1], 0^\beta) || \dots || \text{ECC}_\epsilon(\text{ECC}'_\epsilon(x)[m - \alpha, m - 1], 0^\beta)$$

is a linear code with relative distance $\frac{1}{2} - \frac{3}{2}\epsilon$. In particular, assuming that less than $\frac{3}{4} - \frac{9}{4}\epsilon$ of $C(x)$ is erased, there is an efficient algorithm to obtain a set of size 2 containing x .

4 A New Positive Rate iECC Resilient to 6/11 Erasures

In this section, we discuss our positive rate iECC that is resilient to 6/11 adversarial erasures.

4.1 Overview of Ideas

We begin with an overview of the ideas that go into our positive rate iECC. We first briefly review the iECC of [9] then describe how to modify it to have a linear communication complexity.

The overarching goal of the original protocol, as well as ours, is to perform the following three steps.

1. Bob learns that Alice's value of x is one of two possible values. (This idea is known as list decoding, which achieves better noise resilience than unique decoding.)
2. Bob conveys to Alice an index i on which the two possible inputs differ.
3. Alice sends the value of her input at index i .

Summary of the Protocol of [9]

The original protocol consists of many (say $\approx \frac{n}{\epsilon}$) *chunks*, where in each chunk Alice sends a message followed by Bob's reply. The protocol is designed so that each such chunk will make *progress* towards Bob's unambiguously learning Alice's input, as long as the adversary did not invest more than $\frac{3}{5} - \epsilon$ erasures in that chunk. At a high level, in the first chunk with $< \frac{3}{5} - \epsilon$ erasures, Bob narrows down Alice's input to at most two options. In every future chunk with $< \frac{3}{5} - \epsilon$ erasures, either Alice gets closer to learning the index i on which the two options differ, or Bob fully determines x by ruling out one of the two values of x , e.g. by learning the value of $x[i]$ or by uniquely decoding Alice's message. Alice keeps track of a counter cnt initially set to 0 indicating her guess for i . The main purpose of Bob's messages is to increment Alice's counter to i .

At the beginning of the protocol, Alice sends $\text{ECC}(x, \text{cnt})$ to Bob in every chunk. At the first point, there are $< \frac{3}{5} - \epsilon$ erasures in a chunk, Bob will be able to list decode Alice's message to at most two options, say $(x_0, \text{cnt}_0 = 0)$ and $(x_1, \text{cnt}_1 = 0)$. This must happen because the relative message lengths of Alice and Bob will be such that the adversary cannot

corrupt too much of Alice’s message even if they corrupt none of Bob’s message. Since we are in the setting of erasures, one of the two decodings must be Alice’s true state, and in particular, must contain Alice’s true input.

At this point, Bob begins signaling to Alice to increment cnt . His goal is to tell Alice to increment cnt until $\text{cnt} = i$. He does this by only sending one of two codewords¹ every message that have relative distance 1 apart. This way, if Bob’s message is not entirely erased, Alice learns what Bob tried to send. The key is that every time $< \frac{3}{5} - \epsilon$ of a chunk is corrupted, we can guarantee *both that Bob will decode Alice’s message to two possible messages, and Alice uniquely decodes Bob’s message*,² so that Alice and Bob make progress towards Alice learning i . Once Alice has discovered i , Bob signals for Alice to send the bit $x[i]$,³ which allows him to distinguish whether Alice has x_0 or x_1 .

Modifications to Achieve Positive Rate

The communication complexity of the above protocol is $O(n^2)$. This comes from two parts: (1) $O(n)$ chunks are necessary for Bob to communicate the index $i \in [n]$ to Alice via incrementation, and (2) Alice sends her length n input in every chunk. We show how to lessen both requirements, thus making the final protocol linear in length.

First, for Bob to communicate $i \in [n]$ to Alice, instead of incrementing cnt by 1 until it equals i . More specifically, we describe a process where Bob writes i out in binary, and then sends Alice each bit of this binary representation in sequence. This only requires $O(\log n)$ rounds of interaction, as opposed to the $O(n)$ rounds required by [9]. Designing a protocol to communicate a binary string rather than a unary string requires significant changes to the procedure used in [9].

Second, we show that instead of sending x every message, it suffices for Alice to encode a shorter string that is different than the corresponding short string for any other x' in *most* chunks. More precisely, consider an error correcting code ECC with the following property: set some α and for any $x \neq x'$,

$$\text{ECC}(x)[j\alpha, (j+1)\alpha - 1] \neq \text{ECC}(x')[j\alpha, (j+1)\alpha - 1]$$

for all but an ϵ fraction of values j . Then, Alice rotates through the sections, sending $\text{ECC}(x)[j\alpha, (j+1)\alpha - 1]$ in the $\left(j \bmod \frac{|\text{ECC}(x)|}{\alpha}\right)$ ’th chunk. Then, if Bob has narrowed down Alice’s input to x_0 and x_1 , he can simply ignore the ϵ fraction of chunks in which $\text{ECC}(x_0)[j\alpha, (j+1)\alpha - 1] = \text{ECC}(x_1)[j\alpha, (j+1)\alpha - 1]$. In the remainder of chunks, the segment $\text{ECC}(x)[j\alpha, (j+1)\alpha - 1]$ is sufficient for Bob to distinguish between x_0 and x_1 . If we were to let $\alpha \approx \log n$,⁴ then our chunks are now only length $O(\log n)$.

Combining the two modifications, we see that $\Theta(n)$ communication from Alice is necessary for Bob to narrow down Alice’s input to two options, and then after that, Bob can convey i to Alice in $O(\log n)$ chunks each of size $O(\log n)$. This results in an iECC with total communication $O(n + \log^2 n) = O(n)$.

¹ In our protocol, Bob will send one of four codewords each message. This contributes to the lower erasure resilience of $\frac{6}{11} - \epsilon$.

² It is also possible that instead Bob uniquely decodes Alice’s message, but then he will have uniquely learned x .

³ The reader familiar with the iECC of [9] may recall that in the case that the two Alices Bob sees have different values of cnt , Bob may instruct Alice to send a different bit for the rest of the protocol, but we do not address this for now.

⁴ $\alpha = \Theta(\log n)$ is necessary since Alice also sends her current guess of i each message, which has length $\log n$.

We remark that our protocol has erasure resilience $\frac{6}{11} - \epsilon$. The limiting factor is in the construction of a protocol in which Bob builds i bit by bit: our protocol requires Bob sending 4 codewords with distance $\frac{2}{3}$, rather than 2 codewords with distance $\frac{1}{2}$ to achieve $\frac{3}{5} - \epsilon$ erasure resilience. However, combining our second observation with the protocol from [9] would be enough to give an iECC with $\frac{3}{5} - \epsilon$ erasure resilience with communication $O(n \log n)$.

4.2 Positive Rate iECC Protocol (Informal)

Let $\text{ECC}' : \{0, 1\}^n \rightarrow \{0, 1\}^m$ be an error correcting code satisfying the statement of Theorem 10 with $\alpha = \Theta(\log n)$, and let $\text{ECC} : \{0, 1\}^\alpha \times \{0, 1\}^{\leq \log n} \rightarrow \{0, 1\}^p$ be an error correcting code with distance $\frac{1}{2}$ that is also relative distance $\frac{1}{2}$ from $0^p, 1^p$.

Our iECC consists of $O_\epsilon\left(\frac{m}{\alpha}\right)$ chunks, each consisting of Alice sending a p -bit message followed by Bob sending a $\frac{3p}{8}$ -bit message. Bob's messages are always one of four words $\bar{0}, \bar{1}, \bar{2}, \bar{3} \in \{0, 1\}^{3p/8}$ with relative distance $\frac{2}{3}$. We outline our protocol below. In what follows, we assume that all messages Bob receives are consistent with the same two values of x , otherwise Bob can rule out one of the values of x and determine Alice's true input.

1. Alice initially holds a string $\text{ind} \in \{0, 1\}^{\leq \log n}$ initially set to the empty string $\text{ind} = \emptyset$. Alice begins the protocol by sending $\text{ECC}(\text{ECC}'(x)[j\alpha, (j+1)\alpha - 1], \text{ind})$ to Bob in every chunk.
2. Bob begins the protocol sending $\bar{0}$ every message. Every $\frac{m}{\alpha}$ chunks, he attempts to list-decode Alice's previous $\frac{m}{\alpha}$ messages to find consistent values of x . Note that by Lemma 11, if there are at most $\frac{3}{4} - \frac{3}{2}\epsilon$ erasures in Alice's message in those $\frac{m}{\alpha}$ chunks, then Bob is guaranteed to find at most two possible values of x .
3. When Bob has found two consistent values of x , say \hat{x}_0 and \hat{x}_1 , he determines an index $i \in [n] = \{0, 1\}^{\log n}$ such that $\hat{x}_0[i] \neq \hat{x}_1[i]$. His goal is now to communicate i to Alice, bit by bit. He does this by sending either $\bar{0}, \bar{1}$, or $\bar{2}$ every chunk. To communicate the next'th bit of i , Bob adds $i[\text{next}] + 1$ to mes modulo 3, where $\overline{\text{mes}}$ was the last message he sent, to get his new message mes' , and begins sending $\overline{\text{mes}'}$ every chunk. (When Alice receives a message from Bob that is different from the last message she received, she can calculate the difference in the two messages to determine the bit.) He does this until he list-decodes Alice's message to two possibilities $\text{ECC}(\text{ECC}'(\hat{x}_0)[j\alpha, (j+1)\alpha - 1], \text{ind}_0)$ and $\text{ECC}(\text{ECC}'(\hat{x}_1)[j\alpha, (j+1)\alpha - 1], \text{ind}_1)$ such that $\text{ECC}'(\hat{x}_0)[j\alpha, (j+1)\alpha - 1] \neq \text{ECC}'(\hat{x}_1)[j\alpha, (j+1)\alpha - 1]$, where at least one of $\text{ind}_0, \text{ind}_1$ has length next . If both have length next , he proceeds to communicate the $(\text{next} + 1)$ 'th bit of i in the same way. If only one of the two Alice's has $|\text{ind}_b| = \text{next}$, Bob switches to sending $\bar{3}$ for the rest of the protocol, signaling to Alice to send him the parity of $|\text{ind}|$ so that he can distinguish between whether Alice has $(\hat{x}_0, \text{ind}_0)$ or $(\hat{x}_1, \text{ind}_1)$.
4. Whenever Alice unambiguously sees a *change* in Bob's message from a $\bar{0}$ to a $\bar{1}$ or $\bar{2}$ (or cyclic), she calculates $b = \text{mes}' - \text{mes} - 1 \pmod{3}$ and appends b to ind . If she ever receives a $\bar{3}$, she switches to sending $(|\text{ind}| \bmod 2)^p$ for the rest of the protocol. Otherwise, at some point she has $|\text{ind}| = \log n$, so she can convert ind into an index $i \in [n]$ and send $(x[i])^p$ for the rest of the protocol. Note that Bob can distinguish between \hat{x}_0 and \hat{x}_1 using the value of x at the index i .

In the above outline, one has to be careful around $|\text{ind}| = \log n - 1$. In particular, if one Alice has $|\text{ind}| = \log n - 1$ and the other has $|\text{ind}| = \log n$, the second will be sending $x[i]$ for the rest of the protocol and it is thus incorrect for Bob to send $\bar{3}$ to signal the first Alice to send the parity of the length of ind . Instead, once Bob has list-decoded Alice's message such that $|\text{ind}_0| = |\text{ind}_1| = \log n - 1$, Bob commits to sending the next message $\in \{\bar{0}, \bar{1}, \bar{2}\}$ that conveys to Alice the final bit of i for the rest of the protocol.

► **Theorem 12.** *The above iECC is resilient to a $\frac{6}{11} - O(\epsilon)$ fraction of erasures. For an input of size n , the total communication is $O_\epsilon(n)$. Alice and Bob run in $\text{poly}_\epsilon(n)$ time.*

The formal version of this protocol is excluded from this conference version of the paper. The full protocol can be found in [10].

5 Impossibility Bound on Maximal Noise Resilience of iECC

In this section, we present our main upper bound, that for any non-adaptive iECC, there is some attack consisting of at most $\frac{13}{47}$ corruptions such that Bob cannot guess Alice's input x correctly with probability better than $\frac{1}{2}$.

► **Theorem 13.** *main For sufficiently small $\epsilon > 0$, then for all $k > 100\epsilon^{-4}$, no iECC for $x \in \{0, 1\}^k$ is resilient to more than $\frac{13}{47} + 2\epsilon$ fraction of errors with probability greater than $\frac{1}{2}$.*

The rest of this section will be devoted to the proof of this theorem. Throughout this section, Alice's input will always be denoted $x \in \{0, 1\}^k$. The length of the iECC will be denoted by n .

At a high level, our proof will proceed as follows. We will split any candidate protocol into two sections, the first consisting of the first $\frac{21}{47}n$ rounds of the protocol, and the second consisting of the remaining $\frac{26}{47}n$ rounds of the protocol. In the first section, we denote the number of bits that Alice sends by A_1 , and the number that Bob sends by B_1 . Likewise, in the second section, we denote the number of bits that Alice and Bob send by A_2 and B_2 respectively. We will present three attacks in Sections 5.1, 5.2, and 5.3 such that depending on the values of A_1, B_1, A_2, B_2 , at least one attack is guaranteed to succeed while using at most $\frac{13}{47}$ corruptions.

Throughout this section, a *transcript* is the sequence of bits that is received by either of the parties. Note that since the adversary may corrupt messages, the transcript may be different than what was sent by Alice and Bob. We say that an attack *succeeds* with α corruption if there exist two inputs $x_1, x_2 \in \{0, 1\}^k$ along with respective strategies corrupting at most αn bits such that Bob's view of the transcript in both cases is identical. Then, Bob cannot guess Alice's true value of $x \in \{x_1, x_2\}$ with probability better than $\frac{1}{2}$.

5.1 Attack 1

In the first attack, the adversary behaves the same on both sections of the protocol. She corrupts Alice's bits while leaving Bob's untouched, such that there exist two inputs for which at most of $\frac{1}{3}$ of Alice's communication is corrupted. We remark that this attack has been known since [2].

► **Lemma 14.** *For any protocol consisting of A bits from Alice and B bits from Bob, and for any three possible inputs x_1, x_2, x_3 , there exists two of the three inputs $y_1, y_2 \in \{x_1, x_2, x_3\}$ and a transcript $T \in \{0, 1\}^{A+B}$ such that the adversary can corrupt at most $\frac{1}{3}A + 1$ bits so that the protocol transcript is T in both the case Alice has y_1 or y_2 .*

The proof is omitted in this conference version.

The attack is stated below.

Attack 1

Let $x_1, x_2, x_3 \in \{0, 1\}^k$ be three of Alice's possible inputs. By Lemma 14, there exist $y_1, y_2 \in \{x_1, x_2, x_3\}$ and transcript $T \in \{0, 1\}^n$ such that the adversary can corrupt at most $\frac{1}{3}(A_1 + A_2) + 1$ bits to obtain transcript T in both the case Alice has y_1 and if she has y_2 . The adversary simply corrupts the protocol so that the resulting transcript is T .

► **Lemma 15.** *Attack 1 succeeds with corrupting $\frac{1}{3}A_1 + \frac{1}{3}A_2 + 1$ bits.*

Proof. This follows immediately from Lemma 14: regardless of whether Alice has y_1 or y_2 , the adversary is able to have Bob receive the same transcript, using $\frac{1}{3}(A_1 + A_2) + 1$ bits of corruption. ◀

5.2 Attack 2

In our second attack, the adversary behaves differently in the two sections of the protocol. In the first section, the adversary essentially causes Bob's feedback to look random, so that Alice can do no better than to send a distance $\frac{1}{2}$ error-correcting code. This allows the adversary to corrupt $\frac{1}{4}$ of Alice's bits during this first section so that Bob cannot distinguish between three inputs. Then, in the second section, we use Lemma 14 from the previous section to show that the adversary has a strategy corrupting only $\frac{1}{3}A_2$ bits to confuse Bob between two of the remaining three inputs.

To argue that the adversary can perform her attack in the first section, we need the following lemma.

► **Lemma 16.** *For any $0 < \epsilon < 0.1$, suppose Alice has K possible inputs where $K > (4/\epsilon)^{1/3}$. Then for any protocol consisting of A bits from Alice and B bits from Bob where $A + B \geq \frac{3 \log(1/\epsilon)}{\epsilon^3}$, there exist three inputs x_1, x_2, x_3 and a transcript T such that regardless of which of x_1, x_2, x_3 Alice has as input, the adversary can corrupt at most $(\frac{1}{4} + \frac{3\epsilon}{2}) \cdot A + (\frac{1}{2} + \epsilon) \cdot B + 1$ bits so that the protocol transcript is T .*

The proof is omitted in this conference version.

We now state our second attack.

Attack 2

Let inputs $x_1, x_2, x_3 \in \{0, 1\}^k$ and transcript $T_1 \in \{0, 1\}^{21n/47}$ be such that they satisfy Lemma 16 for the first section of the protocol. Then, for the first section of the protocol, the adversary corrupts the transcript to look like T_1 , using at most $(\frac{1}{4} + \frac{3\epsilon}{2})A_1 + (\frac{1}{2} + \epsilon)B_1 + 1$ bits of corruption in the cases where Alice had x_1, x_2, x_3 .

For the second section of the protocol, the adversary corrupts the communication to the transcript $T_2 \in \{0, 1\}^{26n/47}$ as found in Lemma 14 such that there exist two of x_1, x_2, x_3 , denoted y_1, y_2 , for which the adversary can corrupt at most $\frac{1}{3}A_2 + 1$ of the communication so that the transcript of received bits is T_2 if Alice has y_1 or y_2 .

► **Lemma 17.** *Suppose that $k \geq \frac{141 \log(1/\epsilon)}{21\epsilon^3}$. Attack 2 succeeds with corrupting $(\frac{1}{4} + \frac{3\epsilon}{2}) \cdot A_1 + (\frac{1}{2} + \epsilon) \cdot B_1 + \frac{1}{3}A_2 + 2$ bits.*

Proof. Regardless of whether Alice has y_1 or y_2 , the transcript from Bob's perspective when the adversary employs this attack looks like T_1 followed by T_2 (restricted to Bob's viewpoint). Since $A_1 + B_1 \geq \max\{\frac{21}{26}A_2, k - A_2\}$ (where $A_1 + A_2 \geq k$ holds since Alice needs to send k

bits to communicate x , even noiselessly), it follows that $A_1 + B_1 \geq \frac{21}{47}k \geq 3 \log(1/\epsilon)/\epsilon^3$, so the condition of Lemma 16 is satisfied. Then, by Lemma 16, the number of corruptions used in the first section of the protocol when Alice has y_1 or y_2 is at most $(\frac{1}{4} + \frac{3\epsilon}{2}) \cdot A_1 + (\frac{1}{2} + \epsilon) \cdot B_1$, and by Lemma 14, the number of corrupted bits in the second section whether Alice has y_1 or y_2 is at most $\frac{1}{3}A_2 + 1$. \blacktriangleleft

5.3 Attack 3

In our third attack, we employ the following strategy. At a high level, we choose two inputs x_1 and x_2 . In the first section of the protocol, Bob's view is as if Alice had x_1 , while Bob's bits are corrupted so that Alice thinks that he has been receiving and responding correctly. In the second section of the protocol, Bob's bits are flipped randomly, and Alice's communication is corrupted to look like she has x_2 .

The first lemma we will need is to show that for the first section of the protocol, there are many inputs for which the uncorrupted transcripts have pairwise small Hamming distance.

► **Lemma 18.** *Let $\epsilon > 0$ and suppose Alice has K possible inputs. Then for any protocol consisting of A bits from Alice and B bits from Bob, there exists a set Γ of size $K'_\epsilon(K) = K^\epsilon - \frac{1}{\epsilon}$ inputs such that for any two $x_1, x_2 \in \Gamma$, the relative distance of the (uncorrupted) transcripts in the case where Alice has x_1 or x_2 is $\leq (\frac{1}{2} + \epsilon) \cdot (A + B)$.*

The proof is omitted in this conference version.

► **Lemma 19.** *Let $\epsilon > 0$, and suppose Alice has $K' > \sqrt{2/\epsilon}$ possible inputs. For any protocol consisting of A bits from Alice and B bits from Bob such that $A + B > \frac{3 \log(1/\epsilon)}{\epsilon^3}$, there exist two inputs x_1, x_2 such that for any advice α that Bob receives at the beginning of the protocol (after both Alice and Bob have fixed their strategies), there exist two transcripts T_1, T_2 such that the Bob's view of the two transcripts is the same, and that in the case of Alice having x_1 , the adversary needs only corrupt $(\frac{1}{2} + 2\epsilon)A + (\frac{1}{2} + \epsilon)B$ bits to get transcript T_1 , and in the case of Alice having x_2 , the adversary needs only corrupt $(\frac{1}{2} + \epsilon)B$ bits so that the transcript is T_2 .*

The proof is omitted in this conference version.

Attack 3

Denote by $T_1(y)$ the uncorrupted transcript corresponding to Alice having input y in the first section of the protocol. By Lemma 18, there exists a set M of $2^{\epsilon k} - \frac{1}{\epsilon}$ inputs such that for every $y_1, y_2 \in M$, it holds that $\Delta(T_1(y_1), T_1(y_2)) \leq (\frac{1}{2} + \epsilon) \cdot n$. Next, consider the second section of the protocol, conditioned on Alice having seen $T_1(x)$ (restricted to her view) in the first section of the protocol. By Lemma 19 there exist $x_1, x_2 \in M$ such that no matter what advice α Bob receives at the beginning of this second section, there exist transcripts $T_{2,1}(\alpha)$ and $T_{2,2}(\alpha)$ such that Bob's view of the two transcripts are the same, and these $T_{2,1}(\alpha), T_{2,2}(\alpha)$ satisfy the properties listed in Lemma 19.

In the first section of the protocol, the adversary corrupts the communication so that Bob always receives $T_1(x_1)$ (restricted to the bits that Bob sees), and so that Alice receives $T(x)$ (restricted to the bits that she sees), where x denotes Alice's input.

In the second section of the protocol, the adversary corrupts the communication so that the transcript is $T_{2,1}(\alpha = T_1(x_1))$ in the case of Alice having x_1 , and $T_{2,2}(\alpha = T_1(x_1))$ otherwise.

► **Lemma 20.** *Suppose that $k > \frac{141 \log(1/\epsilon)}{26\epsilon^3}$. Attack 3 succeeds with corrupting*

$$\max \left\{ \left(\frac{1}{2} + 2\epsilon \right) \cdot A_2 + \left(\frac{1}{2} + \epsilon \right) \cdot B_2, \left(\frac{1}{2} + \epsilon \right) \cdot A_1 + \left(\frac{1}{2} + \epsilon \right) \cdot B_1 + \left(\frac{1}{2} + \epsilon \right) \cdot B_2 \right\}$$

bits.

Proof. Regardless of whether Alice has x_1 or x_2 , Bob receives the same transcript (restricted to his view). Since $A_2 + B_2 \geq \max\{\frac{26}{21}A_1, k - A_1\}$ (where $A_1 + A_2 \geq k$ holds since Alice needs to send k bits to communicate x , even noiselessly), it follows that $A_2 + B_2 \geq \frac{26}{47}k > \frac{3 \log(1/\epsilon)}{\epsilon^3}$, so the condition of Lemma 19 is satisfied. If Alice has x_1 , the amount of corruption in the first section is 0, while in the second section the adversary corrupted at most $(\frac{1}{2} + 2\epsilon)A_2 + (\frac{1}{2} + \epsilon) \cdot B_2$ bits. If Alice has x_2 , the amount of corruption in the first section is $\Delta(T_1(x_1), T_1(x_2)) \leq (\frac{1}{2} + \epsilon) \cdot (A_1 + B_1)$, and in the second section the adversary corrupts at most $(\frac{1}{2} + \epsilon) \cdot B_2$ bits. ◀

5.4 Proof of Theorem 13

In this section, we prove our main result, restated below.

► **Theorem 21.** *For sufficiently small $\epsilon > 0$, there exists $k_0 = k_0(\epsilon) \in \mathbb{N}$ such that for any $k > k_0$, no iECC over the binary bit flip channel where Alice is trying to communicate $x \in \{0, 1\}^k$ is resilient to $\frac{13}{47} + \epsilon$ fraction of adversarial bit flips.*

We begin with the following lemma.

► **Lemma 22.** *For any nonnegative $a_1, b_1, a_2, b_2 \in \mathbb{R}$ where $a_1 + b_1 = \frac{21}{47}$ and $a_1 + b_1 + a_2 + b_2 = 1$, define*

$$\begin{aligned} \delta_1 &= \frac{1}{3}a_1 + \frac{1}{3}a_2, \\ \delta_2 &= \frac{1}{4}a_1 + \frac{1}{2}b_1 + \frac{1}{3}a_2, \\ \delta_3 &= \max \left\{ \frac{1}{2}a_2 + \frac{1}{2}b_2, \frac{1}{2}a_1 + \frac{1}{2}b_1 + \frac{1}{2}b_2 \right\}. \end{aligned}$$

It holds that

$$\min\{\delta_1, \delta_2, \delta_3\} \leq \frac{13}{47}.$$

Proof. Using that $b_1 = \frac{21}{47} - a_1$ and $b_2 = \frac{26}{47} - a_2$, we can substitute:

$$\begin{aligned} \delta_1 &= \frac{1}{3}a_1 + \frac{1}{3}a_2, \\ \delta_2 &= \frac{21}{94} - \frac{1}{4}a_1 + \frac{1}{3}a_2, \\ \delta_3 &= \max \left\{ \frac{13}{37}, \frac{1}{2} - \frac{1}{2}a_2 \right\}. \end{aligned}$$

Then,

$$\min\{\delta_1, \delta_2, \delta_3\} \leq \frac{13}{47} \iff \min\{\delta_1, \delta_2, \delta'_3\} \leq \frac{13}{47},$$

where $\delta'_3 = \frac{1}{2} - \frac{1}{2}a_2$. But note that

$$\frac{9}{35}\delta_1 + \frac{12}{35}\delta_2 + \frac{2}{5}\delta'_3 = \frac{13}{47}$$

where the weights $\frac{9}{35}, \frac{12}{35}, \frac{2}{5}$ sum to 1, so at least one of $\delta_1, \delta_2, \delta'_3$ must be at most $\frac{13}{47}$. ◀

Proof of Theorem 13. Recall that the length of the iECC is $n := A_1 + B_1 + A_2 + B_2 \geq A_1 + A_2 \geq k > \epsilon^{-3}$ (since Alice needs to send at least k bits to communicate x , even in the noiseless setting). Our goal is to show that regardless of the values of A_1, B_1, A_2, B_2 , at least one of Attacks 1, 2, and 3 will require at most $(\frac{13}{47} + 2\epsilon) \cdot n$ corruptions.

By Lemma 15, Attack 1 succeeds using

$$\frac{1}{3}A_1 + \frac{1}{3}A_2 + 1 \leq (\delta_1 + 2\epsilon)n$$

bits of corruption, where $\delta_1 := (\frac{1}{3}A_1 + \frac{1}{3}A_2)/n$.

By Lemma 17, Attack 2 succeeds using

$$\left(\frac{1}{4} + \frac{3\epsilon}{2}\right) \cdot A_1 + \left(\frac{1}{2} + \epsilon\right) \cdot B_1 + \frac{1}{3}A_2 + 2 \leq \frac{1}{4}A_1 + \frac{1}{2}B_1 + \frac{1}{3}A_2 + 2\epsilon n = (\delta_2 + 2\epsilon)n$$

bits of corruption, where we define $\delta_2 = (\frac{1}{4}A_1 + \frac{1}{2}B_1 + \frac{1}{3}A_2)/n$.

By Lemma 20, Attack 3 succeeds using

$$\begin{aligned} & \max \left\{ \left(\frac{1}{2} + 2\epsilon\right) \cdot A_2 + \left(\frac{1}{2} + \epsilon\right) \cdot B_2, \left(\frac{1}{2} + \epsilon\right) \cdot A_1 + \left(\frac{1}{2} + \epsilon\right) \cdot B_1 + \left(\frac{1}{2} + \epsilon\right) \cdot B_2 \right\} \\ & \leq \max \left\{ \frac{1}{2}A_2 + \frac{1}{2}B_2 + 2\epsilon n, \frac{1}{2}A_1 + \frac{1}{2}B_1 + \frac{1}{2}B_2 + 2\epsilon n \right\} \\ & = (\delta_3 + 2\epsilon)n \end{aligned}$$

bits of corruption, where we define $\delta_3 = \max \left\{ \frac{1}{2}A_2 + \frac{1}{2}B_2, \frac{1}{2}A_1 + \frac{1}{2}B_1 + \frac{1}{2}B_2 \right\} / n$.

By Lemma 22, we have that $\min\{\delta_1, \delta_2, \delta_3\} \leq \frac{13}{47}$, so at least one of the three attacks succeeds with $(\frac{13}{47} + 2\epsilon) \cdot n$ corruption, regardless of the relative ratios of A_1, B_1, A_2, B_2 . ◀

References

- 1 Rudolf Ahlswede, Christian Deppe, and Vladimir Lebedev. Non-binary error correcting codes with noiseless feedback, localized errors, or both. In *2006 IEEE International Symposium on Information Theory*, pages 2486–2487, 2006. doi:10.1109/ISIT.2006.262057.
- 2 Elwyn R. Berlekamp. Block coding with noiseless feedback. *Massachusetts Institute of Technology, USA*, 1964.
- 3 Elwyn R. Berlekamp. Block coding for the binary symmetric channel with noiseless, delayless feedback. *Error-correcting Codes*, pages 61–88, 1968.
- 4 Mark Braverman, Klim Efremenko, Gillat Kol, Raghuvansh Saxena, and Zhijun Zhang. Round-vs-resilience tradeoffs for binary feedback channels. *Electronic Colloquium on Computational Complexity*, TR22-179, December 2022.
- 5 Marat Burnashev and Hirosuke Yamamoto. On the zero-rate error exponent for a bsc with noisy feedback. *Problems of Information Transmission*, 44, September 2008. doi:10.1134/S0032946008030034.
- 6 Marat V. Burnashev and Hirosuke Yamamoto. On bsc, noisy feedback and three messages. In *2008 IEEE International Symposium on Information Theory*, pages 886–889, 2008. doi:10.1109/ISIT.2008.4595114.
- 7 Klim Efremenko, Gillat Kol, Raghuvansh Saxena, and Zhijun Zhang. Binary codes with resilience beyond 1/4 via interaction. *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS, 2022*. arXiv:TR22-129.
- 8 Meghal Gupta, Venkatesan Guruswami, and Rachel Yun Zhang. Binary error-correcting codes with minimal noiseless feedback. *To appear in STOC 2023*, 2022.
- 9 Meghal Gupta, Yael Tauman Kalai, and Rachel Yun Zhang. Interactive error correcting codes over binary erasure channels resilient to $> 1/2$ adversarial corruption. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 609–622, 2022.

- 10 Meghal Gupta and Rachel Yun Zhang. Positive rate binary interactive error correcting codes resilient to $> 1/2$ adversarial erasures. *arXiv preprint*, 2022. [arXiv:2201.11929](https://arxiv.org/abs/2201.11929).
- 11 Meghal Gupta and Rachel Yun Zhang. A new upper bound on the maximal error resilience of interactive error-correcting codes. *arXiv preprint*, 2023. [arXiv:2305.04376](https://arxiv.org/abs/2305.04376).
- 12 V. Guruswami. List decoding from erasures: bounds and code constructions. *IEEE Transactions on Information Theory*, 49(11):2826–2833, 2003. [doi:10.1109/TIT.2003.815776](https://doi.org/10.1109/TIT.2003.815776).
- 13 Venkatesan Guruswami and Madhu Sudan. List Decoding Algorithms for Certain Concatenated Codes. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, STOC '00*, pages 181–190, New York, NY, USA, 2000. Association for Computing Machinery. [doi:10.1145/335305.335327](https://doi.org/10.1145/335305.335327).
- 14 Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with Partial Noiseless Feedback. In *APPROX-RANDOM*, 2015.
- 15 R. W. Hamming. Error detecting and error correcting codes. *The Bell System Technical Journal*, 29(2):147–160, 1950. [doi:10.1002/j.1538-7305.1950.tb00463.x](https://doi.org/10.1002/j.1538-7305.1950.tb00463.x).
- 16 Claude E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. [doi:10.1002/j.1538-7305.1948.tb01338.x](https://doi.org/10.1002/j.1538-7305.1948.tb01338.x).
- 17 Joel Spencer and Peter Winkler. Three Thresholds for a Liar. *Combinatorics, Probability and Computing*, 1(1):81–93, 1992. [doi:10.1017/S0963548300000080](https://doi.org/10.1017/S0963548300000080).
- 18 Gang Wang, Yanyuan Qin, and Chengjuan Chang. Communication with partial noisy feedback. In *2017 IEEE Symposium on Computers and Communications (ISCC)*, pages 602–607, 2017. [doi:10.1109/ISCC.2017.8024594](https://doi.org/10.1109/ISCC.2017.8024594).
- 19 K.Sh. Zigangirov. Number of correctable errors for transmission over a binary symmetrical channel with feedback. *Problems Inform. Transmission*, 12:85–97, 1976.