Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 179 (2022)

Round-vs-Resilience Tradeoffs for Binary Feedback Channels

Mark Braverman*
Princeton University

Klim Efremenko†
Ben-Gurion University

Gillat Kol‡
Princeton University

Raghuvansh R. Saxena§
Tata Institute of Fundamental Research

Zhijun Zhang¶
Princeton University

Abstract

In a celebrated result from the 60's, Berlekamp showed that feedback can be used to increase the maximum fraction of adversarial noise that can be tolerated by binary error correcting codes from $\frac{1}{4}$ to $\frac{1}{3}$. However, his result relies on the assumption that feedback is "continuous", *i.e.*, after every utilization of the channel, the sender gets the symbol received by the receiver. While this assumption is natural in some settings, in other settings it may be unreasonable or too costly to maintain.

In this work, we initiate the study of *round-restricted feedback channels*, where the number r of feedback rounds is possibly much smaller than the number of utilizations of the channel. *Error correcting codes* for such channels are *protocols* where the sender can ask for feedback at most r times, and, upon a feedback request, it obtains all the symbols received since its last feedback request. We design such error correcting protocols for both the adversarial binary *erasure* channel and for the adversarial binary *corruption* (bit flip) channel. For the erasure channel, we give an exact characterization of the round-vs-resilience tradeoff by designing a (constant rate) protocol with r feedback rounds, for every r, and proving that its noise resilience is optimal.

Designing such error correcting protocols for the *corruption* channel is substantially more involved. We show that obtaining the optimal resilience, even with one feedback round ($r = 1$), requires settling (proving or disproving) a new, seemingly unrelated,

*mbraverm@gmail.com. Supported in part by the NSF Alan T. Waterman Award, Grant No. 1933331, a Packard Fellowship in Science and Engineering, and the Simons Collaboration on Algorithms and Geometry.

†klimefrem@gmail.com. Supported by the Israel Science Foundation (ISF) through grant No. 1456/18 and European Research Council Grant number: 949707.

‡gillat.kol@gmail.com. Supported by a National Science Foundation CAREER award CCF-1750443 and by a BSF grant No. 2018325.

§raghuvansh.saxena@gmail.com. Supported by the Department of Atomic Energy, Government of India, under project no. RTI4001.

¶zhijunz@princeton.edu. Supported by a National Science Foundation CAREER award CCF-1750443.

ISSN 1433-8092

"clean" *combinatorial conjecture*, about the maximum cut in weighted graphs versus the "imbalance" of an average cut. Specifically, we prove an upper bound on the optimal resilience (impossibility result), and show that the existence of a matching lower bound (a protocol) is *equivalent* to the correctness of our conjecture.

Contents

1 Introduction

Cybernetics. Consider the following two scenarios. Scenario one: a steersperson wishes to steer a longship to shore. She maintains a steady course in a changing environment (wind, waves, storms, currents, tides, *etc.*) by adjusting her steering in continual response to the effect it is observed as having. Scenario two: a teacher has a semester-worth of topics he wishes to teach to his class. He schedules exams throughout the semester to help him adapt his pace and determine what material should be repeated.

The above two scenarios are examples of *cybernetics*, a field that studies *self-regulating* processes. A core concept in cybernetics is *circular causality*, which is typically implemented using *feedback* mechanisms, where the observed outcomes of actions are taken as inputs for further actions. This is the case for, *e.g.*, spacecraft navigators, artificial limbs, and our bodies' regulation of hormone and blood sugar levels. The term Cybernetics[1] was coined in 1948 by the mathematician and philosopher Norbert Wiener for "the science of control and communication in the animal and the machine" [Wie48], following exchanges between numerous fields during the 1940s, including anthropology, mathematics, neuroscience, psychology, and engineering.

Feedback in information theory. Cybernetics grew alongside and built on Claude Shannon's information theory, that was developed to improve the transmission of information and introduced the notion of *error correcting codes*. Shannon was interested in knowing whether the existence of a "feedback link" in the channel, where after every utilization of the channel, the (possibly incorrect) symbol obtained by the receiver is also given to the sender, allows for better codes. A discouraging early result by Shannon showed that feedback does not improve the capacity of memoryless channels [Sha56]. It would be another decade or so before Berlekamp proves that feedback can, in fact, increase the maximum fraction of *adversarial* errors that can be tolerated. Specifically, Berlekamp showed that the *maximum noise resilience* of the (adversarial) binary channel increases from $\frac{1}{4}$ to $\frac{1}{3}$ given feedback [Ber64, Ber68] (also see [Zig76, SWS92, ADL06]).

A key property of the feedback channel exploited by Berlekamp's result, as well as by follow up work, is that it supports "*continuous*" feedback – after *every* communication round, the sender gets the symbol received by the receiver. This assumption is natural in some settings, *e.g.*, in scenario one, the steersperson continuously watches the ship's motion as she steers. However, this assumption may be unreasonable or too costly to maintain in other settings, *e.g.*, in scenario two, the teacher may not want to continuously quiz his students.

This work: round-restricted feedback. Motivated by such examples, in this work, we initiate the study of *round-restricted feedback channels*, where the number of feedback rounds is possibly much smaller than the number of communication rounds. Specifically, we wish

[1] Interestingly, Cybernetics comes from the Greek word "Kubernetes", which means steersperson.

to design protocols with optimal noise resilience that allow the sender (Alice) to transmit a message to the receiver (Bob), where during the execution of the protocol, the sender can ask for feedback at most r times. Upon such a request, the sender obtains all the bits received by the receiver from the last time feedback was solicited.

One can consider two models for scheduling the feedback rounds: the *adaptive* and the *non-adaptive* models. In the non-adaptive model, the sender decides ahead of time (before the protocol is run and before the input is known) when to schedule the r feedback rounds, while in the adaptive model, the timing of each feedback request may depend on the previously received feedback. In the second scenario, for example, the non-adaptive setting corresponds to scheduling all exams at the beginning of the semester, while the adaptive setting corresponds to scheduling the next exam after the previous one was given. While our techniques hold for both the adaptive and non-adaptive settings, we choose to present our results for the non-adaptive setting. See Section 1.3 and Section 2.1 for a discussion of the implication of our techniques for the adaptive setting.

We consider such message transmission protocols with r feedback rounds over both the (adversarial) binary *erasure channel*, that erases some of the sent bits (those bits are received as '$\perp$'), and over the (adversarial) binary *corruption channel*, that flips some of the sent bits. As was mentioned before, classical results in information theory show that with no feedback the maximum noise resilience of the binary corruption channel is $\frac{1}{4}$ [Plo60], while with continuous feedback, the maximum resilience improves to $\frac{1}{3}$ [Ber64, Ber68]. For the binary erasure channel, it is known that with no feedback the maximum resilience is $\frac{1}{2}$, and it is easy to see that with continuous feedback it approaches 1: the sender re-transmits each symbol until the receiver receives it.

We mention that rounds (or passes) are often considered to be a scarce resource and that round-restricted algorithms are extensively studied in other communication settings, *e.g.*, communication complexity, distributed computing, streaming algorithms, and cryptographic protocols, and that we draw inspiration from these settings.

1.1 Our Results and Conjecture

1.1.1 The (Adversarial) Binary Erasure Channel

As discussed above, the maximum resilience of the erasure channel is known for the extreme cases of no feedback and of continuous feedback. Our first result is an optimal *round-vs-resilience tradeoff* for the erasure channel with any number of non-adaptive feedback rounds.

Theorem 1.1. *The maximum noise resilience of the (adversarial) binary erasure channel with r rounds of feedback is $\frac{5}{7}$ if $r = 1$ and $1 - \frac{7}{12(r+1)}$ if $r > 1$. Furthermore, the maximum noise resilience can be obtained by a* deterministic, constant-rate *protocol.*

Theorem 1.1 can be viewed as a "hierarchy theorem", showing that more feedback rounds allow for strictly better resilience. On the other hand, Theorem 1.1 also shows that a constant number $O_\epsilon(1)$ of feedback rounds already suffices to get a noise resilience of $1-\epsilon$ for the erasure channel.

Techniques. The main ingredient in our proof of Theorem 1.1 is the construction of a *list decodable* code for the binary erasure channel with m codewords, for *all* (not necessarily asymptotic) values of m. Our code is optimal in the sense that it achieves the maximum error resilience for every list size simultaneously. We emphasize that for our protocols, we need such a code for all possible m, which corresponds to all possible "block sizes". We call codes with small m's "*small codes*". Given these codes, the protocols we use to prove Theorem 1.1 are rather simple – after every feedback round, Alice and Bob agree on a (smaller, unless there was a lot of noise) set Γ of candidate inputs x and Alice encodes x with our optimal list decodable code with $m = |\Gamma|$ codewords. On the analysis front, we are able to argue that, unless the adversary erases many of the sent bits, the size of the candidate set Γ shrinks substantially between feedback rounds, and measure this shrinkage exactly. See Section 2.1 for a detailed overview.

1.1.2 The (Adversarial) Binary Corruption Channel

Theorem 1.1 gives a complete characterization of the noise resilience of the erasure feedback channel as a function of the number of feedback rounds. However, as will be explained next, the case of corruptions is much more involved, and we will focus on protocols with one round of feedback. We mention that since the adaptive and non-adaptive models are the same for protocols with one feedback round, the results in this section hold for both the adaptive and non-adaptive settings. Our next theorem gives an upper bound on the noise resilience of such one-round protocols.

Theorem 1.2. *The maximum noise resilience of the (adversarial) binary corruption channel with one round of feedback is at most $\frac{7}{23}$.*

We conjecture that the upper bound of $\frac{7}{23}$ on the noise resilience in Theorem 1.2 is tight, and that it can be achieved by a constant-rate protocol. Perhaps surprisingly, proving this is *equivalent* to showing the following combinatorial conjecture about the existence of large cuts in graphs.

Conjecture 1.3. *Let G be a graph with n vertices and non-negative edge weights summing up to 1. Let $\mathsf{wt}(S)$ be the sum of weights of all the edges with both endpoints in the subset of vertices S, and let $\mathsf{Max\text{-}Cut}(G)$ be the maximum total weight of all the edges across any cut in G. Then,*[2]

$$\mathsf{Max\text{-}Cut}(G) \geq \frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [n]} \Big[\min\big(\mathsf{wt}(S), \mathsf{wt}(\overline{S})\big)\Big]. \tag{1}$$

[2]As the expectations of $\mathsf{wt}(S)$ and $\mathsf{wt}(\overline{S})$, for a uniformly random S, are $\frac{1}{4}$, Eq. (1) can be equivalently written as $\mathsf{Max\text{-}Cut}(G) \geq \frac{6}{15} + \frac{8}{15} \cdot \mathbb{E}_{S\subseteq[n]}\big[\big|\mathsf{wt}(S) - \mathsf{wt}(\overline{S})\big|\big]$, where the term inside the expectation is the "imbalance" of a random cut.

We prove Conjecture 1.3 for (large enough) graphs where all edges have equal weight, *i.e.*, "unweighted" graphs (see Section 11). However, the case for general weighted graphs seems much harder, and, despite our best effort, we were unable to prove (or disprove) it. We also mention that Conjecture 1.3 is tight for some graphs (*e.g.*, cliques of size 3 and 5 with edges of equal weight), and related bounds on Max-Cut were studied in other contexts, *e.g.*, [PT86, Alo02, GY21].

The next theorem gives the equivalence between Conjecture 1.3 and the tightness of Theorem 1.2.

Theorem 1.4. *Theorem 1.2 is tight if and only if Conjecture 1.3 holds. Furthermore, Conjecture 1.3 implies a* constant rate *protocol achieving the maximum noise resilience.*

In essence, Theorem 1.4 connects the problem of designing optimal error correcting protocols with one round of feedback to a combinatorial question about graphs. As we discuss later in Section 2.2, our techniques can also be used to connect the problem of designing optimal error correcting protocols with multiple rounds of feedback to similar questions about graphs.

Techniques. The proof of Theorem 1.4 is technically involved and a detailed overview can be found in Section 2.2. At a high level, the main ingredient in designing our protocol is the construction of a special type of "weighted" codes, called dc-*codes*. A dc-code C is parameterized by a "distance contribution function" dc that assigns a value in $[0, 1]$ to each possible message $x \in \{0, 1\}^k$. We require that for all $x \neq x' \in \{0, 1\}^k$, the codewords $C(x)$ and $C(x')$ are at least (relative) Hamming distance $\mathsf{dc}(x) + \mathsf{dc}(x')$ apart. Equivalently, we ask that the balls of radii $\mathsf{dc}(x)$ around $C(x)$ are all disjoint.[3] We note that unlike traditional error correcting codes that have only one distance guarantee for all pairs of codewords (*i.e.*, the minimum distance), the distance guarantees for different pairs of codewords in a dc-code are different. In fact, traditional codes can be viewed as dc-codes for a constant dc function.

dc-codes for non-constant dc functions are useful for our protocol as if the adversary already used up many of its corruptions before the feedback round, Alice knows she can afford to send her message x encoded with an error correcting code that does not guarantee a large distance between $C(x)$ and the other codewords. Geometrically, designing a dc-code is a sphere packing problem where we need to pack spheres of different radii $\mathsf{dc}(x)$. As for some x's a small radius $\mathsf{dc}(x)$ suffices, some of the spheres are small, which allows the other spheres being packed to be larger.

The proof of Theorem 1.4 shows that Conjecture 1.3 implies the existence of dc-codes that are needed for our protocol to work. We assume that Alice uses a uniformly random code to encode her message before the feedback. The codeword sent by Alice can be corrupted by the channel in many ways, and each such way would imply a function dc such that Alice would like to use a dc-code to encode her message after the feedback. We denote by Q the set of dc

[3]We mention that dc-codes are an example of non-equally spaced codes defined in [EKSZ22].

functions for which the corresponding dc-codes are needed by our protocol. We also denote by P the set of dc functions for which dc-codes exist. We wish to show $Q \subseteq P$. To this end, we show that both P and Q are closed and convex, and that in every direction z, the extremal point of P in direction z is "farther" than the extremal point of Q in direction z. We then recast this geometric problem as a combinatorial problem by interpreting the direction vector z as a weighted graph G, and show that the extremal point of P in direction z corresponds to a Max-Cut in G (as in the left hand side of Conjecture 1.3), while the extremal point of Q in direction z corresponds to the right hand side of Conjecture 1.3.

For the converse direction of Theorem 1.4, we show that the arguments in the above paragraph are actually equivalences, except for the assumption that Alice uses a randomly sampled code to encode her message before the feedback. At a high level, we use *Ramsey theory* to show that the assumption that this code is a random code is, at least in some sense, without loss of generality (see Section 2.2.2 for a more precise statement).

1.2 Related Work

Feedback channels were studied since the early days of information theory and are still actively studied [Sha56, Hor63, For68, Ber68, Bur76, Sah08, Sha09, ESSG10, SF11, SW13, to cite a few]. While feedback does not increase the capacity of discrete memoryless channels with vanishing error, there are settings where feedback is known to allow improvement, like in the 0-error capacity case [Sha56], and under variable decision time [Bur76].

Partial feedback. Haeupler, Kamath, and Velingker [HKV15] considered the setting where the feedback is partial, and showed that even if Alice receives feedback bits from Bob for an arbitrarily small constant fraction of her transmissions, resilience close to (the optimal resilience of) $\frac{1}{3}$ is possible using a randomized protocol. However, the number of feedback rounds their protocol needs grows linearly with n, the length of Alice's input. See [WQC17] for a subsequent result.

Independently and concurrently to our work, [GGZ23] improved [HKV15] and showed a deterministic protocol that uses $\mathcal{O}(\log n)$ feedback bits over $\mathcal{O}(1)$ feedback rounds to get resilience approaching $\frac{1}{3}$, along with a similar result for the erasure channel showing that the resilience approaches 1 for this channel. The main difference between [GGZ23] and the current work is that we focus on finding the optimal resilience for any given number r of feedback rounds whereas [GGZ23] focuses on showing that the resilience approaches the optimal value as the constant r increases. Additionally, their work measures both the number of feedback rounds and the number of feedback bits, while we only focus on the number of rounds.

Two-way codes and interactive codes. As discussed above, feedback is also known to increase the noise resilience of the adversarial binary corruption channel [Ber64, Ber68], and this result played a big role in recent work in *interactive coding* [EKS20, GZ22b, GZ22a] and *two-way coding* [GKZ22, GZ22c, EKSZ22]. In interactive coding [Sch92, Sch93, Sch96], we wish to simulate a communication protocol Π that was designed to work over the noiseless channel, by a protocol Π' that works over a noisy channel. In the setting of two-way codes, like in the setting of traditional error correcting codes, Alice wishes to transmit a message x to Bob over a noisy channel. However, unlike the case of traditional codes, where Alice is the only party that can transmit messages, in two-way codes Bob can also use the (noisy) channel to transmit messages back to Alice.

Observe that since Bob has no input, any two-way code can be run over the feedback channel and thus two-way error correcting codes can be viewed as protocols over a *noisy feedback* channel. In particular, since the noise tolerance of the binary corruption channel is only $\frac{1}{3}$, the noise resilience of binary two-way codes over the binary corruption channel is at most $\frac{1}{3}$. In the same way, results for the bounded round feedback channel give upper bounds on the noise resilience of the corresponding two-way channels.

Gupta, Kalai, and Zhang [GKZ22, GZ22c] studied two-way error correcting codes over the binary *erasure channel*. Their main result is a code that is resilient to a $\frac{3}{5}$ fraction of adversarial errors, improving on the noise tolerance of the one-way binary erasure channel that is known to be $\frac{1}{2}$. We mention that the two-way coding schemes of [GKZ22, GZ22c] exchange (almost) linear number of messages. The work of [GKZ22] also gives an upper bound of $\frac{2}{3}$ on the maximum tolerance of the two-way binary erasure channel, and an upper bound of $\frac{2}{7}$ on the maximum tolerance of the two-way binary corruption channel. Given those upper bounds, a corollary of our results is that even a single round of noiseless feedback allows for a better error tolerance than any number of noisy feedback rounds over both the erasure and corruption channels[4].

The recent work of Efremenko, Kol, Saxena, and Zhang [EKSZ22] shows that the maximum noise resilience of two-way error correcting codes for the binary corruption channel is strictly better than the noise resilience of traditional error correcting codes for this channel, which is known to be $\frac{1}{4}$ [Plo60]. At a very high level, those results for two-way codes are obtained by implementing a (weak) feedback mechanism over channels with no built-in feedback. Related ideas were used in [EKS20, GZ22b, GZ22a] to give interactive binary error correcting codes with high noise resilience.

[4]To see why, observe that if Bob's messages are noiseless, we can assume without loss of generality that Bob's messages are much shorter, say at most an ϵ fraction, of Alice's messages. Indeed, if not, consider a modified protocol where all messages from Alice are repeated k times, for some large k. For the erasure channel, either all the repetitions of a bit from Alice are erased or Bob knows the bit exactly. Thus, his communication does not grow with k. For the corruption channel, it suffices for Bob to say how many of the repetitions were received as 1, which can be done using $\log k \ll k$ bits.

Moreover, we mention that for this claim, we do not need to rely on Conjecture 1.3, as a lower bound slightly smaller than $\frac{7}{23}$ (but greater than $\frac{2}{7}$) on the maximum error resilience of protocols with one feedback round over the binary corruption channel can be obtained unconditionally using our techniques (but is not included in the current work).

List decodable codes. List decodable codes were introduced in the 50's [Eli57, Woz58] and have been studied over numerous papers and found many applications since then. We next list the works most related to ours. Most of the work on list decoding was done in the asymptotic regime, where the number of codewords goes to infinity. In this work, we are interested in the optimal list decodable codes for any (potentially small) number of codewords. However, as an ingredient in our proof, we use the asymptotic results of [ABP19] (see also [Bli86, GS00, Bli09]) for optimal list decoding of the corruption channel (see Lemma 4.5). The list decoding question was also considered for other channels, for example, over the corruption channel with feedback [Sha09] and the erasure channel [Gur03].

1.3 Open Problems

Our work suggests the study of feedback channels through a new lens, namely, their feedback round complexity. We next list some suggestions for future work in this direction.

Graph-theoretic conjectures. The most immediate question we leave open is proving Conjecture 1.3 for all weighted graphs. We also propose the following potentially related conjecture, which is tight for all odd cliques with edges of equal weight.

Conjecture 1.5. *Let G be a graph with n vertices and non-negative edge weights summing up to* 1*. Let* $\mathsf{wt}(i)$ *be the sum of weights of all edges incident on vertex i. Then,*

$$\textsf{Max-Cut}(G) \geq \frac{1}{2} + \frac{1}{8} \cdot \sum_{i \in [n]} \mathsf{wt}(i)^2.$$

Round-vs-resilience tradeoff for other channels. Proving Conjecture 1.3 would imply that our protocol in Theorem 1.4 has optimal noise resilience among protocols with one round of feedback over the corruption channel. Obtaining a general round-vs-resilience tradeoff for any number of feedback rounds r for the corruption channel and for other well-studied channels (*e.g.*, the binary insertion-deletion channel[5], the binary deletion-only channel, and non-binary channels), would be interesting.

Adaptive corruptions over the erasure channel. Theorem 1.1 considers the case of *non-adaptive* feedback rounds, where Alice decides ahead of time when to ask for feedback. It can be shown that the case of *adaptive* feedback rounds, where Alice chooses when to ask for another round of feedback after seeing the previous feedback, allows for (strictly) better round-vs-resilience tradeoffs. Our techniques can be used to write a recursive formula for the noise resilience in the adaptive case, and finding a "clean", closed-form formula for this setting (if one exists) is left open (see Section 2.1).

[5] We note that this first requires a suitable definition for the insertion-deletion channel with constant number of rounds.

Acknowledgements

We thank Noga Alon for discussions regarding Conjecture 1.3.

2 Proof Overview

In this section, we overview the proofs of Theorems 1.1 and 1.4, starting with the relatively easier Theorem 1.1.

2.1 Result for the Erasure Channel – Theorem 1.1

The defining feature of the erasure channel is that the receiver (Bob) either receives the bit sent by Alice or receives a special erasure symbol $\perp$. This means that in any round where Bob receives $\perp$, he is certain that this is due to the erasures in the channel, while if he receives a symbol different from $\perp$, he is certain that the symbol must be what Alice sent in that round. In turn, this means that Bob knows exactly the amount of erasures introduced by the channel and also means that Bob can (recall that he is trying to determine Alice's input) remove from consideration any candidate input that is "inconsistent" and would make Alice send a different symbol in any such round.

The general format of a protocol. The above observation implies that protocols for the erasure channel with r rounds of feedback (and therefore $r+1$ messages from Alice) have the following format: Alice starts with an input $x \in \Gamma_0 = \{0,1\}^n$. For her first message, she takes a code[6] $C_0 : \Gamma_0 \to \{0,1\}^*$ and sends $C_0(x)$ to Bob. Some of the bits of $C_0(x)$ are received correctly by Bob, while the remaining bits are erased and replaced with $\perp$. Using the bits he received correctly, Bob can calculate the number of erasures N_1 introduced by the channel in this round and can identify a subset $\Gamma_1 \subseteq \Gamma_0$ of inputs for Alice that are consistent with the message he received. Note that Alice's input x must be in Γ_1.

Then, a feedback round takes place, and as Alice learns all the received symbols, she can also compute N_1 and Γ_1. As both parties now know these values, they can now "forget" this round and "reduce"[7] to a smaller problem where Alice wants to transmit an element $x \in \Gamma_1$ to Bob using a protocol with $r-1$ rounds of feedback and the maximum number of erasures the channel can insert is N_1 lower than what it was before. Continuing this way, the goal of the parties is to reduce to a problem with 0 rounds of feedback, and set of inputs Γ_r such that there exists a (standard) error correcting code for elements in Γ_r resilient to the number of erasures that the channel can insert in the last round.

[6]At this point, it may be helpful to view this as a function instead of a code. We explain why we are calling it a code later. Also, a more precise way to state this would be to say that there exists an $L > 0$ such that $C_0 : \Gamma_0 \to \{0,1\}^L$, as all codewords need to be of the same length to avoid the parties from signaling through the length of the codeword. Nonetheless, we stick with statements like $C_0 : \Gamma_0 \to \{0,1\}^*$ throughout this sketch for simplicity.

[7]We elaborate what this means exactly in the paragraph on adaptive feedback rounds below.

List-decodable small codes. It is readily seen that the protocol format described above does not care about the exact strings in the sets $\Gamma_0, \ldots, \Gamma_r$, as long as their sizes stay the same. Thus, the question of whether or not the above protocol format can be instantiated to get a protocol that is resilient to θ fraction of adversarial erasures, for some $\theta \in [0,1]$, reduces to determining when to schedule the feedback rounds, and given two feedback rounds, determining the codes C_i to be used by Alice between these rounds. The codes C_i should be such that, given an initial set size $m = |\Gamma_i|$ and a target set size[8] $k = |\Gamma_{i+1}|$, the number of erasures required to reduce the set size from m to k is the highest. Using such codes, Alice ensures that unless the adversary invests many erasures, the set of candidates shrinks substantially between feedback rounds. We first focus on designing such codes.

Codes like the above are known as list decodable codes, and have been well studied in the asymptotic regime, where m tends to infinity, and exact answers are known (see, *e.g.*, [Eli57, Woz58, Bli86, Gur03, Bli09, ABP19, Sha09] and Lemma 4.5). However, for our purposes, we need the exact answer for smaller values of m as well. Codes with small m, *i.e.*, "small codes" or codes with few codewords, have recently received a lot of attention and have proven to be useful in designing binary protocols with high error resilience in several contexts [EKS20, EKSZ22, GKZ22, GZ22b, GZ22c]. In the current paper, we provide a complete analysis of the list-decodability of these codes for the erasure channel, giving a function $\mathsf{d}(m,k)$ that characterizes exactly the minimum amount of erasure noise needed such that for *any* code $C : [m] \to \{0,1\}^*$, one can erase $\mathsf{d}(m,k)$ fraction of the bits and ensure that Bob gets a list of candidates of length strictly smaller than k.

The formula for $\mathsf{d}(m,k)$ is given in Eq. (7). Proving that this formula is correct requires showing both a construction (of codes with resilience approaching $\mathsf{d}(m,k)$) and an impossibility result. Our construction has the nice property that the same code is tight simultaneously for all values of k. Roughly speaking, our code achieves this optimal erasure noise resilience by ensuring that every coordinate is as differentiating as possible, *i.e.*, we ensure that for all coordinates j, exactly $\left\lfloor \frac{m}{2} \right\rfloor$ (uniformly chosen) codewords have 0 in that coordinate, while the remaining $\left\lceil \frac{m}{2} \right\rceil$ codewords have 1 (see Lemmas 4.3 and 4.4). This is as opposed to randomly sampled codes where, *e.g.*, a $\frac{1}{2^m}$ fraction (which is large for small m) of the coordinates are expected to be 0 for all the codewords, and therefore not differentiate between any pair of codewords.

Scheduling the feedback rounds. Even with an exact formula for $\mathsf{d}(m,k)$ in hand, it still remains to schedule the feedback round correctly in order to maximize the overall noise resilience of the obtained protocol. The fact that our constructed code is tight simultaneously for all values of k is of great help for this part, as the actual value of k is determined by the erasures inserted by the channel and not in our control. This means that in order to schedule the feedback rounds optimally, one needs to go over all possible values of k (across all rounds)

[8] Note that k is not known to the parties in advance, and thus it will be ideal if the code used is optimal for all k simultaneously.

that may happen over the channel and maximize the corresponding error resilience. This requires a careful analysis of the obtained formula for $\mathsf{d}(m, k)$ and is presented partly in the main body of this paper and partly in Appendix A.3.

Adaptive feedback rounds. We finish this section by briefly discussing the extension of our result to adaptive feedback rounds, as hinted in Section 1.3. Recall our reduction above from r to $r-1$ feedback rounds, and note that this reduction is not perfect in the following sense: the erasures inserted by the adversary in Alice's first message in the r-round protocol dictate the set Γ_1 of candidates and the budget of the $(r-1)$-round protocol. Observe that the $(r-1)$-round protocol with maximal noise resilience for transmitting a message depends on the size of the set of candidates and on the erasure budget. Now, since our r-round protocol is non-adaptive, meaning that the timing of all feedback rounds is fixed in advance and cannot be recalculated given the erasures in the first round, our r-round protocol may reduce to a sub-optimal $(r-1)$-round protocol. Therefore, when scheduling the feedback rounds for our r-round protocol, one needs to consider the values of k that are possible across all rounds in order to get the optimal schedule.

On the other hand, if the feedback rounds can be scheduled adaptively, the reduction is indeed perfect. In this case, one just needs to schedule the first feedback round beforehand based on the possible values of $k = |\Gamma_1|$ for this round alone, and then, upon seeing the N_1 and Γ_1 values, one can take the $(r-1)$-feedback round protocol with the maximum error resilience (when Alice's input is from Γ_1 and the number of erasures is N_1 lower) and schedule the remaining feedback rounds according to this protocol. Thus, our techniques also lead to a tight recursive formula for the maximum error resilience in the case of adaptive feedback rounds, but converting it to a "clean" closed form formula (if at all possible) is left open.

2.2 Result for the Corruption Channel – Theorem 1.4

Compared to the erasure channel, where Bob knows exactly the amount of noise inserted and can safely eliminate many candidate inputs for Alice, the corruption channel is much harder. Here, upon receiving a message from Alice, all Bob can compute is, given a candidate input y for Alice, what is the number $\mathsf{N}(y)$ of corruptions the channel inserted assuming Alice's input was indeed y. Crucially, this value of $\mathsf{N}(y)$ may be very different for different y, and unless it exceeds the maximum possible number of corruptions in the channel (which can only happen when the protocol is quite far advanced), it can never have Bob eliminate y from consideration entirely.

Consider now a protocol over the corruption channel with one round of feedback (and therefore, two messages from Alice). Suppose that Alice's input x comes from the set $\{0,1\}^n$. As explained above, after receiving the first message from Alice, Bob knows $\mathsf{N}(y)$ for all $y \in \{0,1\}^n$. By subtracting $\mathsf{N}(y)$ from the maximum possible number of corruptions, Bob can compute, for all $y \in \{0,1\}^n$, a number $\mathsf{dc}(y)$ which is the leftover corruptions, or, equivalently, the degree to which the second message of Alice can be corrupted, assuming her input is y. As Alice receives feedback from Bob, she can also compute the values $\mathsf{dc}(y)$ for all $y \in \{0,1\}^n$. In the remainder of this sketch, we normalize $\mathsf{dc}(y)$ by dividing it by the length of Alice's second message. This will result in a value in $[0,1]$.

dc-codes. Using this feedback, Alice's goal in her second message is to allow Bob to uniquely identify her input. If $C : \{0,1\}^n \to \{0,1\}^*$ is the code used by Alice in her second message, the only way Bob can uniquely decode Alice's input is if for all $y \neq y' \in \{0,1\}^n$, the codewords $C(y)$ and $C(y')$ are at least (relative) Hamming distance $\mathsf{dc}(y) + \mathsf{dc}(y')$ apart. The reason is that if y is Alice's input, then the adversary has fractional budget $\mathsf{dc}(y)$ that it can use to corrupt $C(y)$, and thus the codeword received by Bob can be any string of (relative) Hamming distance at most $\mathsf{dc}(y)$ from $C(y)$. Similarly, if y' is Alice's input, then the codeword received by Bob can be any string of Hamming distance at most $\mathsf{dc}(y')$ from $C(y')$. Note that the adversary cannot arrange for the received encodings to be the same if and only if $C(y)$ and $C(y')$ are at least (relative) Hamming distance $\mathsf{dc}(y) + \mathsf{dc}(y')$ apart. We call a code that satisfies this (relative) Hamming distance property a dc-code and mention that the values $\mathsf{dc}(y)$ can equivalently be seen as the "distance contributed" by y in such a code.

We note that unlike traditional error correcting codes that have only one distance guarantee for all pairs of codewords (*i.e.*, the minimum distance), for dc-codes, the distance between a pair of codewords may be different depending on the "compatibility" of the messages they encode. Specifically, we think of each codeword as having a different "radius" and the code needs to "pack" all the induced balls of different radii. We point out that dc-codes are an example of non-equally spaced codes defined in [EKSZ22].

We also observe that the small code used in our protocol for erasures can be viewed as a dc-code where $\mathsf{dc}(y) = 0$ for all inputs y that Bob has ruled out (and therefore, do not need any distance guarantees), and $\mathsf{dc}(y) = c$ for all inputs y that he has not ruled out, where c is the best possible constant (c is determined by the $\mathsf{d}(m,k)$ function). We mention that for the erasure channel, our protocol also needed list-decoding guarantees that are not needed here as we are only attempting to get a one feedback round protocol.

The discussion so far shows that the existence of a protocol with a given error resilience amounts to determining whether or not it holds that for all functions $\mathsf{dc}(\cdot)$ that can be induced by the corruptions inserted in Alice's first message, there exists a dc-code that Alice can use to compute her second message. Curiously, we show in the next subsection that this question is equivalent to our seemingly unrelated combinatorial conjecture (Conjecture 1.3) about the existence of large cuts in graphs.

Towards multiple rounds of feedback. The above approach of designing dc-codes (that have no rounds of feedback) to construct protocols with one round of feedback can be generalized. One can similarly argue that, for any $r \geq 0$, dc-codes with r rounds of feedback can be used to construct protocols with $r+1$ rounds of feedback. Analogously to the above, the "extra" round is the first round, and dictates which dc-code is used in the rest of the protocol. Moreover, questions about constructing dc-codes with r rounds of feedback can be translated to questions about graphs. The $r = 0$ case is explained next, but similar ideas may be used for general r, with appropriate changes in the definitions of the set P and Q (see below).

2.2.1 Conjecture 1.3 Implies a Tight Protocol

We first show why Conjecture 1.3 implies the existence of a tight protocol. In fact, we shall show the existence of a protocol where Alice's message in the first round is simply the encoding of her input x using a randomly sampled code. Let $m = 2^n$. A distance function is a function $\mathsf{dist} : \binom{[m]}{2} \to \mathbb{R}$, where $\binom{[m]}{2}$ is the set of all subsets of $[m]$ of size 2. For a code $C : [m] \to \{0,1\}^*$, we denote by dist_C the distance function induced by C, *i.e.*, $\mathsf{dist}_C(i,i')$ is the (relative) Hamming distance between $C(i)$ and $C(i')$. For a distance contribution function dc, we denote by $\mathsf{dist}_{\mathsf{dc}}$ the distance function induced by dc, *i.e.*, $\mathsf{dist}_{\mathsf{dc}}(i,i') = \mathsf{dc}(i) + \mathsf{dc}(i')$. For simplicity, throughout this overview we assume that $\mathsf{dc}(y) = 1 - \mathsf{N}(y)$ (recall that $\mathsf{dc}(y)$ is actually the normalized leftover corruption count, but in this sketch we will ignore the exact multiplicative and additive constants in this function).

Recasting as a geometric problem. We denote by P the set of all distance functions dist_C that are induced by codes $C : [m] \to \{0,1\}^*$. We denote by Q the set of all distance functions $\mathsf{dist}_{\mathsf{dc}}$ induced by dc functions that can be obtained by the corruptions inserted in Alice's first message (recall that dc depends on N, which is a function of the corruptions inserted in Alice's first message). In other words, P is the set of distance functions that can be realized and Q is the set of distance functions required by our protocol. We wish to prove $Q \subseteq P$.

We view distance functions dist as $\binom{m}{2}$-dimensional vectors. We observe that both P and Q are closed and convex and that the set P is "downwards-closed", meaning that if $\mathsf{dist} \in P$ then any dist' that is coordinate wise at most dist is also in P. This means that showing $Q \subseteq P$ is equivalent to showing that for all $\binom{m}{2}$-dimensional non-negative hyperplanes z, it holds that:

$$\max_{\mathsf{dist} \in P} \langle z, \mathsf{dist} \rangle \geq \max_{\mathsf{dist} \in Q} \langle z, \mathsf{dist} \rangle, \tag{2}$$

Recasting as a combinatorial problem. By scaling, we can assume that the entries of z sum to 1 and view them as the weights on the edges of an m-vertex graph G_z as in Conjecture 1.3. As both P and Q are closed and convex, both the maximums are attained at one of their vertices.

To reason about Eq. (2), it will be useful to represent a code $C : [m] \to \{0,1\}^L$ as a sequence of L one-bit functions $b : [m] \to \{0,1\}$ (the first one-bit function corresponds to the first coordinate of $C(i)$, *etc.*). Observe that for the code $b : [m] \to \{0,1\}$ (*i.e.*, $L = 1$), it holds that dist_b is a boolean function with $\mathsf{dist}_b(i,i') = 1$ if and only if $b(i) \neq b(i')$.

The LHS of Eq. (2). Since a general code C is a sequence of one-bit functions, it can be shown that the function dist_C is a convex combination of the functions dist_b that are induced by one-bit functions b. In particular, this means that the vertices of P are distance functions induced by one-bit functions. Using the expression above for dist_b for one-bit function $b : [m] \to \{0,1\}$, the value of $\langle z, \mathsf{dist}_b \rangle$ is the value of the cut in the graph G_z indicated by b:

$$\langle z, \mathsf{dist}_b \rangle = \sum_{(i,i')} z_{i,i'} \cdot \mathsf{dist}_b(i,i') = \sum_{(i,i') \colon b(i) \neq b(i')} z_{i,i'}. \tag{3}$$

Thus, the left hand side of Eq. (2) is the maximum cut in G_z, as in Conjecture 1.3.

The RHS of Eq. (2). We view the code used by Alice in her first message as a sequence of one-bit functions. Since in our protocol this code is randomly sampled, each of the 2^m one-bit functions is expected to appear equally often in Alice's message[9]. As the channel can corrupt each of these one-bit functions independently of all the others, we get that a distance function dist can be induced by the corruptions inserted in Alice's first message (*i.e.*, $\mathsf{dist} \in Q$) if and only if it is the expectation (under the uniform distribution over one-bit functions) of the distance functions that can be induced by corrupting one-bit functions.

Now, if Alice is sending a one-bit function $b : [m] \to \{0,1\}$, there are only two possibilities for Bob: either he receives a 0 or he receives a 1. Let dc_b be the distance contribution function dictated by Bob's received bit. We next show that in the former case, where Bob receives 0, the value of $\langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle$ is the value of the cut in G_z indicated by b plus twice the weight of all edges such that $b(\cdot) = 0$ on both its endpoints. To see that, recall that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = \mathsf{dc}_b(i) + \mathsf{dc}_b(i')$ and that we assume $\mathsf{dc}_b(y) = 1 - \mathsf{N}(y)$. In our case, $\mathsf{dc}_b(i) = 1 - 0 = 1$ if $b(i) = 0$ (Alice's bit was not corrupted) and $\mathsf{dc}_b(i) = 0$ if $b(i) = 1$ (Alice's bit was corrupted). This implies that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 0$ if $b(i) = b(i') = 1$, and that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 1$ if $b(i) \neq b(i')$, and that $\mathsf{dist}_{\mathsf{dc}_b}(i,i') = 2$ if $b(i) = b(i') = 0$. Therefore,

$$\max_{\mathsf{dist} \in Q} \langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle = \sum_{(i,i')} z_{i,i'} \cdot \mathsf{dist}_{\mathsf{dc}_b}(i,i') = \sum_{(i,i') \colon b(i) \neq b(i')} z_{i,i'} + \sum_{(i,i') \colon b(i) = b(i') = 0} 2 \cdot z_{i,i'}. \tag{4}$$

Similarly, it can be shown that in the latter case, where Bob gets 1, the value of $\langle z, \mathsf{dist}_{\mathsf{dc}_b} \rangle$ is the value of the cut in G_z indicated by b plus twice the weight of all edges such that $b(\cdot) = 1$ on both its endpoints.

Recall that the bit function b is a uniformly random bit-function. Taking an expectation over one bit functions b, the value of the cut in G_z indicated by b is exactly the constant $\frac{1}{2}$ and the other terms on the right hand side of Eq. (3) and Eq. (4) are exactly as on the right hand side of Conjecture 1.3, where the maximum becomes minimum because of the constants involved. Eq. (2) now directly follows from Conjecture 1.3.

[9] We mention that this is only in expectation and the length of Alice's message need not depend exponentially on m.

2.2.2 A Tight Protocol Implies Conjecture 1.3

We now finish this sketch by arguing why a tight protocol implies Conjecture 1.3. For this, we note that all the arguments in Section 2.2.1 were actually equivalences, except two, one of which was explicitly stated and one was not. The explicit one was our assumption that Alice's first message is simply the encoding of her input using a randomly sampled code. The second one was that Alice gets feedback from Bob at round $\frac{8T}{23}$, where T is the total number of rounds of the protocol. The constant $\frac{8}{23}$ may seem arbitrary, but it is the constant one gets when one tries to match the constants obtained in the analysis in Section 2.2.1 with the constants in Conjecture 1.3.

Both these assumptions are actually without loss of generality. We start by arguing this for the second one, again ignoring the actual constants and only stating the high level idea. Roughly speaking, the second assumption is without loss of generality as Conjecture 1.3 is tight for cliques of size 3 and 5, and if the constant is anything other than $\frac{8}{23}$, Eq. (2) will fail to hold for z corresponding to one of these cliques. For a formal proof, see Claim 10.4.

It remains to show why the first assumption is without loss of generality. For this, our approach is to take an arbitrary code $C : [m] \to \{0,1\}^*$ that Alice may use for her first message, and in several steps, convert it to a code that looks more and more like a random code, at the cost of a smaller m. In each step k, we convert C to a code that is k-random, in the sense that any set of k codewords of the new code looks like k codewords from a randomly sampled code. The exact definition also requires an error parameter ϵ and is given in Definition 9.2.

For $k = 1$, this means that we have to show that each codeword has an equal number of 0s and 1s, and this can be easily achieved by concatenating all codewords with their negations (which preserves the distance properties). We now show how to get a 2-random code from a 1-random code, noting that similar (but technically more involved) ideas allow us to get a $(k+1)$-random code from a k-random code, for any $k \geq 1$. To show that a code is 2-random, we need to show that it is 1-random and that the fractional distance between any pair of codewords is (roughly) $\frac{1}{2}$.

For this, let $\epsilon > 0$ be an error parameter and construct a complete graph with the m codewords as the vertices, and color the edge between codewords i and i' as (1) red, if the fractional distance between them is smaller than $\frac{1}{2} - \epsilon$, (2) blue, if the fractional distance between them is between $\frac{1}{2} - \epsilon$ and $\frac{1}{2} + \epsilon$, (3) green, if the fractional distance between them is larger than $\frac{1}{2} + \epsilon$. As m gets larger and larger, Ramsey theory tells us that there must exist a large (going to infinity with m) monochromatic clique in this graph. This clique cannot be red, as we show that a large number of pairwise close codewords can be used to break the protocol. It also cannot be green, as that would violate known distance bounds for error correcting codes. Thus, it must be blue, implying that restricting attention to this clique gives us our desired 2-random code.

3 Model and Preliminaries

3.1 Notation and Preliminaries

For $x \in \mathbb{R}$, let $\overrightarrow{x}$ be the vector (of appropriate dimension inferred from context) with all its coordinates being x. Throughtout, all inequalities between vectors are coordinate-wise. For $k \geq 0$, $\Delta^k = \{(x_0, \dots, x_k) \in \mathbb{R}^{k+1} \mid \sum_{i=0}^{k} x_i = 1 \text{ and } x_i \geq 0 \text{ for all } i \in [0,k]\}$ denotes the k-dimensional standard simplex. For $x \in \mathbb{R}$ and $k \geq 0$, we write $x^{\underline{k}}$ as a shorthand for falling factorial $\prod_{i=0}^{k-1}(x-i)$.[10] For a set S and $k \geq 0$, let $\binom{S}{k}$ be the collection of all subsets of S of size k. For a function $f : X \to Y$ and subset $X' \subseteq X$, $f|_{X'}$ denotes the restriction of f onto X'. For $x, y \geq 1$, $\mathsf{R}(x, y)$ is the (two-color) Ramsey number for x, y, which is well-known to be finite. For $k \geq 1$ and two bit strings $x, y \in \{0,1\}^k$, their Hamming distance is $\Delta(x, y) = \sum_{i=1}^{k} \mathbb{1}[x_i \neq y_i]$.

3.2 Our Model: Round-Restricted Binary Feedback Channels

We now define (deterministic, binary) *protocols* with (non-adaptive) round-restricted feedback for the message transfer task, where Alice has an input and Bob's goal is to learn this input. Such a protocol is defined by a tuple:

$$\Pi = \Big(n, r, \{L_i\}_{i\in[r+1]}, \{f_i\}_{i\in[r+1]}, \mathsf{out}\Big), \tag{5}$$

where (1) $\{0,1\}^n$ is the set of all possible inputs for Alice. (2) r is the number of feedback rounds. Equivalently, we can say that Alice speaks in $r+1$ rounds. (3) For all $i \in [r+1]$, L_i is the length of Alice's message in the i-th round. Throughout, we use $L = \sum_{i=1}^{r+1} L_i$. (4) For all $i \in [r+1]$, $f_i : \{0,1\}^n \times \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{i-1}} \to \{0,1\}^{L_i}$ is the message function Alice uses in the i-th round. (5) $\mathsf{out} : \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{r+1}} \to \{0,1\}^n$ is the function Bob uses to compute the output.

Execution of a protocol. Let Π be a protocol as above. An adversary for Π is defined by a function $\mathsf{Adv} : \{0,1\}^n \to \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{r+1}}$. For $i \in [r+1]$, we will use $\mathsf{Adv}_i(\cdot)$ to denote the function that outputs the i-th coordinate of $\mathsf{Adv}(\cdot)$. We next define an execution of Π in the presence of an adversary Adv for Π: At the beginning of the execution, Alice starts with an input $x \in \{0,1\}^n$. The execution consists of $r+1$ rounds and before the i-th round, for $i \in [r+1]$, Alice and Bob have the (same) transcript $\tau_{<i} \in \{0,1,\perp\}^{L_1} \times \cdots \times \{0,1,\perp\}^{L_{i-1}}$. In round i, Alice computes the message $f_i(x, \tau_{<i}) \in \{0,1,\perp\}^{L_i}$ and sends it to Bob bit by bit, while Bob receives the string $\tau_i = \mathsf{Adv}_i(x)$. As we assume a feedback channel, if $i \leq r$, Alice also receives the string τ_i and both the parties add τ_i to $\tau_{<i}$ and continue executing the protocol.

[10] See also Falling factorials (Wikipedia) for the notation.

If $i = r+1$, the execution of the protocol terminates and Bob outputs $\mathsf{out}(\tau_{\leq r+1})$. Observe that this execution is completely determined by x, Π, and Adv. We denote the output of Π on input x in the presence of adversary Adv by $\mathsf{out}_{\Pi,\mathsf{Adv}}(x)$.

Counting the noise. Let Π be a protocol as above and Adv be an adversary for Π. For $x \in \{0,1\}^n$, the amount of noise added by Adv in Π on input x is the number of times Bobs' received bit is different from the bit Alice sent. Formally, we have:

$$\mathsf{noise}_{\Pi,\mathsf{Adv}}(x) = \sum_{i=1}^{r+1} \Delta(\mathsf{Adv}_i(x), f_i(x, \mathsf{Adv}_{<i}(x))). \tag{6}$$

For $\theta \in [0,1]$, we say that an adversary Adv has budget θ if we have

$$\max_{x \in \{0,1\}^n} \mathsf{noise}_{\Pi,\mathsf{Adv}}(x) \leq \theta L.$$

Types of Adversaries. Let Π be a protocol as above and Adv be an adversary for Π. We say that Adv is a *corruption adversary* if it never outputs the symbol $\perp$, *i.e.*, for all $x \in \{0,1\}^n$ and all $i \in [r+1]$, we have $\mathsf{Adv}_i(x) \in \{0,1\}^{L_i}$. We say that Adv is an *erasure adversary* if it only "erases" the symbols sent by Alice. More precisely, we say that Adv is an erasure adversary if for all $x \in \{0,1\}^n$, all $i \in [r+1]$, and all $j \in [L_i]$, if $(\mathsf{Adv}_i(x))_j \neq \perp$, then we have $(\mathsf{Adv}_i(x))_j = (f_i(x, \mathsf{Adv}_{<i}(x)))_j$.

Resilience of a protocol. Let Π be a protocol as above and $\theta \in [0,1]$. We say that Π *has resilience* θ over the binary erasure channel if for all erasure adversaries with budget θ and all $x \in \{0,1\}^n$, it holds that $\mathsf{out}_{\Pi,\mathsf{Adv}}(x) = x$. Resilience over the binary corruption channel is defined analogously.

4 Optimal List-Decodable Small Codes

In this section, we construct the codes used by our protocol.

4.1 Definitions of List Decodability

Codes for erasures. We start by defining list decodability for erasures.

Definition 4.1. *Let $m, k, L \geq 1$ and $d \in [0,1]$. We say that a code $C : [m] \to \{0,1\}^L$ is less-than-k-list decodable for erasures up to radius d if for all subsets $\Gamma \in \binom{[m]}{k}$, we have $\mathsf{ns}_C(\Gamma) > d$, where:*

$$\mathsf{ns}_C(\Gamma) = 1 - \frac{1}{L} \cdot \sum_{j=1}^{L} \mathbb{1}[\exists b \in \{0,1\}\ \forall i \in \Gamma : C_j(i) = b].$$

To get the intuition behind the definition of ns, observe that $\mathsf{ns}_C(\Gamma)$ is the minimum fraction e of erasures for which there exists $\tau \in \{0,1,\perp\}^L$ such that for all $i \in \Gamma$, it is possible to erase $e \cdot L$ symbols from $C(i)$ and get τ. Observe that this is equal to the fraction of coordinates where the encodings $\{C(i)\}_{i\in\Gamma}$ are *not* all the same (ns = not same).

For $m, k \geq 1$, we define $\mathsf{d}_{\mathsf{erase}}(m,k)$ to be the supremum of all values $d \in [0,1]$ for which there exists $L \geq 1$ and a code $C : [m] \to \{0,1\}^L$ that is less-than-k-list decodable for erasures up to radius d.

Codes for corruptions. Next, we define list decodability for corruptions:

Definition 4.2. *Let $m, k, L \geq 1$ and $d \in [0,1]$. We say that a code $C : [m] \to \{0,1\}^L$ is less-than-k-list decodable for corruptions up to radius d if for all $\tilde{x} \in \{0,1\}^L$, we have*

$$|\{i \in [m] : \Delta(C(i), \tilde{x}) < dL\}| < k.$$

Analogous to $\mathsf{d}_{\mathsf{erase}}$, for $m, k \geq 1$, we define $\mathsf{d}_{\mathsf{corr}}(m,k)$ to be the supremum of all values $d \in [0,1]$ for which there exists $L \geq 1$ and a code $C : [m] \to \{0,1\}^L$ that is less-than-k-list decodable for corruptions up to radius d.

4.2 Lemmas about $\mathsf{d}_{\mathsf{erase}}$ and $\mathsf{d}_{\mathsf{corr}}$

In this section, we show the results we need about $\mathsf{d}_{\mathsf{erase}}$ and $\mathsf{d}_{\mathsf{corr}}$. First, we define a helper function $\mathsf{d}(\cdot,\cdot)$:

$$\mathsf{d}(m,k) = 1 - \frac{\binom{\lfloor m/2 \rfloor}{k} + \binom{\lceil m/2 \rceil}{k}}{\binom{m}{k}}. \tag{7}$$

In Appendix A, we show useful properties about the function $\mathsf{d}(\cdot,\cdot)$.

4.2.1 Lemmas about $\mathsf{d}_{\mathsf{erase}}$

We now show that the functions $\mathsf{d}_{\mathsf{erase}}$ and d are the exact same. Owing to this lemma, we omit writing erase in the subscript in the rest of this text.

Lemma 4.3. *For all $m, k \geq 1$, we have:*

$$\mathsf{d}_{\mathsf{erase}}(m,k) = \mathsf{d}(m,k).$$

Proof. We first show that $\mathsf{d}_{\mathsf{erase}}(m,k) \leq \mathsf{d}(m,k)$. For this it suffices to show that for all $L \geq 1$ codes $C : [m] \to \{0,1\}^L$, there exists a subset $\Gamma \in \binom{[m]}{k}$ such that $\mathsf{ns}_C(\Gamma) \leq \mathsf{d}(m,k)$. We show such a subset Γ exists using probabilistic method. For $j \in [L]$, denote $z_{j,b} = |\{i \in [m] \mid C_j(i) = b\}|$ for $b \in \{0,1\}$, *i.e.*, the number of codewords with its j-th bit of encoding being b. Note that $z_{j,0} + z_{j,1} = m$ holds for all $j \in [L]$. By linearity of expectation, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\Gamma \in \binom{[m]}{k}} [\mathsf{ns}_C(\Gamma)] &= 1 - \frac{1}{L} \sum_{j=1}^{L} \Pr_{\Gamma \in \binom{[m]}{k}} (\exists b \in \{0,1\} \forall i \in \Gamma : C_j(i) = b) \\
&= 1 - \frac{1}{L} \sum_{j=1}^{L} \frac{\binom{z_{j,0}}{k} + \binom{z_{j,1}}{k}}{\binom{m}{k}} \\
&\le 1 - \frac{1}{L} \sum_{j=1}^{L} \frac{\binom{\lfloor m/2 \rfloor}{k} + \binom{\lceil m/2 \rceil}{k}}{\binom{m}{k}} \\
&= \mathsf{d}(m,k),
\end{aligned}$$

where in the second-to-last step we use the fact that $\binom{x}{k} + \binom{m-x}{k}$ is decreasing in x when $x \le \frac{m}{2}$ and is increasing in x when $x \ge \frac{m}{2}$, for all fixed m, k. The lemma follows by picking the subset $\Gamma \in \binom{[m]}{k}$ that minimizes the value of $\mathsf{ns}_C(\Gamma)$.

It remains to show that $\mathsf{d}_{\mathsf{erase}}(m,k) \ge \mathsf{d}(m,k)$. We show this by showing the stronger lemma Lemma 4.4 below. It is stronger as it shows the existence of codes that are constant rate and are tight for all values of k *simultaneously*.

□

Lemma 4.4. *For all $\epsilon > 0$, there exists a constant K such that for all $K' \ge K$ and for all $m \ge 1$, there exists a code $C : [m] \to \{0,1\}^{K' \log m}$ such that for all $k \in [m]$, the code C is less-than-k-list decodable up to radius $\mathsf{d}(m,k) - \epsilon$.*

Proof. Set $K = \frac{10}{\epsilon^3}$. Let $L = K' \log m$ and $k_0 = \log \frac{1}{\epsilon} + 1$. We first show by the probabilistic method the existence of such a code C satisfying the distance requirement for all subsets $\Gamma \subseteq [m]$ of size no larger than k_0.

For each $j \in [L]$ independently, we sample the j-th bits of all encodings, $C_j(1), \dots, C_j(m)$, uniformly at random conditioned on the event that exactly $\lfloor \frac{m}{2} \rfloor$ of them are 0 while the remaining $\lceil \frac{m}{2} \rceil$ of them are 1. Consider each fixed $k \le k_0$ and subset $\Gamma \in \binom{[m]}{k}$, we have

$$\begin{aligned}
\mathbb{E}[\mathsf{ns}_C(\Gamma)] &= 1 - \frac{1}{L} \cdot \sum_{j=1}^{L} \Pr(\exists b \in \{0,1\} \forall i \in \Gamma : C_j(i) = b) \\
&= 1 - \frac{\binom{m-k}{\lfloor m/2 \rfloor - k} + \binom{m-k}{\lceil m/2 \rceil - k}}{\binom{m}{\lfloor m/2 \rfloor}} \\
&= 1 - \frac{\binom{m}{k} \cdot \binom{m-k}{\lfloor m/2 \rfloor - k} + \binom{m}{k} \cdot \binom{m-k}{\lceil m/2 \rceil - k}}{\binom{m}{k} \cdot \binom{m}{\lfloor m/2 \rfloor}} \\
&= 1 - \frac{\binom{m}{\lfloor m/2 \rfloor} \cdot \binom{\lfloor m/2 \rfloor}{k} + \binom{m}{\lceil m/2 \rceil} \cdot \binom{\lceil m/2 \rceil}{k}}{\binom{m}{k} \cdot \binom{m}{\lfloor m/2 \rfloor}} \\
&= \mathsf{d}(m,k). \qquad \text{(as } \tbinom{m}{\lfloor m/2 \rfloor} = \tbinom{m}{\lceil m/2 \rceil}\text{)}
\end{aligned}$$

By Chernoff bound (Lemma A.1), the distance requirement is not satisfied by any fixed $k \le k_0$ and subset $\Gamma \in \binom{[m]}{k}$ with probability at most $2 \cdot \exp(-2\epsilon^2 L) < m^{-\frac{10}{\epsilon}}$. As there are at most $k_0 \cdot m^{k_0}$ non-empty subsets of size no larger than k_0, by a union bound, there exists some subset of size no larger than k_0 violating the distance requirement with probability upper bounded by $k_0 \cdot m^{k_0} \cdot m^{-\frac{10}{\epsilon}} < 1$. This implies the existence of a code C with the desired property.

Finally, we show how to make the distance requirement hold also for all subsets $\Gamma \subseteq [m]$ of size larger than k_0. In fact, observe that for all subsets $\Gamma \subseteq [m]$ of size larger than k_0, it holds that for all subsets $\Gamma' \subseteq \Gamma$ of size exactly k_0, we have

$$\begin{aligned}
\mathsf{ns}_C(\Gamma) &\ge \mathsf{ns}_C(\Gamma') \\
&\ge \mathsf{d}(m, k_0) - \epsilon \\
&\ge 1 - \frac{1}{2^{k_0-1}} - \epsilon && \text{(by Item 2 of Claim A.3)} \\
&= 1 - 2\epsilon \\
&\ge \mathsf{d}(m, |\Gamma|) - 2\epsilon.
\end{aligned}$$

As a result, simply replacing ϵ with $\frac{\epsilon}{2}$ in the above construction concludes the proof. □

4.2.2 Lemmas about $\mathsf{d}_{\mathsf{corr}}$

Using the results of Section 4.2.1, we show the following lemma:

Lemma 4.5. *For all $m, k \ge 2$, we have:*

$$\mathsf{d}_{\mathsf{corr}}(m, 2) = \frac{\mathsf{d}(m, 2)}{2} \qquad \textit{and} \qquad \lim_{m \to \infty} \mathsf{d}_{\mathsf{corr}}(m, k) = \frac{1}{2} - \frac{\binom{k-1}{\lceil k/2 \rceil - 1}}{2^k}.$$

Proof. We use Lemma 4.3 and show that $\mathsf{d}_{\mathsf{corr}}(m, 2) = \mathsf{d}_{\mathsf{erase}}(m, 2)/2$. For this, it suffices to pick an arbitrary code $C : [m] \to \{0, 1\}^L$ and an arbitrary $d \in [0, 1]$ and show that C is less-than-2-list decodable for erasures up to radius d if and only if it is less-than-2-list decodable for corruptions up to radius $d/2$. Fix such a $C : [m] \to \{0, 1\}^L$ and $d \in [0, 1]$. Observe that C is less-than-2-list decodable for erasures up to radius d if and only if $\Delta(C(i), C(j)) \ge d$ for all $i, j \in [m]$. Also, observe that C is less-than-2-list decodable for corruptions up to radius d if and only if $\Delta(C(i), C(j)) \ge 2d$ for all $i, j \in [m]$. The result (and therefore, the first equation) follows.

The second equation is a known result first proved in [Bli86]; see also [ABP19]. □

5 Protocols Against Erasures

In this section, we show one direction of Theorem 1.1, as formalized below. Later, in Section 5, we prove the other direction.

Theorem 5.1. *For all $\epsilon > 0$ and $r, n \in \mathbb{N}$, there exists a constant-rate (polynomial in ϵ) protocol for message transfer with r rounds of feedback, input length n, and the following resilience over the binary erasure channel:*

$$\begin{cases} \frac{5}{7} - \epsilon, & \text{if } r = 1 \\ 1 - \frac{7}{12(r+1)} - \epsilon, & \text{if } r > 1 \end{cases}.$$

We prove Theorem 5.1 in the rest of this section. Throughout, we fix $\epsilon > 0$ and $r, n \in \mathbb{N}$. We assume $r < \frac{10}{\epsilon}$. This is without loss of generality as a protocol for large r follows from a protocol for smaller r.

5.1 Our Protocol

Let K be the constant from Lemma 4.4 for ϵ. For all $K' \geq K$ and all $m \geq 1$, let $C_{m,K'} : [m] \to \{0,1\}^{K' \log m}$ be as promised by Lemma 4.4. We will omit K' when it is clear from context. For a set Γ of size m, we will also view C_m as a code $C_\Gamma : \Gamma \to \{0,1\}^{K' \log m}$. Our protocol is given in Algorithm 1, where the lengths of the rounds are given as follows:

$$L_i = \begin{cases} \frac{4}{3} \cdot Kn, & \text{if } i = r = 1 \\ Kn, & \text{otherwise} \end{cases}. \tag{8}$$

Algorithm 1 Message transfer protocol over the erasure channel with $r \geq 1$ feedback rounds.

Input: Alice has input $x \in \Gamma_0 = \{0,1\}^n$.
Output: Bob outputs $y \in \{0,1\}^n$.

1: **for** $i = 1, \ldots, r+1$ **do**
2: Alice sends $C_{\Gamma_{i-1}}(x) \in \{0,1\}^{L_i}$ bit by bit.
3: Bob receives $\tau_i \in \{0,1,\perp\}^{L_i}$ and sends τ_i via the noiseless feedback channel.
4: Bob computes

$$\Gamma_i = \{x' \in \Gamma_{i-1} \mid \forall j \in [L_i] : \tau_{i,j} \in \{C_{\Gamma_{i-1},j}(x'), \perp\}\}.$$

5: If $i \leq r$, Alice receives τ_i as feedback and also computes Γ_i as above.
6: **end for**
7: Bob outputs the lexicographically first element in Γ_{r+1}, aborting if $\Gamma_{r+1} = \emptyset$.

5.2 Analysis

We now analyze Algorithm 1 and finish proving Theorem 5.1. That the protocol is constant rate is clear from Algorithm 1. It remains to show that it has the claimed noise resilience. For this, we fix an input x for Alice and an erasure adversary Adv for the protocol with desired budget as in Theorem 5.1. Observe that fixing x and Adv fixes the value of all the variables in the execution of Algorithm 1. For the analysis, we first show that:

Lemma 5.2. *For all $i \in [0, r+1]$, we have $x \in \Gamma_i$.*

Proof. The base case of $i = 0$ holds trivially. For $i \geq 1$, we know $x \in \Gamma_{i-1}$ by induction. Since Alice sends $C_{\Gamma_{i-1}}(x)$ in the i-th round and the adversary is only capable of erasing some of the symbols Alice sends, what Bob receives, namely τ_i, must still be compatible with $C_{\Gamma_{i-1}}(x)$. Therefore, with the help of the noiseless feedback channel, Alice and Bob always agree on a subset $\Gamma_i \subseteq \Gamma_{i-1}$ that still contains x, at the end of the i-th round. □

Lemma 5.3. *For all $m \geq k' \geq k \geq 2$ such that $(k', k) \neq (3, 2)$, it holds that*

$$\mathsf{d}(m, k') + \mathsf{d}(k', k) \geq 1 + \mathsf{d}(m, k).$$

The proof of Lemma 5.3 is deferred to Appendix A.3. At a high level, Lemma 5.3 will be applied as follows: Consider an adversary that shrinks the set Γ from size m to size k' in a given round and from size k' to size k in the next round. Lemma 5.3 shows that, if the two rounds are of equal length (recall from Eq. (8) that the round lengths are always the same except when $i = r = 1$), then it is always better for the adversary to erase one of the rounds completely and shrink from size m to size k directly in the other round.

We now divide the proof into two cases based on whether or not $r = 1$.

5.2.1 Proof of Theorem 5.1 When $r = 1$

Let $k_i = |\Gamma_i|$ for $i \in [0, 2]$. We prove the theorem by showing that $k_2 \leq 1$. Together with Lemma 5.2 and Line 7, this shows the correctness of Algorithm 1.

Observe that at the beginning of the i-th round, Alice and Bob agree on Γ_{i-1}, the subset of all remaining possibilities for x from the perspective of Bob, that are still consistent with the partial transcript $\tau_1, \ldots, \tau_{i-1}$ so far. In order to keep Bob confused among Γ_i, the adversary has to erase at least a $\mathsf{d}(k_{i-1}, k_i) - \epsilon$ fraction of Alice's i-th message, due to Lemma 4.4. As this holds for all rounds, the overall fraction of erasures is lower bounded by

$$\frac{4}{7}(\mathsf{d}(k_0, k_1) - \epsilon) + \frac{3}{7}(\mathsf{d}(k_1, k_2) - \epsilon) = \frac{4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2)}{7} - \epsilon.$$

Now suppose $k_2 \geq 2$. It is sufficient to show $4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) \geq 5$ for a contradiction. Without loss of generality, we assume $k_2 = 2$ since $\mathsf{d}(k_1, k_2)$ only decrease as k_2 becomes smaller by Item 3 of Claim A.3. If $k_1 = 3$, we have

$$4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) = 4\mathsf{d}(k_0, 3) + 3\mathsf{d}(3, 2) \geq 4 \cdot \frac{3}{4} + 3 \cdot \frac{2}{3} = 5$$

by Item 2 of Claim A.3. Otherwise, by Lemma 5.3, we also get

$$4\mathsf{d}(k_0, k_1) + 3\mathsf{d}(k_1, k_2) = 4(\mathsf{d}(k_0, k_1) + \mathsf{d}(k_1, 2)) - \mathsf{d}(k_1, 2)$$

$$
\begin{aligned}
&\geq 4(1+\mathsf{d}(k_0,2)) - \mathsf{d}(k_1,2) \\
&\geq 3 + 4\mathsf{d}(k_0,2) && \text{(as } \mathsf{d}(\cdot,\cdot) \text{ is always upper bounded by 1)} \\
&\geq 3 + 4\cdot\frac{1}{2} && \text{(by Item 2 of Claim A.3)} \\
&= 5.
\end{aligned}
$$

5.2.2 Proof of Theorem 5.1 When $r > 1$

Let $k_i = |\Gamma_i|$ for $i \in [0, r+1]$. Similarly to the proof in Section 5.2.1, we show that $k_{r+1} \leq 1$. This ensures Bob outputs the correct $y = x$ because of Lemma 5.2 and Line 7. Using a similar argument to Section 5.2.1, we have that the overall fraction of erasures is lower bounded by

$$
\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i) - \epsilon.
$$

Now for the purpose of contradiction, suppose that $k_{r+1} \geq 2$. It is sufficient to show

$$
\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i) \geq 1 - \frac{7}{12(r+1)}.
$$

In the following, we again assume without loss of generality that $k_{r+1} = 2$ since $\mathsf{d}(k_r, k_{r+1})$ decreases as k_{r+1} becomes smaller by Item 3 of Claim A.3. Let $j = \min\{t \in [r+1] \mid k_t \leq 3\}$. By repeatedly applying Lemma 5.3, we have

$$
\begin{aligned}
&\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i) \\
&\qquad \geq \frac{1}{r+1}\cdot\left(1 + \mathsf{d}(k_0,k_2) + \sum_{i=3}^{r+1}\mathsf{d}(k_{i-1},k_i)\right) \\
&\qquad\ \ \vdots \\
&\qquad \geq \frac{1}{r+1}\cdot\left(j - 1 + \mathsf{d}(k_0,k_j) + \sum_{i=j+1}^{r+1}\mathsf{d}(k_{i-1},k_i)\right).
\end{aligned}
$$

Since $k_0 \geq \cdots \geq k_{r+1} = 2$ by definition of Γ_i, either $k_j = 2$ or $k_j = 3$.

In the former case where $k_j = 2$, we also have $k_{j+1} = \cdots = k_{r+1} = 2$ and thus

$$
\begin{aligned}
&\frac{1}{r+1}\cdot\sum_{i=1}^{r+1}\mathsf{d}(k_{i-1},k_i) \\
&\qquad \geq \frac{1}{r+1}\cdot\left(j - 1 + \mathsf{d}(k_0,k_j) + \sum_{i=j+1}^{r+1}\mathsf{d}(k_{i-1},k_i)\right)
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{r+1} \cdot (j - 1 + \mathsf{d}(k_0, 2) + (r+1-j) \cdot \mathsf{d}(2,2)) \\
&\geq \frac{1}{r+1} \cdot \left(j - 1 + \frac{1}{2} + r + 1 - j\right) && \text{(by Item 2 of Claim A.3)} \\
&= 1 - \frac{1}{2(r+1)} \\
&\geq 1 - \frac{7}{12(r+1)}.
\end{aligned}
$$

In the latter case where $k_j = 3$, let $j' = \min\{t \in [r+1] \mid k_t = 2\}$. Then we have

$$
\begin{aligned}
&\frac{1}{r+1} \cdot \sum_{i=1}^{r+1} \mathsf{d}(k_{i-1}, k_i) \\
&\quad \geq \frac{1}{r+1} \cdot \left(j - 1 + \mathsf{d}(k_0, k_j) + \sum_{i=j+1}^{r+1} \mathsf{d}(k_{i-1}, k_i)\right) \\
&\quad = \frac{1}{r+1} \cdot \big(j - 1 + \mathsf{d}(k_0, 3) + (j' - 1 - j) \cdot \mathsf{d}(3,3) + \mathsf{d}(3,2) + (r + 1 - j') \cdot \mathsf{d}(2,2)\big) \\
&\quad \geq \frac{1}{r+1} \cdot \left(j - 1 + \frac{3}{4} + j' - 1 - j + \frac{2}{3} + r + 1 - j'\right) && \text{(by Item 2 of Claim A.3)} \\
&\quad = 1 - \frac{7}{12(r+1)}.
\end{aligned}
$$

This concludes the proof.

6 Impossibility Result for Erasures

In this section, we show the other direction of Theorem 1.1, as formalized below.

Theorem 6.1. *For all $r \in \mathbb{N}$, there exists an $n \in \mathbb{N}$ such that the resilience of any protocol for message transfer with r rounds of feedback and input length n over the binary erasure channel is at most:*

$$
\begin{cases} \frac{5}{7}, & \text{if } r = 1 \\ 1 - \frac{7}{12(r+1)}, & \text{if } r > 1 \end{cases}.
$$

We prove Theorem 6.1 in the rest of this section. Throughout, we work with a fixed $r \in \mathbb{N}$ and define n to be large enough for asymptotic inequalities to hold. We now divide the proof into two cases based on whether or not $r = 1$.

6.1 Proof of Theorem 6.1 When $r = 1$

Fix a protocol Π with input length n and one round of feedback. Recall Eq. (5) and let L_1, L_2 be the lengths of Alice's messages sent in the two rounds, and $f_1 : \{0,1\}^n \to \{0,1\}^{L_1}$, f_2 :

$\{0,1\}^n \times \{0,1,\perp\}^{L_1} \to \{0,1\}^{L_2}$ be the two message functions Alice uses in the two rounds.

First suppose that $L_1 \geq \frac{4}{7}(L_1+L_2)$. By Lemma 4.3, there exists a subset $\Gamma = \{x_1, x_2\} \in \binom{\{0,1\}^n}{2}$ such that $\mathsf{ns}_{f_1}(\Gamma) \leq \mathsf{d}(2^n, 2)$. This implies the adversary is able to erase a $\mathsf{d}(2^n, 2)$ fraction of Alice's first message so that Bob's view when Alice's input is x_1 is identical to Bob's view when Alice's input is x_2, and therefore Bob is forced to send the same feedback $\tau_1 \in \{0,1,\perp\}^{L_1}$ in both cases. Now the adversary simply erases Alice's second message entirely implying that Bob can never output the correct answer. By Item 2 of Claim A.3, the overall fraction of erasures is upper bounded as

$$\mathsf{d}(2^n,2) \cdot \frac{L_1}{L_1+L_2} + 1 \cdot \frac{L_2}{L_1+L_2} \leq \frac{4 \cdot \mathsf{d}(2^n,2)+3}{7} \xrightarrow{n\to\infty} \frac{5}{7}.$$

Now consider the other case where $L_1 \leq \frac{4}{7}(L_1+L_2)$. Again by Lemma 4.3, there exists a subset $\Gamma = \{x_1, x_2, x_3\} \in \binom{\{0,1\}^n}{3}$ such that $\mathsf{ns}_{f_1}(\Gamma) \leq \mathsf{d}(2^n, 3)$. In this case, the adversary erases a $\mathsf{d}(2^n, 3)$ fraction of Alice's first message so that Bob's view is the same when Alice's input is any of x_1, x_2, x_3. Bob must send the same feedback $\tau_1 \in \{0,1\}^{L_1}$ in all three cases. Note that $f_2(\cdot, \tau_1)$ can also be viewed as a valid code and thus Lemma 4.3 still applies. In particular, it is always possible to erase a $\mathsf{d}(3,2) = \frac{2}{3}$ fraction of Alice's second message so that for at least two of x_1, x_2, x_3, Bob's view remains the same at the end of the protocol. This concludes the proof as the overall fraction of erasures is at most

$$\mathsf{d}(2^n,3) \cdot \frac{L_1}{L_1+L_2} + \frac{2}{3} \cdot \frac{L_2}{L_1+L_2} \leq \frac{4 \cdot \mathsf{d}(2^n,3)+3\cdot\frac{2}{3}}{7} \xrightarrow{n\to\infty} \frac{5}{7}.$$

6.2 Proof of Theorem 6.1 When $r > 1$

Fix a protocol Π with input length n and r rounds of feedback. Recall Eq. (5) and for $t \in [r+1]$, let L_t be the length of Alice's message sent in the t-th round. Let $L = \sum_{t=1}^{r+1} L_t$.

We prove the theorem using an approach similar to Section 6.2, *i.e.*, the adversary is always able to erase Alice's messages in such a way that Bob has the same view at the end of the protocol for at least two different inputs. In particular, the adversary erases the entire messages of all rounds except for $i = \arg\max_{t\in[r+1]} L_t$, the longest round, and $j = \arg\max_{t\in[r+1]\setminus\{i\}} L_t$, the second longest round. Then we have

$$L_i \geq \frac{L}{r+1}, \tag{9}$$

$$L_j \geq \frac{L-L_i}{r}. \tag{10}$$

First consider the case where $i < j$. Since the first $i-1$ rounds are completely erased, Bob obviously has the same view for all possible inputs at the beginning of the i-th round. By Lemma 4.3, the adversary can erase a $\mathsf{d}(2^n, 3)$ fraction of Alice's i-th message so that Bob's view is the same when Alice's input is any of some subset $\Gamma = \{x_1, x_2, x_3\} \subseteq \{0,1\}^n$.

This remains true at the beginning of the j-th round as all intermediate rounds are completely erased. Now again by Lemma 4.3, the adversary is able to erase a $\mathsf{d}(3,2) = \frac{2}{3}$ fraction of Alice's j-th message so that Bob still has the same view at the end of the j-th round, for at least two of x_1, x_2, x_3. As all remaining rounds are also completely erased, Bob can never output the correct answer at the end of the protocol. By Item 2 of Claim A.3, the overall fraction of erasures is upper bounded as

$$\begin{aligned}
&\mathsf{d}(2^n,3) \cdot \frac{L_i}{L} + \frac{2}{3} \cdot \frac{L_j}{L} + 1 \cdot \frac{L - L_i - L_j}{L} \\
&\quad = 1 - \big(1 - \mathsf{d}(2^n,3)\big) \cdot \frac{L_i}{L} - \frac{1}{3} \cdot \frac{L_j}{L} \\
&\quad \le 1 - \big(1 - \mathsf{d}(2^n,3)\big) \cdot \frac{L_i}{L} - \frac{1}{3r} \cdot \frac{L - L_i}{L} && \text{(by Eq. (10))} \\
&\quad = 1 - \frac{1}{3r} - \left(1 - \mathsf{d}(2^n,3) - \frac{1}{3r}\right) \cdot \frac{L_i}{L} \\
&\quad \le 1 - \frac{1}{3r} - \left(1 - \mathsf{d}(2^n,3) - \frac{1}{3r}\right) \cdot \frac{1}{r+1} \\
&\qquad\qquad \text{(as } 1 - \mathsf{d}(2^n,3) \xrightarrow{n\to\infty} \tfrac{1}{4} > \tfrac{1}{3r} \text{ for } r \ge 2\text{, and by Eq. (9))} \\
&\quad \xrightarrow{n\to\infty} 1 - \frac{7}{12(r+1)}.
\end{aligned}$$

Now suppose that $i > j$. In this case, a similar argument shows the adversary must be able to confuse Bob by erasing a $\mathsf{d}(2^n,3)$ fraction of Alice's j-th message as well as a $\mathsf{d}(3,2) = \frac{2}{3}$ fraction of Alice's i-th message (in addition to completely erasing all other rounds of messages). Observe that $L_i \ge L_j$ by definition and that $\mathsf{d}(2^n,3) \xrightarrow{n\to\infty} \frac{3}{4} \ge \frac{2}{3}$. So the overall fraction of erasures is at most

$$\mathsf{d}(2^n,3) \cdot \frac{L_j}{L} + \frac{2}{3} \cdot \frac{L_i}{L} + 1 \cdot \frac{L - L_i - L_j}{L} \le \mathsf{d}(2^n,3) \cdot \frac{L_i}{L} + \frac{2}{3} \cdot \frac{L_j}{L} + \frac{L - L_i - L_j}{L},$$

which has the desired upper bound as already shown above.

7 Impossibility Result for Corruptions

We now explore the setting with corruptions. As discussed in Section 1, we will focus on the case of a single feedback round. In this section, we present a simple proof of Theorem 1.2. Apart from deriving an upper bound (conjectured to be tight) on the maximum possible noise resilience, the proof also helps shape our protocol to the extent that it gives insights into when the feedback has to occur. Specifically, we get that the ratio between the lengths of the two rounds has to be around $\frac{8}{15}$ if the protocol aims to achieve a matching lower bound of $\frac{7}{23}$ on the noise resilience.

Proof of Theorem 1.2. Fix Π to be any protocol with a fixed order of speaking for message transfer over a binary corruption channel with a single round of noiseless feedback. Let n be the length of input and L_1, L_2 the length of communication before and after feedback, respectively. Also let $C_1 : \{0,1\}^n \to \{0,1\}^{L_1}$ and $C_2 : \{0,1\}^n \to \{0,1\}^{L_2}$ be the two codes Alices uses in the two rounds, respectively.

First suppose $L_1 \geq \frac{8}{23}(L_1+L_2)$. There exist three codewords $x_1, x_2, x_3 \in \{0,1\}^n$ and some corruption $\tau_1 \in \{0,1\}^{L_1}$ such that $\Delta(C_1(x_i), \tau_1) \leq \mathsf{d}_{\mathsf{corr}}(2^n, 3) \cdot L_1$ for all $i \in [3]$. Suppose τ_1 is what Bob receives in the first round. Alice can learn nothing but τ_1 from the feedback. In the second round, it is always possible to corrupt Alice message to some $\tau_2 \in \{0,1\}^{L_2}$ such that $\Delta(C_2(x_i), \tau_2) \leq \mathsf{d}_{\mathsf{corr}}(3, 2) \cdot L_2 = \frac{1}{3} L_2$ in at least two of the three cases. Therefore, by Lemma 4.5, the adversary is capable of confusing Bob between at least two possibilities with the total fraction of corruptions upper bounded by

$$\mathsf{d}_{\mathsf{corr}}(2^n, 3) \cdot \frac{L_1}{L_1 + L_2} + \frac{1}{3} \cdot \frac{L_2}{L_1 + L_2} \xrightarrow{n \to \infty} \frac{1}{4} \cdot \frac{L_1}{L_1 + L_2} + \frac{1}{3} \cdot \frac{L_2}{L_1 + L_2} \leq \frac{7}{23}.$$

In the other case of $L_1 \leq \frac{8}{23}(L_1 + L_2)$, a similar strategy is used with the only difference that the adversary seeks five remaining possibilities after the first round. As a result, $\mathsf{d}_{\mathsf{corr}}(2^n, 5) \cdot L_1$ and $\mathsf{d}_{\mathsf{corr}}(5, 2) \cdot L_2 = \frac{3}{10} L_2$ are the corruptions required in the two rounds, respectively. Again by Lemma 4.5, the total fraction of corruptions then becomes

$$\mathsf{d}_{\mathsf{corr}}(2^n, 5) \cdot \frac{L_1}{L_1 + L_2} + \frac{3}{10} \cdot \frac{L_2}{L_1 + L_2} \xrightarrow{n \to \infty} \frac{5}{16} \cdot \frac{L_1}{L_1 + L_2} + \frac{3}{10} \cdot \frac{L_2}{L_1 + L_2} \leq \frac{7}{23},$$

as desired, concluding the proof. □

8 An Equivalent Form of Conjecture 1.3

In this section, we recast Conjecture 1.3 in a form that allows it to be used for protocols. This is done in Lemma 8.1 which will be useful in both directions of Theorem 1.4. For this, we first show how both the messages sent by Alice and the adversary's corruptions can be viewed as vectors.

8.1 The Vectors Interpretation

Let $m, L \in \mathbb{N}$ and consider a code $C : [m] \to \{0,1\}^L$. Observe that the distance guarantees of C do not depend on the order in which its coordinates are written, namely, we can take any permutation π over $[L]$ and permute $C(i)$ for all i by π and preserve the distance guarantees. This means that the "essence" of C is simply, for all $b : [m] \to \{0,1\}$ what is the fraction of coordinates j of C that "match" b, *i.e.*, what is the fraction of coordinates j such that it holds for all $i \in [m]$ that $C_j(i) = b(i)$.

To make this formal, we view such a code C as a distribution $h \in \Delta^{2^m - 1}$ over functions $b : [m] \to \{0,1\}$ where probability of sampling a function $b : [m] \to \{0,1\}$ is exactly the fraction of coordinates j of C that match b. The Hamming distance between two distinct codewords i and i' is then equal to

$$\mathsf{D}(h)_{\{i,i'\}} = \sum_{b:[m]\to\{0,1\}} \mathbb{1}(b(i) \neq b(i')) \cdot h_b. \tag{11}$$

We will denote by $\mathsf{D}(h)$ the vector $\mathsf{D}(h) = \big(\mathsf{D}(h)_{\{i,i'\}}\big)_{\{i,i'\}\in\binom{m}{2}}$.

Similarly, note that when the adversary corrupts a message sent by Alice, all that matters is, for any given m and $b : [m] \to \{0,1\}$, what fraction of coordinates that match b were corrupted by the adversary. Thus, we can capture an adversary by a vector $g \in [0,1]^{2^m}$ where, for all $b : [m] \to \{0,1\}$, the value of g_b is simply the fraction of coordinates j such that Bob receives a 1 in coordinate j out of all the coordinates that match b. In this interpretation, if the code sent by Alice is captured by $f \in \Delta^{2^m-1}$ and the adversary is captured by $g \in [0,1]^{2^m}$, then the total number of corruptions needed to corrupt two messages, say $i, i' \in [m]$, of Alice according to g is given by:

$$\mathsf{D}(f,g)_{\{i,i'\}} = \sum_{i''\in\{i,i'\}} \sum_{b:[m]\to\{0,1\}} f_b \cdot \Big(b(i'') \cdot (1-g_b) + (1-b(i'')) \cdot g_b\Big). \tag{12}$$

We will denote by $\mathsf{D}(f,g)$ the vector $\mathsf{D}(f,g) = \big(\mathsf{D}(f,g)_{\{i,i'\}}\big)_{\{i,i'\}\in\binom{m}{2}}$.

8.2 Recasting Conjecture 1.3

We are now ready to write an equivalent form of Conjecture 1.3.

Lemma 8.1. *Conjecture 1.3 holds if and only if for all $m > 0$, all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:*

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}\Big(\frac{\vec{1}}{2^m}, g\Big).$$

Proof. We will actually show a slightly stronger statement that, for all $m > 0$, Conjecture 1.3 holds for graphs with m vertices if and only if for all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}\Big(\frac{\vec{1}}{2^m}, g\Big). \tag{13}$$

Fix $m > 0$. Observe that the set P defined as:

$$P = \Big\{v \in \mathbb{R}^{\binom{m}{2}} \mid \exists h \in \Delta^{2^m-1} : \mathsf{D}(h) \geq v\Big\},$$

is a closed and convex set. The convexity is because Δ^{2^m-1} is convex, $\mathsf{D}(\cdot)$ is linear, and for all $\lambda \in [0,1]$, the fact that $\mathsf{D}(h_1) \geq v_1$ and $\mathsf{D}(h_2) \geq v_2$ implies that $\mathsf{D}(\lambda h_1 + (1-\lambda)h_2) = \lambda \cdot \mathsf{D}(h_1) + (1-\lambda) \cdot \mathsf{D}(h_2) \geq \lambda v_1 + (1-\lambda)v_2$ while the closed-ness is because Δ^{2^m-1} is closed and $\mathsf{D}(\cdot)$ is continuous (in fact, linear).[11] Next, define:

$$Q = \left\{ \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, g\right) \mid g \in [0,1]^{2^m} \right\}.$$

Again, using the fact that $[0,1]^{2^m}$ is closed and convex and $\mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, \cdot\right)$ is linear, we get that Q is closed and convex. Observe that showing Eq. (13) is equivalent to showing $Q \subseteq P$. Define the set:

$$\mathcal{Z} = \left\{ z \in \mathbb{R}^{\binom{m}{2}} \mid z \geq \overrightarrow{0}, \sum_{i<i' \in [m]} z_{\{i,i'\}} = 1 \right\}. \tag{14}$$

Namely, $\mathcal{Z}$ is the set of all non-negative vectors in $\mathbb{R}^{\binom{m}{2}}$ whose entries sum to 1. We claim that $Q \subseteq P$ if and only if for all $z \in \mathcal{Z}$, we have $\max_{v \in P}\langle z, v\rangle \geq \max_{v \in Q}\langle z, v\rangle$. Indeed, the "only if" is straightforward and we focus on the "if" direction and prove it in the contrapositive. Suppose that $Q \not\subseteq P$ implying that there exists $x \in Q \setminus P$. By the separating hyperplane theorem, there exists a vector $z \in \mathbb{R}^{\binom{m}{2}}$ such that $\langle z, x\rangle > \max_{v \in P}\langle z, v\rangle$. Now, observe from the definition of P that this can only happen if $z \geq \overrightarrow{0}$ and z is not the all-zeros vector. Thus, by scaling, we can assume that $z \in \mathcal{Z}$. As $x \in Q$, we get that $\max_{v \in Q}\langle z, v\rangle > \max_{v \in P}\langle z, v\rangle$, as desired.

Claim 8.2. *For all $z \in \mathcal{Z}$, we have:*

$$\max_{v \in P}\langle z, v\rangle = \max_{b:[m]\to\{0,1\}} \sum_{\substack{i<i' \in [m] \\ b(i) \neq b(i')}} z_{\{i,i'\}}.$$

Claim 8.3. *For all $z \in \mathcal{Z}$, we have:*

$$\max_{v \in Q}\langle z, v\rangle = \frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [m]}\left[\min\left(\sum_{i<i' \in S} z_{\{i,i'\}}, \sum_{i<i' \in \overline{S}} z_{\{i,i'\}}\right)\right].$$

Using Claims 8.2 and 8.3, note that Eq. (13) is equivalent to showing that for all $z \in \mathbb{R}^{\binom{m}{2}}$ such that $z \geq \overrightarrow{0}$, we have:

$$\max_{b:[m]\to\{0,1\}} \sum_{\substack{i<i' \in [m] \\ b(i) \neq b(i')}} z_{\{i,i'\}} \geq \frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [m]}\left[\min\left(\sum_{i<i' \in S} z_{\{i,i'\}}, \sum_{i<i' \in \overline{S}} z_{\{i,i'\}}\right)\right].$$

Considering z as the edge weights on a graph G with m vertices, we get that the is equivalent

[11] Observe that if a set $S \subseteq \mathbb{R}^d$ is closed, then $S' = \left\{x' \in \mathbb{R}^d \mid \exists x \in S : x \geq x'\right\}$ is also closed.

to:

$$\textsf{Max-Cut}(G) \geq \frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [m]} \left[\min\left(\textsf{wt}(S), \textsf{wt}(\overline{S}) \right) \right],$$

which is exactly Conjecture 1.3, as desired. $\square$

It remains to show claim Claims 8.2 and 8.3, and we do this next.

Proof of Claim 8.2. Fix $z \in \mathcal{Z}$ and recall from Eq. (14) that $z \geq \overrightarrow{0}$. From this and the definition of P, conclude that $\max_{v \in P} \langle z, v \rangle$ is attained at $\mathsf{D}(h)$ for some $h \in \Delta^{2^m - 1}$. Furthermore, as both $\mathsf{D}(\cdot)$ and the inner product function are linear, we can assume that h is one of the extrema of Δ^{2^m-1}, *i.e.*, one of the 2^m dimensional standard basis vectors. From these, we get:

$$\begin{aligned}
\max_{v \in P} \langle z, v \rangle &= \max_{h \in \Delta^{2^m-1}} \langle z, \mathsf{D}(h) \rangle \\
&= \max_{h \in \Delta^{2^m-1}} \sum_{i < i' \in [m]} z_{\{i,i'\}} \cdot \mathsf{D}(h)_{\{i,i'\}} \\
&= \max_{b:[m] \to \{0,1\}} \sum_{i < i' \in [m]} z_{\{i,i'\}} \cdot \mathbb{1}(b(i) \neq b(i')) && \text{(Eq. (11))} \\
&= \max_{b:[m] \to \{0,1\}} \sum_{\substack{i < i' \in [m] \\ b(i) \neq b(i')}} z_{\{i,i'\}}.
\end{aligned}$$

$\square$

Proof of Claim 8.3. Fix $z \in \mathcal{Z}$. As both $\mathsf{D}(\cdot)$ and the inner product function are linear, we conclude from the definition of Q that $\max_{v \in Q} \langle z, v \rangle$ is attained at one of the extrema of $[0,1]^{2^m}$. We get:

$$\begin{aligned}
\max_{v \in Q} \langle z, v \rangle &= \max_{g \in \{0,1\}^{2^m}} \sum_{i < i' \in [m]} z_{\{i,i'\}} \cdot \left(\frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, g \right)_{\{i,i'\}} \right) \\
&= \frac{14}{15} - \frac{8}{15} \min_{g \in \{0,1\}^{2^m}} \sum_{i < i' \in [m]} z_{\{i,i'\}} \cdot \mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, g \right)_{\{i,i'\}} && \text{(As } z \in \mathcal{Z} \text{ and Eq. (14))} \\
&= \frac{14}{15} - \frac{8}{15} \min_{g \in \{0,1\}^{2^m}} \sum_{i < i' \in [m]} \sum_{i'' \in \{i,i'\}} \sum_{b:[m] \to \{0,1\}} \frac{z_{\{i,i'\}}}{2^m} \cdot \Big(b(i'')(1 - g_b) + (1 - b(i''))g_b \Big) \\
& && \text{(Eq. (12))} \\
&= \frac{14}{15} - \frac{8}{15} \sum_{b:[m] \to \{0,1\}} \min_{g_b \in \{0,1\}} \sum_{i < i' \in [m]} \sum_{i'' \in \{i,i'\}} \frac{z_{\{i,i'\}}}{2^m} \cdot \Big(b(i'')(1 - g_b) + (1 - b(i''))g_b \Big)
\end{aligned}$$

To continue, we take the 2^m and use it to write the sum as an expectation over b. We get:

$$\begin{aligned}
\max_{v\in Q}\langle z,v\rangle &= \frac{14}{15}-\frac{8}{15}\mathop{\mathbb{E}}_{b}\left[\min_{g_b\in\{0,1\}}\sum_{i<i'\in[m]} z_{\{i,i'\}}\cdot\Big((b(i)+b(i'))(1-2g_b)+2g_b\Big)\right]\\
&= \frac{14}{15}-\frac{8}{15}\mathop{\mathbb{E}}_{b}\left[\min_{g_b\in\{0,1\}}\sum_{i<i'\in[m]} z_{\{i,i'\}}\cdot\Big((b(i)+b(i'))(1-2g_b)+2g_b\Big)\right]\\
&= \frac{14}{15}-\frac{8}{15}\mathop{\mathbb{E}}_{b}\left[\min_{g_b\in\{0,1\}}\sum_{\substack{i<i'\in[m]\\ b(i)+b(i')=1}} z_{\{i,i'\}}+\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=0}} 2g_b z_{\{i,i'\}}+\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=1}} 2(1-g_b) z_{\{i,i'\}}\right]\\
&= \frac{14}{15}-\frac{8}{15}\mathop{\mathbb{E}}_{b}\left[\sum_{\substack{i<i'\in[m]\\ b(i)+b(i')=1}} z_{\{i,i'\}}+2\cdot\min\left(\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=0}} z_{\{i,i'\}},\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=1}} z_{\{i,i'\}}\right)\right]\\
&= \frac{2}{3}-\frac{16}{15}\mathop{\mathbb{E}}_{b}\left[\min\left(\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=0}} z_{\{i,i'\}},\sum_{\substack{i<i'\in[m]\\ b(i)=b(i')=1}} z_{\{i,i'\}}\right)\right]. \quad (\text{As } z\in\mathcal{Z} \text{ and Eq. (14)})
\end{aligned}$$

Interpreting b as the indicator vector of a uniformly random set $S\subseteq[m]$ finishes the proof as we get:

$$\max_{v\in Q}\langle z,v\rangle = \frac{2}{3}-\frac{16}{15}\cdot\mathop{\mathbb{E}}_{S\subseteq[m]}\left[\min\left(\sum_{i<i'\in S} z_{\{i,i'\}},\sum_{i<i'\in\overline{S}} z_{\{i,i'\}}\right)\right].$$

□

9 Protocols Against Corruptions

We are now ready to prove Theorem 1.4. The "if" direction is shown here while the "only if" direction is shown in the next section.

9.1 The Protocol

We first show the "if" direction, *i.e.*, we show that Conjecture 1.3 implies that Theorem 1.2 is tight. This direction is formalized as Theorem 9.1 below:

Theorem 9.1. *Assume that Conjecture 1.3 holds. For all $\epsilon>0$ and $n\in\mathbb{N}$, there exists a constant-rate (depending on ϵ) protocol for message transfer with one round of feedback, input length n, and resilience $\frac{7}{23}-\epsilon$ over the binary corruption channel.*

At a high level, the idea for our protocol that proves Theorem 9.1 is to generalize Algorithm 1 when $r = 1$. More specifically, there are two rounds of communication by Alice. We are going to keep the first round essentially unchanged as randomly sampled codes are used, although we will need a stronger guarantee, formalized as (k, ϵ)-random codes below. Regarding the second round, we adapt the codes used in the second round of Algorithm 1 (when $r = 1$) to the case when not all codewords are treated the same, formalized as dc-codes.

(k, ϵ)-random codes. We now define the notion of a (k, ϵ)-random code.

Definition 9.2. *Let $m, L, k \geq 1$ and $\epsilon \geq 0$. A code $C : [m] \to \{0,1\}^L$ is (k, ϵ)-random if the following holds for all subsets $\Gamma \subseteq [m]$ of size at most k and $b : \Gamma \to \{0,1\}$:*

$$\left| \frac{1}{L} \cdot \sum_{j=1}^{L} \mathbb{1}[\forall i \in \Gamma : C_j(i) = b(i)] - \frac{1}{2^{|\Gamma|}} \right| \leq \epsilon.$$

Note that the inequality above is satisfied by a uniformly random code *in expectation*. Thus, roughly speaking, a code is (k, ϵ)-random if it satisfies the inequality *pointwise* when we look at any collection of at most k codewords. We next show that such codes can be constructed with constant rate (as needed for our result):

Lemma 9.3. *For all $k \geq 1$ and $\epsilon > 0$, there exists a constant K such that for all $K' \geq K$ and all $m \geq 2$, there exists a (k, ϵ)-random code $C : [m] \to \{0,1\}^{K' \log m}$.*

Proof. Set $K = \frac{10k}{\epsilon^2}$. Let $L = K' \log m$. We show the existence of a (k, ϵ)-random code C by the probabilistic method.

For each $i \in [m]$ and $j \in [L]$ independently, we sample the j-th bit of $C(i)$ uniformly at random. Consider each fixed subset $\Gamma \subseteq [m]$ of size at most k and $b : \Gamma \to \{0,1\}$, we have

$$\mathbb{E}\left[\frac{1}{L} \cdot \sum_{j=1}^{L} \mathbb{1}[\forall i \in \Gamma : C_j(i) = b(i)] \right] = \frac{1}{2^{|\Gamma|}}.$$

By Chernoff bound (Lemma A.1), the randomness requirement is not satisfied by any fixed subset $\Gamma \subseteq [m]$ of size at most k and $b : \Gamma \to \{0,1\}$ with probability at most $2 \cdot \exp(-2\epsilon^2 L) < m^{-10k}$. As there are at most $k \cdot m^k$ non-empty subsets of size at most k, each of them having to satisfy at most 2^k randomness requirements, by a union bound, there exists some subset $\Gamma \subseteq [m]$ of size at most k and $b : \Gamma \to \{0,1\}$ violating the randomness requirement with probability upper bounded by $k \cdot m^k \cdot 2^k \cdot m^{-10k} < 1$. This concludes the proof. $\square$

dc-codes. The (k, ϵ)-random code defined above will be used by Alice in the first round of our protocol. For the second round, she will use a different set of codes that we call dc-codes:

Definition 9.4. *For $m, L \geq 1$ and $\mathsf{dc} : [m] \to [0,1]$, a function $C : [m] \to \{0,1\}^L$ is a* dc*-code (over the binary corruption channel) if for all $\{i,j\} \in \binom{[m]}{2}$, it holds that*

$$\Delta(C(i), C(j)) \geq (\mathsf{dc}(i) + \mathsf{dc}(j)) \cdot L.$$

Definition 9.4 above is closely tied to Definition 4.2 (for $k = 2$) with the only difference being that instead of requiring the same distance guarantee for all codewords, Definition 9.4 has a parameter $\mathsf{dc}(i)$ for each codeword i and the distance guarantees for this codeword are determined by $\mathsf{dc}(i)$. This is needed in the second round of our protocol as prior to the second round, Bob already has some information about Alice's input in the sense that he knows that certain inputs are closer to the message he received in the first round than others. Interestingly, this subtlety does not arise in our protocol for erasure noise as there, either an input is impossible and Bob can rule it out or it is the same distance from the message he received that all the other codewords.

Unlike (k, ϵ)-random codes, we do not have an unconditional proof that the dc-codes that we shall use in our protocol exist. This part of the argument is deferred to the analysis and shall rely on Conjecture 1.3.

Notation. Fix $\epsilon > 0$ and $n \geq 1$ as in Theorem 9.1. Let k be a sufficiently large constant such that $\binom{k-1}{\lceil k/2 \rceil - 1} \cdot \frac{1}{2^k} \leq \frac{\epsilon}{10}$. Also let K be the constant from Lemma 9.3 for k and $\epsilon' = \frac{\epsilon}{10 \cdot 2^k}$. Define $L_1 = 8Kn$ and $L_2 = 15Kn$ and let C be the (k, ϵ')-random code promised by Lemma 9.3 for $K' = 8K$ and $m = 2^n$, and $C_{\mathsf{dc}} : \{0,1\}^n \to \{0,1\}^{L_2}$ be a dc-code (whose existence we shall prove later).

Algorithm 2 Protocol for message transfer over a corruption channel with a single round of feedback.

Input: Alice has input $x \in \{0,1\}^n$.
Output: Bob outputs $y \in \{0,1\}^n$.

Alice sends $C(x) \in \{0,1\}^{L_1}$ bit by bit.
Bob receives $\tau_1 \in \{0,1\}^{L_1}$ and sends τ_1 via the noiseless feedback channel.
Bob computes, for all $y \in \{0,1\}^n$:

$$\mathsf{dc}(y) = \frac{7}{15} - \frac{\Delta(C(y), \tau_1)}{L_2} - \epsilon. \tag{15}$$

Alice receives τ_1 as feedback and also computes $\mathsf{dc}(\cdot)$ as above.
Alice sends $C_{\mathsf{dc}}(x) \in \{0,1\}^{L_2}$ bit by bit.
Bob receives $\tau_2 \in \{0,1\}^{L_2}$ and outputs (breaking ties arbitrarily):

$$\operatorname*{arg\,min}_{y \in \{0,1\}^n} \Big(\Delta(C(y), \tau_1) + \Delta(C_{\mathsf{dc}}(y), \tau_2) \Big).$$

9.2 Proof of Theorem 9.1

We now show that Theorem 9.1 holds. That Algorithm 2 is constant rate is straightforward and we only need to show that it has noise resilience $\frac{7}{23}-\epsilon$. For this, we recall Section 3.2 and fix an input $x \in \{0,1\}^n$ for Alice and a corruption adversary for Algorithm 2 with budget $\frac{7}{23}-\epsilon$. Fixing x and such an adversary fixes the value of all the variables in Algorithm 2 and all we need to show is that Bob outputs x at the end of Algorithm 2 and that the dc-code used by Alice in Line 12 exists. Throughout this proof, we will use the variable name to denote its value, *e.g.*, dc will denote the value of dc that was fixed when we fixed the input x and the adversary.

We first assume the existence of dc-codes and show that Bob outputs x and later show that dc-codes exist.

Bob outputs x. We now show that Bob outputs x. Owing to Line 13, it suffices to show that for all $y \neq x \in \{0,1\}^n$, we have:

$$\Delta(C(x), \tau_1) + \Delta(C_{\mathsf{dc}}(x), \tau_2) < \Delta(C(y), \tau_1) + \Delta(C_{\mathsf{dc}}(y), \tau_2). \tag{16}$$

We first note that, as the budget of our adversary is $\frac{7}{23}-\epsilon$, it holds that:

$$\Delta(C(x), \tau_1) + \Delta(C_{\mathsf{dc}}(x), \tau_2) \leq \left(\frac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) = (7 - 23\epsilon) \cdot Kn. \tag{17}$$

Also, for all $y \neq x \in \{0,1\}^n$, we have by the triangle inequality:

$$\begin{aligned} &\Delta(C(x), \tau_1) + \Delta(C_{\mathsf{dc}}(x), \tau_2) + \Delta(C(y), \tau_1) + \Delta(C_{\mathsf{dc}}(y), \tau_2) \\ &\quad \geq \Delta(C(x), \tau_1) + \Delta(C(y), \tau_1) + \Delta(C_{\mathsf{dc}}(x), C_{\mathsf{dc}}(y)) \\ &\quad \geq \Delta(C(x), \tau_1) + \Delta(C(y), \tau_1) + (\mathsf{dc}(x) + \mathsf{dc}(y)) \cdot L_2 \\ &\quad \geq (14 - 30\epsilon) \cdot Kn \\ &\quad > 2 \cdot (7 - 23\epsilon) \cdot Kn, \end{aligned} \tag{18}$$

where we use Eq. (15) in the penultimate step and Definition 9.4 in the step before. Combining Eqs. (17) and (18) proves Eq. (16).

Existence of a dc-code. We now show the following lemma, whose proof spans the rest of this section.

Lemma 9.5. *If Conjecture 1.3 holds, there exists a dc-code $C_{\mathsf{dc}} : \{0,1\}^n \to \{0,1\}^{L_2}$.*

Define $m = 2^n$ for the rest of this section. In order to prove Lemma 9.5, we work in the vectors interpretation from Section 8. Let $f \in \Delta^{2^m-1}$ be the vector corresponding to the first message sent by Alice and $g \in [0,1]^{2^m}$ be correspond to the corruptions inserted by the adversary in f. Recall that $\epsilon' = \frac{\epsilon}{10 \cdot 2^k}$. We will show that:

Lemma 9.6. *If Conjecture 1.3 holds, there exists $h \in \Delta^{2^m-1}$ such that:*

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}(f, g) - (2\epsilon - \epsilon') \cdot \overrightarrow{1}.$$

We now show Lemma 9.5 assuming Lemma 9.6, and later show Lemma 9.6.

Proof of Lemma 9.5 assuming Lemma 9.6. Since Conjecture 1.3 is assumed to hold, Lemma 9.6 guarantees the existence of $h \in \Delta^{2^m-1}$ such that

$$\mathsf{D}(h)_{\{i,i'\}} \geq \frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}(f, g)_{\{i,i'\}} - 2\epsilon + \epsilon'$$

holds for all $\{i, i'\} \in \binom{[m]}{2}$. We show the existence of a dc-code C_{dc} by the probabilistic method. For each $j \in [L_2]$ independently, we sample the j-th bits of all encodings, namely $C_j(1), \ldots, C_j(m)$, such that for $b : [m] \to \{0, 1\}$, with probability h_b, $C_j(i) = b(i)$ holds simultaneously for all $i \in [m]$. Note that, from Eqs. (12) and (15), we also have that for any $\{i, i'\} \in \binom{[m]}{2}$:

$$\mathsf{dc}(i) + \mathsf{dc}(i') = \frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}(f, g)_{\{i,i'\}} - 2\epsilon \leq \mathsf{D}(h)_{\{i,i'\}} - \epsilon'.$$

Moreover, consider any fixed $\{i, i'\} \in \binom{[m]}{2}$, by Eq. (11), we have

$$\mathbb{E}\left[\frac{\Delta(C(i), C(i'))}{L_2}\right] = \mathsf{D}(h)_{\{i,i'\}} \geq \mathsf{dc}(i) + \mathsf{dc}(i') + \epsilon'.$$

By Chernoff bound (Lemma A.1), the distance requirement is not satisfied by $\{i, i'\}$ with probability at most $2 \cdot \exp(-2 \cdot (\epsilon')^2 L_2) < \exp(-2n)$. By a union bound, there exists some $\{i, i'\}$ violating the distance requirement with probability upper bounded by $\binom{m}{2} \cdot \exp(-2n) < 1$. This concludes the proof. □

9.3 Proof of Lemma 9.6

Proof of Lemma 9.6. We start by showing the code C less-than-k-list decodable for corruptions up to radius $\frac{1}{2} - \frac{\epsilon}{4}$ (see Definition 4.2). For this, we have to show that:

Claim 9.7. *For any $\tilde{\tau} \in \{0, 1\}^{L_1}$, we have:*

$$\left| \left\{ i \in [m] : \Delta(C(i), \tilde{\tau}) < \left(\frac{1}{2} - \frac{\epsilon}{4} \right) \cdot L_1 \right\} \right| < k.$$

Proof. We prove by contradiction. Let $\tilde{\tau}$ be a counterexample and assume that the set in the statement of the claim has at least k elements. Without loss of generality, we assume that these elements are the element of $[k]$. Thus, we know that for all $i \in [k]$, we have

$\Delta(C(i), \tilde{\tau}) < \left(\frac{1}{2} - \frac{\epsilon}{4}\right) \cdot L_1$. It follows that:

$$\min_{\tau' \in \{0,1\}^{L_1}} \sum_{i \in [k]} \Delta(C(i), \tau') \leq \sum_{i \in [k]} \Delta(C(i), \tilde{\tau}) < \left(\frac{1}{2} - \frac{\epsilon}{4}\right) \cdot L_1 k. \tag{19}$$

However, we also have:

$$\min_{\tau' \in \{0,1\}^{L_1}} \sum_{i \in [k]} \Delta(C(i), \tau') = \sum_{j=1}^{L_1} \min_{b' \in \{0,1\}} \sum_{i \in [k]} \mathbb{1}[C_j(i) \neq b'].$$

Recall that, for a function $b : [m] \to \{0,1\}$, the value of f_b is the fraction of coordinates j such that for all $i \in [m]$, we have $C_j(i) = b(i)$. We extend this notation and define, for all sets $\Gamma \subseteq [m]$ and functions $b^* : \Gamma \to \{0,1\}$, the value $f_{b^*}[\Gamma]$ to be the fraction of coordinates j such that for all $i \in \Gamma$, we have $C_j(i) = b^*(i)$. When $\Gamma = [k]$, we simply write $f_{b^*}[k]$ instead of $f_{b^*}[[k]]$. Also, observe that the j-th term of the summation above depends only on $C_j(1), \ldots, C_j(k)$. Grouping terms with the same values of $C_j(1), \ldots, C_j(k)$ together, we get:

$$\min_{\tau' \in \{0,1\}^{L_1}} \sum_{i \in [k]} \Delta(C(i), \tau') = \sum_{b^* : [k] \to \{0,1\}} f_{b^*}[k] \cdot L_1 \cdot \min_{b' \in \{0,1\}} \sum_{i \in [k]} \mathbb{1}[b^*(i) \neq b'].$$

This implies that

$$\begin{aligned}
\min_{\tau' \in \{0,1\}^{L_1}} \sum_{i \in [k]} \Delta(C(i), \tau') &= \sum_{b^* : [k] \to \{0,1\}} f_{b^*}[k] \cdot L_1 \cdot \min\left(\sum_{i \in [k]} b^*(i), k - \sum_{i \in [k]} b^*(i)\right) \\
&= \sum_{s=0}^{k} \sum_{\substack{b^* : [k] \to \{0,1\} \\ \sum_{i \in [k]} b^*(i) = s}} f_{b^*}[k] \cdot L_1 \cdot \min(s, k-s) \\
&\geq \sum_{s=0}^{k} \sum_{\substack{b^* : [k] \to \{0,1\} \\ \sum_{i \in [k]} b^*(i) = s}} \left(\frac{1}{2^k} - \epsilon'\right) \cdot L_1 \cdot \min(s, k-s)
\end{aligned}$$

(C is (k, ϵ')-random, Definition 9.2)

As each term only depends on s and the number of b^* corresponding to a given s is $\binom{k}{s}$, we get:

$$\begin{aligned}
\min_{\tau' \in \{0,1\}^{L_1}} \sum_{i \in [k]} \Delta(C(i), \tau') &\geq \frac{1}{2^k} \cdot L_1 \cdot \sum_{s=0}^{k} \binom{k}{s} \cdot \min(s, k-s) - 2^k k L_1 \epsilon' \\
&\geq L_1 k \cdot \left(\frac{1}{2} - \binom{k-1}{\lceil k/2 \rceil - 1} \cdot \frac{1}{2^k}\right) - 2^k k L_1 \epsilon' \qquad \text{(Lemma A.2)}
\end{aligned}$$

$$\geq L_1 k \cdot \left(\frac{1}{2} - \frac{\epsilon}{10}\right) - 2^k k L_1 \epsilon' \qquad \left(\text{As } \binom{k-1}{\lceil k/2 \rceil - 1} \cdot \frac{1}{2^k} \leq \frac{\epsilon}{10}\right)$$

$$\geq L_1 k \cdot \left(\frac{1}{2} - \frac{\epsilon}{5}\right), \qquad \left(\text{As } \epsilon' = \frac{\epsilon}{10 \cdot 2^k}\right)$$

contradicting Eq. (19). □

We now apply Claim 9.7 on τ_1. Assume without loss of generality that the set in the statement of the claim is contained in $[k]$. Thus, we have for all $i \in [m] \setminus [k]$ that

$$\Delta(C(i), \tau_1) \geq \left(\frac{1}{2} - \frac{\epsilon}{4}\right) \cdot L_1. \tag{20}$$

Our strategy to prove Lemma 9.6 is to use f and g to construct a function $g' \in [0,1]^{2^m}$ that we can apply Lemma 8.1 on (we will have $m' = m$). Roughly speaking, for $b : [m] \to \{0,1\}$, coordinate b of g' only depends on $b(1), \ldots, b(k)$ and is the average of all coordinate of g with the same value of $b(1), \ldots, b(k)$. Formally, for all $b : [m] \to \{0,1\}$, we define g'_b using the equation:

$$g'_b \cdot \sum_{\substack{b:[m]\to\{0,1\} \\ b|_{[k]} = b^*}} f_b = \sum_{\substack{b:[m]\to\{0,1\} \\ b|_{[k]} = b^*}} f_b g_b. \tag{21}$$

Observe that g'_b is determined by $b|_{[k]}$. This allows to write, for a function $b^* : [k] \to \{0,1\}$, the value g'_{b^*} as the common value of g'_b for all $b : [m] \to \{0,1\}$ satisfying $b|_{[k]} = b^*$. Applying Lemma 8.1 on m and g' (recall that Lemma 9.6 assumes Conjecture 1.3), we get that there exists $h \in \Delta^{2^m - 1}$ such that:

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, g'\right).$$

Thus, in order to show Lemma 9.6, it suffices to show that $\mathsf{D}\left(\frac{\overrightarrow{1}}{2^m}, g'\right) \leq \mathsf{D}(f,g) + 2 \cdot \overrightarrow{\epsilon}$. We show this inequality coordinate-wise. Due to Eq. (12), this follows if we show that for all $i \in [m]$, we have:

$$\sum_{b:[m]\to\{0,1\}} \frac{1}{2^m} \cdot \Big(b(i) \cdot (1 - g'_b) + (1 - b(i)) \cdot g'_b\Big) \leq \epsilon + \sum_{b:[m]\to\{0,1\}} f_b \cdot \Big(b(i) \cdot (1 - g_b) + (1 - b(i)) \cdot g_b\Big).$$

We prove this by considering the following cases:

- **When $i \in [k]$:** We have:

$$\sum_{b:[m]\to\{0,1\}} \frac{1}{2^m} \cdot \Big(b(i) \cdot (1 - g'_b) + (1 - b(i)) \cdot g'_b\Big)$$

$$= \sum_{b^*:[k]\to\{0,1\}} \sum_{\substack{b:[m]\to\{0,1\}\\ b|_{[k]}=b^*}} \frac{1}{2^m} \cdot \Big(b^*(i)\cdot(1-g'_b) + (1-b^*(i))\cdot g'_b\Big) \qquad \text{(As } i \in [k]\text{)}$$

$$= \sum_{b^*:[k]\to\{0,1\}} \frac{1}{2^k} \cdot \Big(b^*(i)\cdot(1-g'_{b^*}) + (1-b^*(i))\cdot g'_{b^*}\Big)$$

(As g'_b is determined by $b|_{[k]}$)

$$\leq \sum_{b^*:[k]\to\{0,1\}} \left(\sum_{\substack{b:[m]\to\{0,1\}\\ b|_{[k]}=b^*}} f_b + \epsilon'\right) \cdot \Big(b^*(i)\cdot(1-g'_{b^*}) + (1-b^*(i))\cdot g'_{b^*}\Big)$$

(C is (k,ϵ')-random, Definition 9.2)

$$\leq \frac{\epsilon}{10} + \sum_{b^*:[k]\to\{0,1\}} \left(\sum_{\substack{b:[m]\to\{0,1\}\\ b|_{[k]}=b^*}} f_b\right) \cdot \Big(b^*(i)\cdot(1-g'_{b^*}) + (1-b^*(i))\cdot g'_{b^*}\Big)$$

(As $\epsilon' = \frac{\epsilon}{10\cdot 2^k}$)

$$\leq \frac{\epsilon}{10} + \sum_{b^*:[k]\to\{0,1\}} \left(\sum_{\substack{b:[m]\to\{0,1\}\\ b|_{[k]}=b^*}} f_b \cdot \Big(b^*(i)\cdot(1-g_b) + (1-b^*(i))\cdot g_b\Big)\right)$$

(Eq. (21))

$$= \frac{\epsilon}{10} + \sum_{b:[m]\to\{0,1\}} f_b \cdot \Big(b(i)\cdot(1-g_b) + (1-b(i))\cdot g_b\Big).$$

- **When $i \in [m]\setminus[k]$:** Recall that g'_b is determined by $b|_{[k]}$. As $i \in [m]\setminus[k]$, this means that for all $b:[m]\to\{0,1\}$, we have $g'_b = g'_{\tilde{b}}$ where $\tilde{b}$ is defined to be the same b except that the i-th coordinate is flipped. Re-parametrizing the sum writing $\tilde{b}$ for b and using $g'_b = g'_{\tilde{b}}$, we get:

  $$\sum_{b:[m]\to\{0,1\}} \frac{1}{2^m}\cdot\Big(b(i)\cdot(1-g'_b)+(1-b(i))\cdot g'_b\Big) = \sum_{b:[m]\to\{0,1\}} \frac{1}{2^m}\cdot\Big((1-b(i))\cdot(1-g'_b)+b(i)\cdot g'_b\Big).$$

  When two quantities are equal, the are both equal to the average. This gives:

  $$\sum_{b:[m]\to\{0,1\}} \frac{1}{2^m}\cdot\Big(b(i)\cdot(1-g'_b)+(1-b(i))\cdot g'_b\Big) = \sum_{b:[m]\to\{0,1\}} \frac{1}{2^m}\cdot\frac{1}{2} = \frac{1}{2}.$$

  Next, as $i \in [m]\setminus[k]$, we have by Eq. (20) that:

  $$\sum_{b:[m]\to\{0,1\}} \frac{1}{2^m}\cdot\Big(b(i)\cdot(1-g'_b)+(1-b(i))\cdot g'_b\Big) \leq \frac{\epsilon}{4} + \frac{\Delta(C(i),\tau_1)}{L_1}.$$

Recall from our definition of f and g that for all $b : [m] \to \{0,1\}$, the value $f_b \cdot L_1$ is the number of coordinates j where Alice's code satisfies $C_j(i') = b(i')$ for all $i' \in [m]$. Furthermore, g_b is the fraction of these $f_b \cdot L_1$ coordinates where Bob receives 1. Thus, we get:

$$\Delta(C(i), \tau_1) = \sum_{b:[m]\to\{0,1\}} f_b \cdot L_1 \cdot \Big(b(i) \cdot (1 - g_b) + (1 - b(i)) \cdot g_b \Big).$$

Combining the last two equations finishes the proof.

□

10 Converse of Theorem 9.1

We now show the "only if" direction of Theorem 1.4. This is formalized as:

Theorem 10.1. *Assume that for all $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a protocol (not necessarily constant rate) for message transfer with one round of feedback, input length n, and resilience $\frac{7}{23} - \epsilon$ over the binary corruption channel. Then, Conjecture 1.3 holds.*

We prove Theorem 10.1 in the following equivalent form, whose equivalence is due to Lemma 8.1:

Theorem 10.2. *Assume that for all $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a protocol (not necessarily constant rate) for message transfer with one round of feedback, input length n, and resilience $\frac{7}{23} - \epsilon$ over the binary corruption channel. Then, for all $m > 0$, all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:*

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}\Big(\tfrac{\overrightarrow{1}}{2^m}, g\Big).$$

The proof of Theorem 10.2 spans the rest of this section.

10.1 Proving a Weaker Version of Theorem 10.2

In this section, we show a weaker version of Theorem 10.2 that allows for a general f instead of $f = \frac{\overrightarrow{1}}{2^m}$.

Lemma 10.3. *Assume that for all $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a protocol (not necessarily constant rate) for message transfer with one round of feedback, input length n, and resilience $\frac{7}{23} - \epsilon$ over the binary corruption channel. Then, for all $m > 0$, there exists $f \in \Delta^{2^m-1}$ such that for all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:*

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}(f, g).$$

Roughly speaking, the f is Alice's first message in the protocol when viewed as a vector (see Section 8) and any g corresponds to the action of an adversary on f. Then, the fact that the protocol works will mean that there exists a message that Alice can send after receiving feedback from Bob that allows Bob to output correctly. This message will correspond to the desired h. We formalize this below.

Proof of Lemma 10.3. Fix $m > 0$. For any $\epsilon > 0$, we pick $m' > m$ to be large enough such that $\mathsf{d}_{\mathsf{corr}}(m', 3) \leq \frac{1}{4} + \epsilon$ and $\mathsf{d}_{\mathsf{corr}}(m', 5) \leq \frac{5}{16} + \epsilon$ both hold. Such an m' exists because of Lemma 4.5. Let Π be the protocol corresponding to ϵ and m' as promised by the assumption in Lemma 10.3, and let L_1 and L_2 be the lengths of the rounds in Π. By increasing the chosen value of m', we can assume without loss of generality that $L_1 \geq \frac{2^{m+1}}{\epsilon}$. We first claim that:

Claim 10.4. *It holds that:*

$$\frac{8}{23} - 300\epsilon \leq \frac{L_1}{L_1 + L_2} \leq \frac{8}{23} + 40\epsilon.$$

Proof. This proof roughly follows the arguments in Section 7. Consider the following attacks on the protocol for $k \in \{3, 5\}$: The adversary corrupts Alice's message in the first round to a pattern τ such that there exist k inputs for Alice whose encodings are all within distance $(\mathsf{d}_{\mathsf{corr}}(m', k) + \epsilon) \cdot L_1$ of τ. Note that such a τ exists by definition of $\mathsf{d}_{\mathsf{corr}}(m', k)$. During the feedback round, even if Alice and Bob can somehow agree on these k codewords, and even if the distances are exactly $(\mathsf{d}_{\mathsf{corr}}(m', k) + \epsilon) \cdot L_1$, there will be two inputs out of the k that the adversary can corrupt to the same message in the second round using at most $(\mathsf{d}_{\mathsf{corr}}(k, 2) + \epsilon) \cdot L_2$ corruptions. When this happens, the protocol fails and therefore, we must have, for $k \in \{3, 5\}$:

$$\left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) \leq (\mathsf{d}_{\mathsf{corr}}(m', k) + \epsilon) \cdot L_1 + (\mathsf{d}_{\mathsf{corr}}(k, 2) + \epsilon) \cdot L_2.$$

Using our choice of m', this means

$$\left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) \leq \min\Big(\left(\tfrac{1}{4} + 2\epsilon\right) \cdot L_1 + \left(\tfrac{1}{3} + \epsilon\right) \cdot L_2, \left(\tfrac{5}{16} + 2\epsilon\right) \cdot L_1 + \left(\tfrac{3}{10} + \epsilon\right) \cdot L_2\Big).$$

From the above inequalities, we get the claim. $\square$

Henceforth, we consider Π restricting attention to only the first m inputs for Alice, *i.e.* inputs in $[m]$. Recall the vector interpretation from Section 8 and let $f \in \Delta^{2^m - 1}$ correspond to the first message sent by Alice in Π. Next, fix an arbitrary $g \in [0, 1]^{2^m}$. We claim that there exists an adversary for Π such that, for all $i \neq i' \in [m]$, its corruptions for the inputs i and i' in the first round add up to at most $\big(\mathsf{D}(f, g)_{\{i, i'\}} + \epsilon\big) \cdot L_1$ and the message received by Bob is the same for all inputs in $[m]$. Indeed, for $b : [m] \to \{0, 1\}$, the adversary corrupts the first $\lfloor f_b g_b L_1 \rfloor$ coordinates "matching" b to bit 1 and the remaining $f_b L_1 - \lfloor f_b g_b L_1 \rfloor$ coordinates "matching" b to bit 0 (both regardless of the input). Observe that in Eq. (12), the summand corresponding to each b is offset by at most $\frac{2}{L_1}$ due to rounding in constructing the adversary above. Summing over all $b : [m] \to \{0,1\}$, we then get that the total corruptions for i, i' is at most

$$\left(\mathsf{D}(f,g)_{\{i,i'\}} + 2^m \cdot \frac{2}{L_1}\right) \cdot L_1 \leq \left(\mathsf{D}(f,g)_{\{i,i'\}} + \epsilon\right) \cdot L_1$$

by our assumption that $L_1 \geq \frac{2^{m+1}}{\epsilon}$. This holds for all $i \neq i' \in [m]$. Now we claim that:

Claim 10.5. *There exists $h \in \Delta^{2^m-1}$ such that the following holds for all $i \neq i' \in [m]$:*

$$2 \cdot \left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) \leq \left(\mathsf{D}(f,g)_{\{i,i'\}} + \epsilon\right) \cdot L_1 + \left(\mathsf{D}(h)_{\{i,i'\}} + \epsilon\right) \cdot L_2. \tag{22}$$

Proof. We prove the claim by contradiction. Let $h \in \Delta^{2^m-1}$ be the vector corresponding to the code Alice uses in the second round for the adversary above. Suppose for the purpose of contradiction that there exists some $i \neq i' \in [m]$ violating Eq. (22). That is,

$$2 \cdot \left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) > \left(\mathsf{D}(f,g)_{\{i,i'\}} + \epsilon\right) \cdot L_1 + \left(\mathsf{D}(h)_{\{i,i'\}} + \epsilon\right) \cdot L_2. \tag{23}$$

Consider the adversary constructed above. By Eq. (12) and the above discussion about rounding, the number of corruptions for i, i' are upper bounded by

$$c = \left(\frac{\epsilon}{2} + \sum_{b:[m]\to\{0,1\}} f_b \cdot (b(i) \cdot (1 - g_b) + (1 - b(i)) \cdot g_b)\right) \cdot L_1$$

and

$$c' = \left(\frac{\epsilon}{2} + \sum_{b:[m]\to\{0,1\}} f_b \cdot (b(i') \cdot (1 - g_b) + (1 - b(i')) \cdot g_b)\right) \cdot L_1$$

respectively. Note that $c + c' = \left(\mathsf{D}(f,g)_{\{i,i'\}} + \epsilon\right) \cdot L_1$. Now there are two cases.

- **If** $c, c' \leq \left(\frac{7}{23} - \epsilon\right) \cdot (L_1 + L_2)$**:** By Eq. (11), $\mathsf{D}(h)_{\{i,i'\}} \cdot L_2$ is exactly the Hamming distance between the encodings of i, i' in the second round. Therefore, the adversary can simply corrupt them to the same message in the second round such that the number of corruptions is at most $\left(\frac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) - c$ for i and is at most $\left(\frac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) - c'$ for i'. This is always possible by Eq. (23). As a result, the adversary is able to confuse Bob between i and i' after all two rounds of communication with no more than $\left(\frac{7}{23} - \epsilon\right) \cdot (L_1 + L_2)$ corruptions for both i and i'. This contradicts the protocol having resilience $\frac{7}{23} - \epsilon$.
- **Otherwise:** By our argument above, no pair of i, i' violating Eq. (22) can be of the first case. In other words, any pair of i, i' violating Eq. (22) has at least one of them

fall into the subset $\Gamma \subseteq [m]$ containing all $i'' \in [m]$ such that

$$\left(\frac{\epsilon}{2} + \sum_{b:[m]\to\{0,1\}} f_b \cdot (b(i'') \cdot (1 - g_b) + (1 - b(i'')) \cdot g_b)\right) \cdot L_1 > \left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2).$$

That is, the number of corruptions by the constructed adversary already exceeds the budget for $i'' \in \Gamma$. In such scenario, our argument for the first case may not work. Instead, for any fixed $i'' \in \Gamma$, we show how to adjust h slightly so that Eq. (22) is satisfied by all pairs containing i'', without affecting all other pairs. The claim follows by repeatedly applying the following argument to all $i'' \in \Gamma$.

Fix $i'' \in \Gamma$. We construct a new $h' \in \Delta^{2^m-1}$ from h as follows. For all $b : [m] \to \{0, 1\}$, let

$$h'_b = \frac{1}{2} \cdot \sum_{\substack{b':[m]\to\{0,1\} \\ b'|_{[m]\setminus\{i''\}}=b|_{[m]\setminus\{i''\}}}} h_{b'}.$$

At a high level, h' is just an "averaged" version of h where fixing the encoding bit for all input other than i'', the encoding bit of i'' has an equal probability of being 0 or 1. As a result, by Eq. (11), $\mathsf{D}(h')_{\{i,i''\}} = \frac{1}{2}$ for all $i \neq i''$ while $\mathsf{D}(h')_{\{i,i'\}} = \mathsf{D}(h)_{\{i,i'\}}$ for all pairs of i, i' not containing i''. Overall, since

$$\left(\tfrac{7}{23} - \epsilon\right) \cdot (L_1 + L_2) \leq \frac{L_2}{2},$$

we then have that Eq. (22) is satisfied by all pairs containing i'', with h replaced by h', while all other pairs are unaffected. This concludes the proof.

$\square$

Now from Claims 10.4 and 10.5, we get an h such that, for all $i \neq i' \in [m]$:

$$(14 - 69\epsilon) \leq \mathsf{D}(f, g)_{\{i,i'\}} \cdot (8 + 1000\epsilon) + \mathsf{D}(h)_{\{i,i'\}} \cdot (15 + 7000\epsilon)$$

This implies:

$$\min_{\{i,i'\}\in\binom{[m]}{2}} 8 \cdot \mathsf{D}(f, g)_{\{i,i'\}} + 15 \cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14 - 10^4 \cdot \epsilon.$$

Since $\epsilon > 0$ was arbitrary, this means that:

$$\sup_{f\in\Delta^{2^m-1}} \inf_{g\in[0,1]^{2^m}} \sup_{h\in\Delta^{2^m-1}} \min_{\{i,i'\}\in\binom{[m]}{2}} 8 \cdot \mathsf{D}(f, g)_{\{i,i'\}} + 15 \cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14.$$

Note that for any fixed $f \in \Delta^{2^m-1}$ and $g \in [0, 1]^{2^m}$, $\min_{\{i,i'\}\in\binom{[m]}{2}} 8 \cdot \mathsf{D}(f, g)_{\{i,i'\}} + 15 \cdot \mathsf{D}(h)_{\{i,i'\}}$ is a continuous real function in h over Δ^{2^m-1}, which is compact. By the extreme value theorem, its supremum over Δ^{2^m-1} is always attained. Applying similar arguments to g and

f as well, we get:

$$\max_{f \in \Delta^{2^m-1}} \min_{g \in [0,1]^{2^m}} \max_{h \in \Delta^{2^m-1}} \min_{\{i,i'\} \in \binom{[m]}{2}} 8 \cdot \mathsf{D}(f,g)_{\{i,i'\}} + 15 \cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14,$$

which proves Lemma 10.3. □

10.2 Proof of Theorem 10.2

Note that the only difference between Lemma 10.3 and Theorem 10.2 is that the former only promises that there exists a suitable f while the latter guarantees that $f = \frac{\vec{1}}{2^m}$. The high level plan to show this stronger guarantee is to take an arbitrary f from Lemma 10.3 and progressively convert it to look more and more like $f = \frac{\vec{1}}{2^m}$. Specifically, noting that $f = \frac{\vec{1}}{2^m}$ corresponds (in expectation) to a uniformly random code in the vectors interpretation of Section 8, we will in each step convert f from a (k, ϵ')-random code to a $(k+1, \epsilon)$ random code (for some k and $\epsilon' < \epsilon$), and when k becomes large enough, show that it can be replaced by a prefect random code (corresponding to $f = \frac{\vec{1}}{2^m}$).

We start by recasting Definition 9.2 in terms of f. Let $m > 0$ and $f \in \Delta^{2^m-1}$. For all sets $\Gamma \subseteq [m]$ and functions $b^* : \Gamma \to \{0,1\}$, we define $f_{b^*}[\Gamma] = \sum_{b:[m]\to\{0,1\}, b|_\Gamma = b^*} f_b$.

Definition 10.6 (Definition 9.2 in the vectors interpretation)**.** *Let $m, k \geq 1$ and $\epsilon \geq 0$. We say that $f \in \Delta^{2^m-1}$ is (k, ϵ)-random if for all subsets $\Gamma \subseteq [m]$ of size at most k and all $b^* : \Gamma \to \{0,1\}$, we have:*

$$\left| f_{b^*}[\Gamma] - \frac{1}{2^{|\Gamma|}} \right| \leq \epsilon.$$

Lemma 10.7. *Assume that for all $\epsilon > 0$ and $n \in \mathbb{N}$, there exists a protocol (not necessarily constant rate) for message transfer with one round of feedback, input length n, and resilience $\frac{7}{23} - \epsilon$ over the binary corruption channel. Then, for all $k \in \mathbb{N}$, $\epsilon' > 0$, and $m \geq k \in \mathbb{N}$, there exists $f \in \Delta^{2^m-1}$ that is (k, ϵ')-random such that for all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:*

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15} \cdot \mathsf{D}(f,g).$$

Before showing Lemma 10.7, we show why it implies Theorem 10.2.

Proof of Theorem 10.2. Let $m \in \mathbb{N}$ and $\delta > 0$ be arbitrary. Applying Lemma 10.7 with $k = m$ and $\epsilon' = \delta/2^m$ gives us $f \in \Delta^{2^m-1}$ that is $(m, \delta/2^m)$-random and satisfies the condition in Lemma 10.7. By Definition 10.6, we have for all $b : [m] \to \{0,1\}$ that $\left| f_b - \frac{1}{2^m} \right| \leq \delta/2^m$. In turn, by Eq. (12), this means that for all $g \in [0,1]^{2^m}$, we have $\mathsf{D}(f,g) \leq \mathsf{D}\left(\frac{\vec{1}}{2^m}, g\right) + 2 \cdot \vec{\delta}$. Using this and our choice of f, we get that for all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that:

$$\min_{\{i,i'\} \in \binom{[m]}{2}} 8 \cdot \mathsf{D}(f,g)_{\{i,i'\}} + 15 \cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14 - 2\delta.$$

As $\delta > 0$ was arbitrary, it follows that:

$$\inf_{g\in[0,1]^{2^m}} \sup_{h\in\Delta^{2^m-1}} \min_{\{i,i'\}\in\binom{[m]}{2}} 8\cdot \mathsf{D}(f,g)_{\{i,i'\}} + 15\cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14.$$

Note that for any fixed $g \in [0,1]^{2^m}$, $\min_{\{i,i'\}\in\binom{[m]}{2}} 8\cdot \mathsf{D}(f,g)_{\{i,i'\}} + 15\cdot \mathsf{D}(h)_{\{i,i'\}}$ is a continuous real function in h over Δ^{2^m-1}, which is compact. By the extreme value theorem, its supremum over Δ^{2^m-1} is always attained. Applying similar arguments to g as well, we get:

$$\min_{g\in[0,1]^{2^m}} \max_{h\in\Delta^{2^m-1}} \min_{\{i,i'\}\in\binom{[m]}{2}} 8\cdot \mathsf{D}(f,g)_{\{i,i'\}} + 15\cdot \mathsf{D}(h)_{\{i,i'\}} \geq 14.$$

Theorem 10.2 follows. □

It remains to show Lemma 10.7.

Proof of Lemma 10.7. We prove the lemma by induction on k. First, we show the base case $k = 1$.

Base case. In this case, we shall actually show the the lemma holds even when $\epsilon' = 0$. Fix $m \in \mathbb{N}$ and let $f \in \Delta^{2^m-1}$ be as promised by Lemma 10.3. Define $f' \in \Delta^{2^m-1}$ to be such that $f'_b = f_{\overline{b}}$ for all $b : [m] \to \{0,1\}$, where $\overline{b} : [m] \to \{0,1\}$ is the function satisfying $\overline{b}(i) = 1 - b(i)$ for all $i \in [m]$. We will show that Lemma 10.7 holds with $f^* = \frac{f+f'}{2}$, which is indeed $(1,0)$-random. For this, we fix $g \in [0,1]^{2^m}$ and define $g' \in [0,1]^{2^m}$ to be such that $g'_b = 1 - g_b$ for all $b : [m] \to \{0,1\}$. Then by our choice of f, there exists $h, h' \in \Delta^{2^m-1}$ such that

$$\mathsf{D}(h) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15}\cdot \mathsf{D}(f,g) \quad \text{and} \quad \mathsf{D}(h') \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15}\cdot \mathsf{D}(f,g').$$

Since $\mathsf{D}(f,g)$ is linear in f and $\mathsf{D}(h)$ is linear in h, we now get:

$$\mathsf{D}\big(\tfrac{h+h'}{2}\big) \geq \frac{\overrightarrow{14}}{15} - \frac{8}{15}\cdot \mathsf{D}(f^*,g'),$$

as desired.

Inductive case. We now fix $k \geq 1$ and show Lemma 10.7 for $k+1$ assuming it holds for k. Fix $\epsilon' > 0$, and $m \geq k+1 \in \mathbb{N}$. We apply the induction hypothesis with $\epsilon'' = \epsilon'/(k+2)$ and $m' = 2(R+1)\cdot m^{2k}$, where $R = \mathsf{R}(3, \frac{1}{\epsilon''^2}+1)$ and $\mathsf{R}(\cdot)$ denotes the Ramsey number as in Section 3. By the induction hypothesis, there exists an $f' \in \Delta^{2^{m'}-1}$ that is (k,ϵ'')-random and satisfies the induction hypothesis.

We say that a set $\Gamma \in \binom{[m']}{k+1}$ is irregular if there exists $b^* : \Gamma \to \{0,1\}$ such that $\left|f'_{b^*}[\Gamma] - \frac{1}{2^{k+1}}\right| > \epsilon'$, and call it regular otherwise. Consider now the following algorithm that uses f' to construct a $(k+1,\epsilon')$-random $f \in \Delta^{2^m-1}$.

Claim 10.8. *Algorithm 3 outputs $f \in \Delta^{2^m-1}$ that is $(k+1,\epsilon')$-random.*

Algorithm 3 A procedure for finding $(k+1,\epsilon')$-random $f \in \Delta^{2^m-1}$ given f'.

Input: A (k,ϵ'')-random $f' \in \Delta^{2^{m'}-1}$.
Output: A $(k+1,\epsilon')$-random $f \in \Delta^{2^m-1}$.

1: Let $\Gamma = [k]$ and $\tilde{\Gamma} = [m'] \setminus [k]$.
2: **for** $i \in [m-k]$ **do**
3: $\quad\tilde{\Gamma} \longleftarrow \{x \in \tilde{\Gamma} \mid \forall X \in \binom{\Gamma}{k} : X \cup \{x\} \text{ is regular}\}$.
4: $\quad\Gamma \longleftarrow \Gamma \cup \{\min(\tilde{\Gamma})\}$, aborting if $\tilde{\Gamma} = \emptyset$.
5: $\quad\tilde{\Gamma} \longleftarrow \tilde{\Gamma} \setminus \{\min(\tilde{\Gamma})\}$.
6: **end for**
7: Output $f = f'[\Gamma]$.

We prove Claim 10.8 but assuming it for now, we can finish the proof of Lemma 10.7. As Claim 10.8 guarantees that f is $(k+1,\epsilon')$-random, all that remains to be shown is that for all $g \in [0,1]^{2^m}$, there exists $h \in \Delta^{2^m-1}$ such that $\mathsf{D}(h) \geq \frac{\vec{14}}{15} - \frac{8}{15} \cdot \mathsf{D}(f,g)$. This essentially follows from the fact that $f = f'[\Gamma]$ for some set $\Gamma \subseteq [m']$ (see Line 7). We flesh out the details next.

Fix an arbitrary $g \in [0,1]^{2^m}$. Define $g' \in [0,1]^{2^{m'}}$ to be such that for all $b' : [m'] \to \{0,1\}$, we have $g'_{b'} = g_{b'|_\Gamma}$. By the induction hypothesis, there exists $h' \in \Delta^{2^{m'}-1}$ such that $\mathsf{D}(h') \geq \frac{\vec{14}}{15} - \frac{8}{15} \cdot \mathsf{D}(f',g')$. Define $h \in \Delta^{2^m-1}$ to be such that for all $b \in [m] \to \{0,1\}$, we have $h_b = h'_b[\Gamma]$. Observe from Eqs. (11) and (12) that, for all $i \neq i' \in \Gamma$, we have:

$$\mathsf{D}(h)_{\{i,i'\}} = \mathsf{D}(h')_{\{i,i'\}} \geq \frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}(f',g')_{\{i,i'\}} = \frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}(f,g)_{\{i,i'\}},$$

as desired. □

Proof of Claim 10.8. Assume for now that the algorithm never aborts in Line 4. Under this assumption, note that Γ increases by 1 in every iteration of Line 2 and thus, has size m at the end. This means that the output f satisfies $f \in \Delta^{2^m-1}$. Suppose for the sake of contradiction that f is not $(k+1,\epsilon')$-random and let $\Gamma^* \subseteq \Gamma$ be a set violating Definition 10.6. Note that $|\Gamma^*| = k+1$, as otherwise $f = f'[\Gamma]$ implies that Γ^* would also violate Definition 10.6 for f'. Observe that $|\Gamma^*| = k+1$ and the fact that Γ^* violates Definition 10.6 implies that Γ^* is an irregular $(k+1)$-sized subset of Γ. We derive a contradiction by showing that all $(k+1)$-sized subsets of Γ are regular. Indeed, this holds at the beginning of the algorithm (as there are no such subsets) and whenever a new element is added Γ, Line 3 ensures that any $k+1$ sized subset containing the new element is regular.

It remains to show that the algorithm never aborts. Note that $m' = 2(R+1) \cdot m^{2k} > 2R \cdot m^{2k} + m + k$ and at the start of the algorithm, the set $\tilde{\Gamma}$ has size $m' - k$. Moreover, at most m elements are removed from $\tilde{\Gamma}$ in Line 5 (one in each iteration). Thus, in order to show that the algorithm never aborts, it suffices to show that for any iteration $i \in [m-k]$, at most $2R \cdot m^k$ elements are removed from $\tilde{\Gamma}$ in Line 3. We fix an $i \in [m-k]$ and show this next.

Let $\tilde{\Gamma}_o, \Gamma_o$ be the "old" values of $\tilde{\Gamma}$ and Γ respectively before Line 3 is executed in iteration i and $\tilde{\Gamma}_n$ be the "new" value of $\tilde{\Gamma}$ after Line 3 is executed in iteration i (Line 3 does not change Γ). Suppose of the sake of contradiction that $|\tilde{\Gamma}_o \setminus \tilde{\Gamma}_n| > 2R \cdot m^k$. By Line 3, for every $x \in \tilde{\Gamma}_o \setminus \tilde{\Gamma}_n$, there exists $X \in \binom{\Gamma_o}{k}$ such that $X \cup \{x\}$ is irregular. As $|\Gamma_o| \leq m$, there are at most m^k such X and thus, for some choice of X, there exist $> 2R$ values $x \in \tilde{\Gamma}_o \setminus \tilde{\Gamma}_n$ such that $X \cup \{x\}$ is irregular. Fix X to be this value. The following claim essentially categorizes any irregular set into two categories (determined by z in the claim):

Claim 10.9. *For all irregular $\Gamma \in \binom{[m']}{k+1}$, there exists $z \in \{0,1\}$ such that for all $b' : \Gamma \to \{0,1\}$, we have:*

$$\begin{cases} f'_{b'}[\Gamma] - \frac{1}{2^{k+1}} > \epsilon'', & \textit{if } |\{i \in \Gamma \mid b'(i) = 1\}| + z \textit{ is even} \\ f'_{b'}[\Gamma] - \frac{1}{2^{k+1}} < -\epsilon'', & \textit{if } |\{i \in \Gamma \mid b'(i) = 1\}| + z \textit{ is odd} \end{cases}.$$

Proof. Fix Γ. As Γ is irregular, we have $b^* : \Gamma \to \{0,1\}$ such that $\left| f'_{b^*}[\Gamma] - \frac{1}{2^{k+1}} \right| > \epsilon'$. Pick the smallest such b^*. If $f'_{b^*}[\Gamma] > \frac{1}{2^{k+1}}$, define $z = |\{i \in \Gamma \mid b^*(i) = 1\}| \bmod 2$. Otherwise, define $z = 1 + |\{i \in \Gamma \mid b^*(i) = 1\}| \bmod 2$. For $b, b' : \Gamma \to \{0,1\}$, define $\Delta(b, b')$ to be the number of coordinates where b and b' differ. We will actually show the stronger statement that, for all $b' : \Gamma \to \{0,1\}$, we have:

$$\begin{cases} f'_{b'}[\Gamma] - \frac{1}{2^{k+1}} > (k + 1 - \Delta(b', b^*)) \cdot \epsilon'', & \text{if } \Delta(b', b^*) \text{ is even} \\ f'_{b'}[\Gamma] - \frac{1}{2^{k+1}} < -(k + 1 - \Delta(b', b^*)) \cdot \epsilon'', & \text{if } \Delta(b', b^*) \text{ is odd} \end{cases}.$$

The base case $\Delta(b', b^*) = 0$ is trivial as it implies $b' = b^*$. We now fix $d > 0$ and show the result for all b' satisfying $\Delta(b', b^*) = d$ assuming it holds for all b' satisfying $\Delta(b', b^*) = d - 1$. We assume without loss of generality that d is odd and the case when d is even is analogous. Fix b' satisfying $\Delta(b', b^*) = d$ and let b'' be such that $\Delta(b'', b^*) = d - 1$ and $\Delta(b', b'') = 1$ (such a b'' always exists). Let $x \in \Gamma$ be the unique entry such that $b'(x) \neq b''(x)$. By our induction hypothesis, we have $f'_{b''}[\Gamma] - \frac{1}{2^{k+1}} > (k + 2 - d) \cdot \epsilon''$. As f' is (k, ϵ'')-random, we also have $\left| f'_{b'}[\Gamma] + f'_{b''}[\Gamma] - \frac{1}{2^k} \right| \leq \epsilon''$. This is only possible if $f'_{b'}[\Gamma] - \frac{1}{2^{k+1}} < -(k + 1 - d) \cdot \epsilon''$, as desired. □

From Claim 10.9, we conclude that there exists $z \in \{0,1\}$ such that for $> R$ values $x \in \tilde{\Gamma}_o \setminus \tilde{\Gamma}_n$, we have that $X \cup \{x\}$ is irregular and satisfies Claim 10.9 with z. Fix this z and let $x_1, \ldots, x_R$ be the R smallest values of x. Define a graph over $x_1, \ldots, x_R$ where, for all $i \neq i' \in [R]$, the vertices x_i and $x_{i'}$ are connected if and only if $\mathsf{D}(f')_{\{x_i, x_{i'}\}} < \frac{1}{2}$. By definition of R the graph either has a triangle, or a large independent set of size at least $\frac{1}{\epsilon''^2} + 1$. We derive a contradiction in both cases:

- **When the graph has a triangle:** Without loss of generality, assume the triangle consists of the vertices x_1, x_2, x_3. Define $g' \in [0,1]^{2^{m'}}$ to be such that for all $b' : [m'] \to \{0,1\}$, we have $g'_{b'} = \mathsf{Maj}(b'(x_1), b'(x_2), b'(x_3))$, where Maj denote the majority function over 3 bits. With this definition, we have $g'_{b'} \in \{0,1\}$ and therefore, Eq. (12) says that, for all $i \neq i' \in [3]$, we have:

$$\begin{aligned}
\mathsf{D}(f', g')_{\{x_i, x_{i'}\}} &= \sum_{x' \in \{x_i, x_{i'}\}} \sum_{b':[m'] \to \{0,1\}} f'_{b'} \cdot \mathbb{1}(b'(x') \neq g'_{b'}) \\
&= \sum_{b':[m'] \to \{0,1\}} \sum_{x' \in \{x_i, x_{i'}\}} \mathbb{1}(b'(x_i) \neq b'(x_{i'})) f'_{b'} \cdot \mathbb{1}(b'(x') \neq g'_{b'}) \\
& \qquad\qquad (\text{As } b'(x_i) = b'(x_{i'}) \implies b'(x_i) = g'_{b'}) \\
&= \sum_{b':[m'] \to \{0,1\}} \mathbb{1}(b'(x_i) \neq b'(x_{i'})) f'_{b'} \\
&> \frac{1}{2}. \qquad\qquad (\text{Eq. (11) and } \mathsf{D}(f')_{\{x_i, x_{i'}\}} < \tfrac{1}{2})
\end{aligned}$$

By the induction hypothesis, there exists $h' \in \Delta^{2^{m'}-1}$ such that, for all $i \neq i' \in [3]$, we have:

$$\mathsf{D}(h')_{\{x_i, x_{i'}\}} \geq \frac{14}{15} - \frac{8}{15} \cdot \mathsf{D}(f', g')_{\{x_i, x_{i'}\}} > \frac{2}{3}.$$

However, using the same concentration arguments as in the proof of Lemma 9.5, this means that h' can be used to sample a code $C : [3] \to \{0,1\}^L$ (for some large enough L) such that is less-than-2-list decodable for corruptions up to radius that is strictly larger than $\frac{1}{3}$. This contradicts Lemma 4.5.

- **When the graph has a large independent set:** Define $t = \frac{1}{\epsilon''^2} + 1$ for convenience and assume without loss of generality that the independent sets consists of the vertices $x_1, \ldots, x_t$. Define $X' = \{x_1, \ldots, x_t\}$ for convenience. As the vertices in X' form an independent set, we have by Eq. (11) that:

$$\frac{\binom{t}{2}}{2} \leq \sum_{i < i' \in [t]} \mathsf{D}(f')_{\{x_i, x_{i'}\}} = \sum_{i < i' \in [t]} \sum_{b':[m'] \to \{0,1\}} \mathbb{1}(b'(x_i) \neq b'(x_{i'})) \cdot f'_{b'}.$$

For $b^* : X' \to \{0,1\}$, we define $\alpha_{b^*} = \frac{1}{t} \cdot \sum_{i \in [t]} \mathbb{1}(b^*(x_i) = 0)$ and we get:

$$\frac{t-1}{4t} \leq \sum_{b':[m'] \to \{0,1\}} f'_{b'} \cdot \alpha_{b'|_{X'}} \cdot \left(1 - \alpha_{b'|_{X'}}\right) = \sum_{b^*:X' \to \{0,1\}} f'_{b^*}[X'] \cdot \alpha_{b^*} \cdot (1 - \alpha_{b^*}).$$

Next, for functions $b^* : X' \to \{0,1\}$ and $\hat{b} : X \to \{0,1\}$, we will use $\hat{b} \diamond b^*$ to denote the function mapping $X \cup X' \to \{0,1\}$ that, on input $x \in X \cup X'$, outputs $b^*(x)$ if $x \in X'$ and $\hat{b}(x)$ if $x \in X$. We get:

$$\frac{t-1}{4t} \leq \sum_{\hat{b}:X \to \{0,1\}} \sum_{b^*:X' \to \{0,1\}} f'_{\hat{b} \diamond b^*}[X \cup X'] \cdot \alpha_{b^*} \cdot (1 - \alpha_{b^*}).$$

Next, for $\hat{b} : X \to \{0,1\}$, define $\beta_{\hat{b}} = \sum_{b^*:X' \to \{0,1\}} \frac{f'_{\hat{b} \diamond b^*}[X \cup X'] \cdot \alpha_{b^*}}{f'_{\hat{b}}[X]}$. As $F(u) = u(1-u)$ is concave and $t > \frac{1}{\epsilon''^2}$, we get:

$$\frac{1-\epsilon''^2}{4} < \frac{t-1}{4t} \leq \sum_{\hat{b}:X \to \{0,1\}} f'_{\hat{b}}[X] \cdot \beta_{\hat{b}} \cdot (1 - \beta_{\hat{b}}).$$

Thus, to derive a contradiction, it suffices to show that for all $\hat{b} : X \to \{0,1\}$, we either have $\beta_{\hat{b}} \leq \frac{1-\epsilon''}{2}$ or we have $\beta_{\hat{b}} \geq \frac{1+\epsilon''}{2}$. We do this next, fixing an arbitrary $\hat{b} : X \to \{0,1\}$ such that $|\{i \in X \mid \hat{b}(i) = 1\}| = z \bmod 2$. The other case $|\{i \in X \mid \hat{b}(i) = 1\}| = 1 - z \bmod 2$ can be proved similarly. Using our definitions of α_{b^*} and $\beta_{\hat{b}}$, we have:

$$\beta_{\hat{b}} = \frac{1}{t} \cdot \sum_{i \in [t]} \sum_{b^*:X' \to \{0,1\}} \frac{f'_{\hat{b} \diamond b^*}[X \cup X'] \cdot \mathbb{1}(b^*(x_i) = 0)}{f'_{\hat{b}}[X]}$$

For $i \in [t]$, we use $\hat{b} \diamond (x_i = 0)$ to denote the function mapping $X \cup \{x_i\} \to \{0,1\}$ that takes the value 0 on x_i and the value given by $\hat{b}(\cdot)$ on inputs in X. The notation $\hat{b} \diamond (x_i = 1)$ is defined similarly. We get:

$$\begin{aligned}
\beta_{\hat{b}} &= \frac{1}{t} \cdot \sum_{i \in [t]} \frac{f'_{\hat{b} \diamond (x_i = 0)}[X \cup \{x_i\}]}{f'_{\hat{b} \diamond (x_i = 0)}[X \cup \{x_i\}] + f'_{\hat{b} \diamond (x_i = 1)}[X \cup \{x_i\}]} \\
&\geq \frac{1}{t} \cdot \sum_{i \in [t]} \frac{f'_{\hat{b} \diamond (x_i = 0)}[X \cup \{x_i\}]}{f'_{\hat{b} \diamond (x_i = 0)}[X \cup \{x_i\}] + \frac{1}{2^{k+1}} - \epsilon''} \\
&\qquad\qquad (\text{Claim 10.9 and } |\{i \in X \mid \hat{b}(i) = 1\}| = z \bmod 2) \\
&\geq \frac{1}{t} \cdot \sum_{i \in [t]} \frac{\frac{1}{2^{k+1}} + \epsilon''}{\frac{1}{2^k}} \qquad (\text{Claim 10.9 and } |\{i \in X \mid \hat{b}(i) = 1\}| = z \bmod 2) \\
&\geq \frac{1 + \epsilon''}{2},
\end{aligned}$$

as required for a contradiction.

$\square$

11 Discussion on Conjecture 1.3

11.1 Conjecture 1.3 on Cliques

As a warmup, we first consider the case of an (unweighted) n-clique. The expectation term in Eq. (1) can be computed as follows.

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\subseteq[n]}\left[\min(\mathsf{wt}(S),\mathsf{wt}(\overline{S}))\right] &= \sum_{k=0}^{n} \frac{\binom{n}{k}}{2^n\cdot\binom{n}{2}}\cdot\min\left(\binom{k}{2},\binom{n-k}{2}\right)\\
&= \sum_{k=2}^{n-2} \frac{\binom{n}{k}}{2^n\cdot\binom{n}{2}}\cdot\min\left(\binom{k}{2},\binom{n-k}{2}\right)\\
&= \sum_{k=2}^{n-2} \frac{\binom{n}{k}}{2^n}\cdot\frac{\binom{\min(k,n-k)}{2}}{\binom{n}{2}}.
\end{aligned}$$

Defining $z(k)=\min(k,n-k)$ for $k\in[n]$ and observing that $\binom{n}{k}=\binom{n}{z(k)}$, we get:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\subseteq[n]}\left[\min(\mathsf{wt}(S),\mathsf{wt}(\overline{S}))\right] &= \sum_{k=2}^{n-2} \frac{\binom{n}{z(k)}}{2^n}\cdot\frac{\binom{z(k)}{2}}{\binom{n}{2}}\\
&= \frac{1}{2^n}\cdot\sum_{k=2}^{n-2}\binom{n-2}{z(k)-2} \qquad\qquad \left(\text{As } \tbinom{a}{b}\cdot\tbinom{b}{2}=\tbinom{a}{2}\cdot\tbinom{a-2}{b-2}\right)\\
&= \frac{1}{2^n}\cdot\left(2^{n-2}-\binom{n-2}{\lfloor(n-1)/2\rfloor-1}-\binom{n-2}{\lfloor(n-1)/2\rfloor}\right),
\end{aligned}$$

where to get the last equality, note that the only terms of the form $\binom{n-2}{i}$ that are missing the above sum are the terms for $i=\lfloor(n-1)/2\rfloor-1$ and $i=\lfloor(n-1)/2\rfloor$. Now, using the identity $\binom{a}{b}+\binom{a}{b+1}=\binom{a+1}{b+1}$, we get:

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\subseteq[n]}\left[\min(\mathsf{wt}(S),\mathsf{wt}(\overline{S}))\right] &= \frac{1}{2^n}\cdot\left(2^{n-2}-\binom{n-1}{\lfloor(n-1)/2\rfloor}\right)\\
&= \frac{1}{4}-\frac{\binom{n-1}{\lfloor(n-1)/2\rfloor}}{2^n}.
\end{aligned}$$

Observe that the second term above is decreasing in n and is strictly less than $3/32$ for $n\geq 20$, which makes the right hand side of Eq. (1) at most $\frac{1}{2}$ for $n\geq 20$. As there is always a cut containing at least half the edges, we get that Eq. (1) holds for all cliques of size $n\geq 20$. The case of cliques of size $n<20$ can be verified by brute calculation. It will also show that Conjecture 1.3 is tight for $n=3$ and $n=5$.

11.2 Conjecture 1.3 on Unweighted Graphs

In this section, we show that Conjecture 1.3 is indeed correct for all unweighted graphs (all edges having the same weight), at least when the graph is sufficiently large. Specifically, we show:

Theorem 11.1. *Conjecture 1.3 holds for all graphs G with at least 10^9 edges, where all edge weights are the same.*

Proof. Let m be the number of edges in G (so each edge has weight $1/m$) and $\epsilon = 1/100$. We run the following procedure on G.

1. Iteratively, find and remove maximal matchings $M_1, \ldots, M_k$ of size larger than $m' = \sqrt{\epsilon m/2} \geq 2000$ whenever possible, each time in the remaining graph excluding all matching edges previously found.

2. Find a maximal matching M_0 in the remaining graph, and let M' be the subset of all remaining edges.

Let V_0 be the subset of vertices matched in M_0 and V_1 the subset of all other vertices. Also let E_0, E', E_1 be the subsets of edges (in the original graph G) connecting two vertices of V_0, connecting a vertex of V_0 and a vertex of V_1, and connecting two vertices of V_1, respectively.

For a subset A of the edges in G, we let $\mathsf{wt}_A(\cdot)$ denote the same function as $\mathsf{wt}(\cdot)$ except that only edges in A are considered. Note that, for any two disjoint subsets A, B of edges, we have:

$$\min\big(\mathsf{wt}_A(S), \mathsf{wt}_A(\overline{S})\big) + \min\big(\mathsf{wt}_B(S), \mathsf{wt}_B(\overline{S})\big) \leq \min\big(\mathsf{wt}_{A\cup B}(S), \mathsf{wt}_{A\cup B}(\overline{S})\big). \tag{24}$$

We next note that, for all $i \in [k]$, we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{S\subseteq[n]}\big[\min\big(\mathsf{wt}_{M_i}(S), \mathsf{wt}_{M_i}(\overline{S})\big)\big] &= \frac{1}{2}\cdot \mathop{\mathbb{E}}_{S\subseteq[n]}\big[\mathsf{wt}_{M_i}(S) + \mathsf{wt}_{M_i}(\overline{S}) - \big|\mathsf{wt}_{M_i}(S) - \mathsf{wt}_{M_i}(\overline{S})\big|\big] \\
&= \frac{1}{2}\cdot \mathop{\mathbb{E}}_{S\subseteq[n]}\big[\mathsf{wt}_{M_i}(S) + \mathsf{wt}_{M_i}(\overline{S})\big] - \frac{1}{2}\cdot \mathop{\mathbb{E}}_{S\subseteq[n]}\big[\big|\mathsf{wt}_{M_i}(S) - \mathsf{wt}_{M_i}(\overline{S})\big|\big] \\
&= \frac{|M_i|}{4m} - \frac{1}{2}\cdot \mathop{\mathbb{E}}_{S\subseteq[n]}\big[\big|\mathsf{wt}_{M_i}(S) - \mathsf{wt}_{M_i}(\overline{S})\big|\big].
\end{aligned}$$

(as each edge is cut with probability $1/2$)

Furthermore, $\big|\mathsf{wt}_{M_i}(S) - \mathsf{wt}_{M_i}(\overline{S})\big|$ is actually equal to $\frac{1}{2m}$ times the difference in the number of matched vertices on the two sides. As a result, we have (using $z = |M_i|$ for convenience):

$$\mathop{\mathbb{E}}_{S\subseteq[n]}\big[\min\big(\mathsf{wt}_{M_i}(S), \mathsf{wt}_{M_i}(\overline{S})\big)\big] = \frac{z}{4m} - \frac{1}{4m}\cdot \sum_{j=0}^{2z} \frac{\binom{2z}{j}}{2^{2z}} \cdot |j - (2z - j)|$$

(as there are $2z$ matched vertices)

$$
\begin{aligned}
&= \frac{z}{4m} - \frac{1}{4m} \cdot \sum_{j=0}^{2z} \frac{\binom{2z}{j}}{2^{2z}} \cdot (2z - 2 \cdot \min(j, 2z-j)) \\
&= \frac{z}{4m} - \frac{z}{2m} \cdot \sum_{j=0}^{2z} \frac{\binom{2z}{j}}{2^{2z}} + \frac{1}{2m} \cdot \sum_{j=0}^{2z} \frac{\binom{2z}{j}}{2^{2z}} \cdot \min(j, 2z-j) \\
&= \frac{z}{4m} - \frac{z}{2m} + \frac{1}{2^{2z+1}m} \cdot \left[2^{2z-1} - \binom{2z-1}{z-1}\right] \cdot 2z \\
&\hspace{20em} \text{(by Lemma A.2)} \\
&= \frac{z}{4m} - \frac{z}{2^{2z+1}m} \cdot \binom{2z}{z}.
\end{aligned}
$$

Using Stirling's approximation and the fact that $z = |M_i| \geq m' > 1800$, we can lower bound this as:

$$
\begin{aligned}
\mathop{\mathbb{E}}_{S \subseteq [n]} \left[\min\left(\mathsf{wt}_{M_i}(S), \mathsf{wt}_{M_i}(\overline{S})\right)\right] &= \frac{z}{4m} - \frac{z}{2^{2z+1}m} \cdot \binom{2z}{z} \\
&\geq \frac{z}{4m} - \frac{z}{2^{2z+1}m} \cdot \frac{(2z/e)^{2z} \cdot e \cdot \sqrt{2z}}{\left((z/e)^z \cdot \sqrt{2\pi} \cdot \sqrt{z}\right)^2} \\
&\geq \frac{z}{4m} - \frac{z}{2m} \cdot \sqrt{\frac{1}{2z}} \\
&\geq \frac{29z}{120m}.
\end{aligned} \tag{25}
$$

We finish the proof of Eq. (1) as follows:

$$
\begin{aligned}
\frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [n]} \left[\min\left(\mathsf{wt}(S), \mathsf{wt}(\overline{S})\right)\right] &\leq \frac{2}{3} - \frac{16}{15} \cdot \sum_{i=1}^{k} \mathop{\mathbb{E}}_{S \subseteq [n]} \left[\min\left(\mathsf{wt}_{M_i}(S), \mathsf{wt}_{M_i}(\overline{S})\right)\right] && \text{(Eq. (24))} \\
&\leq \frac{2}{3} - \frac{58}{225m} \cdot \sum_{i=1}^{k} |M_i| && \text{(Eq. (25))} \\
&= \frac{2}{3} - \frac{58}{225m} \cdot \left|\bigcup_{i=1}^{k} M_i\right|. && \text{(As the matchings are disjoint)}
\end{aligned}
$$

We now claim that $E_1 \subseteq \bigcup_{i=1}^{k} M_i$. Indeed, any edge in E_1 is not in M_0 by definition. It also cannot be in M', as then it would contradict the fact that M_0 is maximal. We get:

$$
\begin{aligned}
\frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [n]} \left[\min\left(\mathsf{wt}(S), \mathsf{wt}(\overline{S})\right)\right] &\leq \frac{2}{3} - \frac{58}{225m} \cdot |E_1| \\
&= \frac{2}{3} \cdot \left(\frac{|E_0|}{m} + \frac{|E'|}{m} + \frac{|E_1|}{m}\right) - \frac{58}{225} \cdot \frac{|E_1|}{m}
\end{aligned}
$$

$$\leq \frac{2}{3} \cdot \left(\frac{|E_0|}{m} + \frac{|E'|}{m} \right) + \frac{41}{100} \cdot \frac{|E_1|}{m}$$
$$\leq \frac{7}{10} \cdot \left(\frac{|E_0|}{m} + \frac{|E'|}{m} \right) + \frac{21}{50} \cdot \frac{|E_1|}{m} - \frac{1}{100}.$$

We now claim that E_0 contains at most $\binom{2m'}{2} \leq \epsilon m$ edges, because of the fact that V_0 contains at most $2m'$ vertices (as M_0 is of size at most m'). This gives:

$$\begin{aligned}
\frac{2}{3} - \frac{16}{15} \cdot \mathop{\mathbb{E}}_{S \subseteq [n]} \left[\min\left(\mathsf{wt}(S), \mathsf{wt}(\overline{S})\right) \right] &\leq \frac{7\epsilon}{10} + \frac{7}{10} \cdot \frac{|E'|}{m} + \frac{21}{50} \cdot \frac{|E_1|}{m} - \frac{1}{100} \\
&\leq \frac{7}{10} \cdot \frac{|E'|}{m} + \frac{21}{50} \cdot \frac{|E_1|}{m} \\
&= p \cdot \frac{|E'|}{m} + 2p(1-p) \cdot \frac{|E_1|}{m}. \qquad \text{(setting } p = 7/10\text{)}
\end{aligned}$$

We now show that the right hand side can be upper bounded by $\mathsf{Max\text{-}Cut}(G)$, thereby finishing the proof of Eq. (1). For this, consider a cut that places all vertices of V_0 to the left side while each vertex of V_1 is independently assigned to the right side with probability p and to the left side otherwise. The maximum cut in G is at least the (expected) value of this cut, which is just the right hand side above, as desired.

□

References

[ABP19] Noga Alon, Boris Bukh, and Yury Polyanskiy. List-decodable zero-rate codes. *IEEE Trans. Inf. Theory*, 65(3):1657–1667, 2019. 7, 9, 19

[ADL06] Rudolf Ahlswede, Christian Deppe, and Vladimir S. Lebedev. Non-binary error correcting codes with noiseless feedback, localized errors, or both. In *International Symposium on Information Theory (ISIT)*, pages 2486–2487, 2006. 1

[Alo02] Noga Alon. Voting paradoxes and digraphs realizations. *Adv. Appl. Math.*, 29(1):126–135, 2002. 4

[Ber64] Elwyn R. Berlekamp. *Block Coding with Noiseless Feedback*. PhD thesis, Massachusetts Institute of Technology (MIT), 1964. 1, 2, 5

[Ber68] Elwyn R Berlekamp. Block coding for the binary symmetric channel with noiseless, delayless feedback. *Error-correcting codes*, pages 61–68, 1968. 1, 2, 5

[Bli86] Vladimir Markovich Blinovskii. Bounds for codes in the case of finite-volume list decoding. *Problemy Peredachi Informatsii*, 22(1):11–25, 1986. 7, 9, 19

[Bli09] Vladimir M Blinovsky. Plotkin bound generalization to the case of multiple packings. *Problems of Information Transmission*, 45(1):1–4, 2009. 7, 9

[Bur76] Marat Valievich Burnashev. Data transmission over a discrete channel with feedback. random transmission time. *Problemy peredachi informatsii*, 12(4):10–30, 1976. 5

[EKS20] Klim Efremenko, Gillat Kol, and Raghuvansh R. Saxena. Binary interactive error resilience beyond 1/8. In *Foundations of Computer Science (FOCS)*, pages 470–481, 2020. 5, 6, 9

[EKSZ22] Klim Efremenko, Gillat Kol, Raghuvansh Saxena, and Zhijun Zhang. Binary codes with resilience beyond 1/4 via interaction. In *Foundations of Computer Science (FOCS)*, 2022. 4, 6, 9, 11

[Eli57] Peter Elias. List decoding for noisy channels. 1957. 7, 9

[ESSG10] Krishnan Eswaran, Anand D. Sarwate, Anant Sahai, and Michael Gastpar. Zero-rate feedback can achieve the empirical capacity. *IEEE Transactions on Information Theory*, 56(1):25–39, 2010. 5

[For68] David G. Forney. Exponential error bounds for erasure, list, and decision feedback schemes. *IEEE Transactions on Information Theory*, 14(2):206–220, 1968. 5

[GGZ23] Meghal Gupta, Venkatesan Guruswami, and Rachel Yun Zhang. Binary error-correcting codes with minimal noiseless feedback. In *Symposium on Theory of Computing (STOC)*, 2023. 5

[GKZ22] Meghal Gupta, Yael Tauman Kalai, and Rachel Yun Zhang. Interactive error correcting codes over binary erasure channels resilient to 1/2 adversarial corruption. In *Symposium on Theory of Computing (STOC)*, 2022. 6, 9

[GS00] 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*, pages 181–190, 2000. 7

[Gur03] Venkatesan Guruswami. List decoding from erasures: Bounds and code constructions. *IEEE Transactions on Information Theory*, 2003. 7, 9

[GY21] Gregory Z. Gutin and Anders Yeo. Lower bounds for maximum weighted cut. *CoRR*, abs/2104.05536, 2021. 4

[GZ22a] Meghal Gupta and Rachel Yun Zhang. Efficient interactive coding achieving optimal error resilience over the binary channel. *CoRR*, abs/2207.01144, 2022. 5, 6

[GZ22b] Meghal Gupta and Rachel Yun Zhang. The optimal error resilience of interactive communication over binary channels. In *Symposium on Theory of Computing (STOC)*, 2022. 5, 6, 9

[GZ22c] Meghal Gupta and Rachel Yun Zhang. Positive rate binary interactive error correcting codes resilient to $> 1/2$ adversarial erasures. *CoRR*, abs/2201.11929, 2022. 6, 9

[HKV15] Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with partial noiseless feedback. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, volume 40 of *LIPIcs*, pages 881–897, 2015. 5

[Hor63] Michael Horstein. Sequential transmission using noiseless feedback. *IEEE Transactions on Information Theory*, 9(3):136–143, 1963. 5

[Plo60] M. Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. 2, 6

[PT86] Svatopluk Poljak and Daniel Turzík. A polynomial time heuristic for certain subgraph optimization problems with guaranteed worst case bound. *Discrete Mathematics*, 58(1):99–104, 1986. 4

[Sah08] Anant Sahai. Why do block length and delay behave differently if feedback is present? *IEEE Transactions on Information Theory*, 54(5):1860–1886, 2008. 5

[Sch92] Leonard J Schulman. Communication on noisy channels: A coding theorem for computation. In *Foundations of Computer Science (FOCS)*, pages 724–733, 1992. 6

[Sch93] Leonard J Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of computing (STOC)*, pages 747–756, 1993. 6

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. 6

[SF11] Ofer Shayevitz and Meir Feder. Optimal feedback communication via posterior matching. *IEEE Transactions on Information Theory*, 57(3):1186–1222, 2011. 5

[Sha56] Claude E. Shannon. The zero error capacity of a noisy channel. *IRE Transactions on Information Theory*, 2(3):8–19, 1956. 1, 5

[Sha09] Ofer Shayevitz. On error correction with feedback under list decoding. In *IEEE International Symposium on Information Theory, ISIT*, pages 1253–1257, 2009. 5, 7, 9

[SW13] Ofer Shayevitz and Michèle A. Wigger. On the capacity of the discrete memoryless broadcast channel with feedback. *IEEE Transactions on Information Theory*, 59(3):1329–1345, 2013. 5

[SWS92] Joel Spencer, Peter Winkler, and South St. Three thresholds for a liar. *Combinatorics, Probability and Computing*, 1:81–93, 1992. 1

[Wie48] Norbert Wiener. Cybernetics: Or control and communication in the animal and the machine. 1948. 1

[Woz58] John M Wozencraft. List decoding. *Quarterly Progress Report*, 48:90–95, 1958. 7, 9

[WQC17] Gang Wang, Yanyuan Qin, and Chengjuan Chang. Communication with partial noisy feedback. In *IEEE Symposium on Computers and Communications (ISCC)*, pages 602–607, 2017. 5

[Zig76] K.Sh. Zigangirov. On the number of correctable errors for transmission over a binary symmetrical channel with feedback. *Problems of Information Transmission*, 12:85–97, 1976. 1

A Technical Preliminaries

A.1 Concentration Bounds

We use the following version of Chernoff bound.

Lemma A.1 (Chernoff bound). *For all $n \geq 1$ and independent random variables $X_1, \ldots, X_n \in [0,1]$, let $X = \frac{1}{n} \cdot \sum_{i \in [n]} X_i$. For all $\epsilon > 0$, it holds that*

$$\Pr(|X - \mathbb{E}[X]| \geq \epsilon) \leq 2 \cdot \exp\bigl(-2\epsilon^2 n\bigr).$$

A.2 Properties of Binomial Coefficients

We use the following folklore identities involving binomial coefficients. Proofs are included for completeness.

Lemma A.2. *For all $n \geq 0$, it holds that*

$$\sum_{i=0}^{n} \min(i, n-i) \cdot \binom{n}{i} = \left[2^{n-1} - \binom{n-1}{\lceil n/2 \rceil - 1}\right] \cdot n.$$

Proof. Using the properties of binomial coefficients, we have

$$\begin{aligned}
&\sum_{i=0}^{n} \min(i, n-i) \cdot \binom{n}{i} \\
&\qquad = 2 \cdot \sum_{i=0}^{\lfloor n/2 \rfloor} i \cdot \binom{n}{i} - \mathbb{1}[n \text{ is even}] \cdot \frac{n}{2} \cdot \binom{n}{n/2} \\
&\qquad = 2 \cdot \sum_{i=1}^{\lfloor n/2 \rfloor} n \cdot \binom{n-1}{i-1} - \mathbb{1}[n \text{ is even}] \cdot \frac{n}{2} \cdot \binom{n}{n/2} \\
&\qquad = 2 \cdot \sum_{i=0}^{\lfloor n/2 \rfloor - 1} n \cdot \binom{n-1}{i} - \mathbb{1}[n \text{ is even}] \cdot \frac{n}{2} \cdot \binom{n}{n/2} \\
&\qquad = 2n \cdot \frac{2^{n-1} - \mathbb{1}[n-1 \text{ is even}] \cdot \binom{n-1}{(n-1)/2}}{2} - \mathbb{1}[n \text{ is even}] \cdot \frac{n}{2} \cdot \binom{n}{n/2} \\
&\qquad = 2^{n-1} n - \mathbb{1}[n \text{ is odd}] \cdot n \cdot \binom{n-1}{(n-1)/2} - \mathbb{1}[n \text{ is even}] \cdot n \cdot \binom{n-1}{n/2-1} \\
&\qquad = \left[2^{n-1} - \binom{n-1}{\lceil n/2 \rceil - 1} \right] \cdot n,
\end{aligned}$$

as claimed. □

A.3 Properties of the Function d

We now show some useful properties of the function d defined in Eq. (7).

Claim A.3. *The following hold:*

1. *For all $m \geq k \geq 1$, it holds that*

$$\mathsf{d}(m, k) = 1 - \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}}.$$

2. *For all $m_2 \geq m_1 \geq k \geq 1$, it holds that $\mathsf{d}(m_2, k) \leq \mathsf{d}(m_1, k)$, and moreover,*

$$\lim_{m \to \infty} \mathsf{d}(m, k) = 1 - \frac{1}{2^{k-1}}.$$

   It follows that $\mathsf{d}(m, k) \geq 1 - \frac{1}{2^{k-1}}$ for all $m \geq k \geq 1$.

3. *For all $m \geq k_2 \geq k_1 \geq 1$, it holds that $\mathsf{d}(m, k_2) \geq \mathsf{d}(m, k_1)$.*

Proof. For Item 1, observe that

$$1 - \mathsf{d}(m, k) = \frac{\lfloor m/2 \rfloor^{\underline{k}} + \lceil m/2 \rceil^{\underline{k}}}{m^{\underline{k}}}. \tag{26}$$

When m is odd, Eq. (26) simplifies to

$$1 - \mathsf{d}(m,k) = \frac{\lfloor m/2 \rfloor - k + 1 + \lceil m/2 \rceil}{m-k+1} \cdot \frac{\lfloor m/2 \rfloor^{\underline{k-1}}}{m^{\underline{k-1}}} = \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}},$$

as $\lfloor m/2 \rfloor + \lceil m/2 \rceil = m$. Similarly, when m is even, Eq. (26) becomes

$$1 - \mathsf{d}(m,k) = \frac{\lfloor m/2 \rfloor + \lceil m/2 \rceil}{m} \cdot \frac{(\lfloor m/2 \rfloor - 1)^{\underline{k-1}}}{(m-1)^{\underline{k-1}}} = \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}}.$$

This proves Item 1.

Fix $k \geq 1$. In order to show $\mathsf{d}(m,k)$ is decreasing in m, it is sufficient to show $\mathsf{d}(m+1,k) \leq \mathsf{d}(m,k)$ for any even $m \geq k$ by Item 1. To this end, observe that

$$\begin{aligned} 1 - \mathsf{d}(m+1,k) &= \frac{(m/2)^{\underline{k-1}}}{(m+1)^{\underline{k-1}}} \\ &= \frac{m/2}{m/2-k+1} \cdot \frac{(m-k+2)(m-k+1)}{(m+1)\cdot m} \cdot \frac{(m/2-1)^{\underline{k-1}}}{(m-1)^{\underline{k-1}}} \\ &= \frac{(m-k+2)(m-k+1)}{(m-2k+2)(m+1)} \cdot (1 - \mathsf{d}(m,k)) \\ &\geq 1 - \mathsf{d}(m,k) \end{aligned}$$

because

$$\frac{(m-k+2)(m-k+1)}{(m-2k+2)(m+1)} = \frac{m^2 - (2k-3)\cdot m + (k-2)(k-1)}{m^2 - (2k-3)\cdot m - (2k-2)} \geq 1.$$

Moreover, we also have

$$\lim_{m\to\infty} \mathsf{d}(m,k) = 1 - \lim_{m\to\infty} \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}} = 1 - \prod_{i=1}^{k-1} \lim_{m\to\infty} \frac{\lceil m/2 \rceil - i}{2\lceil m/2 \rceil - i} = 1 - \frac{1}{2^{k-1}},$$

as claimed, concluding the proof of Item 2.

Finally, for any $m > k \geq 1$, we similarly have

$$1 - \mathsf{d}(m,k+1) = \frac{(\lceil m/2 \rceil - 1)^{\underline{k}}}{(2\lceil m/2 \rceil - 1)^{\underline{k}}} = \frac{\lceil m/2 \rceil - k}{2\lceil m/2 \rceil - k} \cdot \frac{(\lceil m/2 \rceil - 1)^{\underline{k-1}}}{(2\lceil m/2 \rceil - 1)^{\underline{k-1}}} \leq 1 - \mathsf{d}(m,k).$$

Item 3 then follows. □

We finish by proving Lemma 5.3.

Proof of Lemma 5.3. It is sufficient to prove

$$1 - \mathsf{d}(m,k') + 1 - \mathsf{d}(k',k) \leq 1 - \mathsf{d}(m,k),$$

or equivalently, by Item 1 of Claim A.3,

$$\frac{(t-1)^{\underline{k'-1}}}{(2t-1)^{\underline{k'-1}}}+\frac{(s-1)^{\underline{k-1}}}{(2s-1)^{\underline{k-1}}}\le\frac{(t-1)^{\underline{k-1}}}{(2t-1)^{\underline{k-1}}},\tag{27}$$

where $t=\lceil m/2\rceil$ and $s=\lceil k'/2\rceil$. Without loss of generality, we assume $t\ge k'$ and $s\ge k$, as otherwise either $\mathsf{d}(m,k')=1$ or $\mathsf{d}(k',k)=1$ by definition. In both cases, the claim easily follows from Claim A.3. Rearranging Eq. (27), we also have

$$\frac{(s-1)^{\underline{k-1}}(2t-1)^{\underline{k-1}}}{(2s-1)^{\underline{k-1}}(t-1)^{\underline{k-1}}}\le 1-\frac{(t-k)^{\underline{k'-k}}}{(2t-k)^{\underline{k'-k}}}.$$

Observe that

$$\frac{(2t-1)^{\underline{k-1}}}{(t-1)^{\underline{k-1}}}=\prod_{u=1}^{k-1}\frac{2t-u}{t-u}\le\prod_{u=1}^{k-1}\frac{2k'-u}{k'-u}=\frac{(2k'-1)^{\underline{k-1}}}{(k'-1)^{\underline{k-1}}}$$

as $t\ge k'$ while we also have

$$\frac{(t-k)^{\underline{k'-k}}}{(2t-k)^{\underline{k'-k}}}=\prod_{u=k}^{k'-1}\frac{t-u}{2t-u}\le\frac{1}{2^{k'-k}}.$$

So it is also sufficient to show

$$\frac{(s-1)^{\underline{k-1}}(2k'-1)^{\underline{k-1}}}{(2s-1)^{\underline{k-1}}(k'-1)^{\underline{k-1}}}\le 1-\frac{1}{2^{k'-k}}.\tag{28}$$

To this end, we have

$$\begin{aligned}
\frac{(s-1)^{\underline{k-1}}(2k'-1)^{\underline{k-1}}}{(2s-1)^{\underline{k-1}}(k'-1)^{\underline{k-1}}}&=\prod_{u=1}^{k-1}\frac{(s-u)(2k'-u)}{(2s-u)(k'-u)}\\
&=\prod_{u=1}^{k-1}\left(1-\frac{(k'-s)\cdot u}{(2s-u)(k'-u)}\right)\\
&\le\prod_{u=1}^{k-1}\left(1-\frac{(k'-s)\cdot u}{(2s-1)(k'-1)}\right)\\
&\le\prod_{u=1}^{k-1}\exp\left(-\frac{(k'-s)\cdot u}{(2s-1)(k'-1)}\right) && (\text{as } 1-x\le\exp(-x))\\
&=\exp\left(-\sum_{u=1}^{k-1}\frac{(k'-s)\cdot u}{(2s-1)(k'-1)}\right)\\
&=\exp\left(-\frac{(k'-s)\cdot k(k-1)}{(4s-2)(k'-1)}\right)
\end{aligned}$$

$$\leq \exp\left(-\frac{k(k-1)}{8s-4}\right) \qquad (\text{as } \tfrac{k'-s}{k'-1} \geq \tfrac{2s-1-s}{2s-1-1} = \tfrac{1}{2} \text{ due to } k' \geq 2s-1)$$

Now suppose $\frac{k(k-1)}{8s-4} \geq \frac{3}{10}$. Then we easily have

$$\frac{(s-1)^{\underline{k-1}}(2k'-1)^{\underline{k-1}}}{(2s-1)^{\underline{k-1}}(k'-1)^{\underline{k-1}}} \leq \exp(-\frac{3}{10}) \leq \frac{3}{4} \leq 1 - \frac{1}{2^{k'-k}}$$

as $k'-k \geq 2$ by the assumption $(k', k) \neq (3, 2)$. So it remains to consider the case where $\frac{k(k-1)}{8s-4} \leq \frac{3}{10}$. Since $\exp(-x) \leq 1 - \frac{5}{6}x$ for $x \in [0, \frac{3}{10}]$, to prove Eq. (28), it is sufficient to show in this case that

$$\frac{5}{6} \cdot \frac{k(k-1)}{8s-4} \geq \frac{1}{2^{2s-1-k}}$$

as $k' \geq 2s-1$. Equivalently, we have

$$\frac{5k(k-1)}{2^k} \geq \frac{48s-24}{2^{2s-1}}. \tag{29}$$

For $k \geq 3$, as k increases to $k+1$, the left hand side of Eq. (29) is multiplied by a factor of $\frac{k+1}{2(k-1)}$, which is always upper bounded by 1. Thus we can get

$$\frac{5k(k-1)}{2^k} \geq \frac{5s(s-1)}{2^s} \geq \frac{48s-24}{2^{2s-1}}$$

as $3 \leq k \leq s$. The only remaining case is $k=2$, where Eq. (29) holds for $s \geq 4$. For $s \in [2,3]$, namely $k' \in [4,6]$, it turns out not to hold as some steps in the above argument are not tight. However, plugging $k=2$, $s=3$, and $k' \in [5,6]$ into Eq. (28) shows they indeed satisfy the equation. For the other case where $k=s=2$ and $k'=4$, Eq. (27) is equivalent to

$$\frac{(t-1)(t-2)(t-3)}{(2t-1)(2t-2)(2t-3)} + \frac{1}{3} - \frac{t-1}{2t-1} = -\frac{(t-2)(t+3)}{6(2t-1)(2t-3)} \leq 0,$$

which is always true as $t \geq k' \geq 4$. This finally concludes the proof. $\square$

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il