Electronic Colloquium on Computational Complexity, Report No. 320 (2026)

Binary Two-Way Codes with Optimal Erasure Resilience

Klim Efremenko[*]
Ben-Gurion University

Gillat Kol[†]
Princeton University

Sergey Komech[‡]
Ben-Gurion University

Raghuvansh R. Saxena[§]
Tata Institute of
Fundamental Research

Zhijun Zhang[¶]
INSAIT, Sofia University
"St. Kliment Ohridski"

Abstract

Two-way codes use communication in both directions to convey a message from a sender to a receiver over a noisy channel. In a beautiful work, [GKZ22] proved that for binary channels with adversarial erasures, interaction strictly outperforms one-way communication: they constructed a two-way code with erasure resilience approaching $\frac{3}{5}$, surpassing the $\frac{1}{2}$ upper bound for traditional binary codes implied by the Plotkin bound. They also proved that no binary two-way code can have erasure resilience greater than $\frac{2}{3}$. Whether this bound could be approached remained a central open problem.

We resolve this problem in the affirmative by constructing a deterministic binary two-way code that approaches the optimal erasure resilience of $\frac{2}{3}$ while achieving constant rate and linear encoding and decoding times. Our construction is remarkably simple and takes an entirely different approach from prior work.

[*]klimefrem@gmail.com. Supported by the European Research Council Grant number: 949707.
[†]gillat.kol@gmail.com.
[‡]sergey.komech@gmail.com. Supported by the European Research Council Grant number: 949707.
[§]raghuvansh.saxena@gmail.com. Supported by the Department of Atomic Energy, Government of India, under project no. RTI4014.
[¶]zhijun.zhang@insait.ai. This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure).

ISSN 1433-8092

Contents

1 Introduction **1**
1.1 Our Results 1
1.2 Related Work 2

2 Proof Overview **3**

3 Preliminaries **6**

4 Our Two-Way Code **8**

5 Analysis of Our Two-Way Code **9**

1 Introduction

Alice wishes to send a message $x \in \{0,1\}^n$ to Bob over a noisy binary channel. Classical error-correcting codes approach this problem via one-way communication: Alice encodes her message and transmits the resulting codeword to Bob. Many communication systems, however, have built-in bidirectional interaction, used, for instance, for acknowledgments, synchronization, or control. *Two-way codes* [GKZ22, EKSZ22] seek to exploit this capability by allowing both parties to communicate back and forth over the channel, with transmissions in both directions subject to noise. The ultimate objective, however, remains unchanged: Bob must recover Alice's message.

Over memoryless stochastic channels, Shannon's feedback theorem [Sha56] establishes that feedback and two-way interaction yield no gain in capacity. Perhaps surprisingly, over an *adversarial* channel, Bob's replies can significantly enhance error resilience, even though Bob has no private input of his own and communicates over the very same adversarial channel [GKZ22, EKSZ22]. Intuitively, when an adversary operates under a fixed corruption budget, noise inflicted in earlier rounds directly constrains the adversary's future capabilities. Bob's replies can thus inform Alice of these past corruptions, enabling her to adapt subsequent transmissions accordingly. This budgeting advantage has no analogue in the memoryless stochastic setting, where corruptions are independent across rounds.

[GKZ22] initiated the study of two-way codes over binary channels subject to *adversarial erasures*, a setting extensively studied in the one-way regime. In this model, the two parties take turns transmitting bits, while an adversary with an erasure budget bounded by a fixed fraction of the total communication decides, for each transmitted bit, whether to deliver it intact or replace it with the erasure symbol '$\perp$'.

By the Plotkin bound [Plo60], it has long been known that binary one-way codes can achieve an erasure resilience of at most $\frac{1}{2}$. [GKZ22] surpassed this barrier by constructing two-way codes with erasure resilience approaching $\frac{3}{5}$, while also proving an upper bound of $\frac{2}{3}$ on the maximum achievable resilience. Subsequent work by [GZ23b] obtained constant-rate codes with resilience approaching $\frac{6}{11}$. Whether two-way codes can achieve the optimal erasure resilience of $\frac{2}{3}$ remained a central open problem explicitly posed in these works.

1.1 Our Results

We resolve this open problem in the affirmative. We construct deterministic binary two-way codes that approach the optimal erasure resilience of $\frac{2}{3}$, while simultaneously achieving constant rate and linear encoding and decoding times. Our construction is remarkably simple, relying on ideas fundamentally different from prior literature.

Theorem 1.1. *For every $\varepsilon \in (0, \frac{2}{3})$, there exists a deterministic, constant-rate binary two-way code with linear encoding and decoding time that tolerates a $\left(\frac{2}{3} - \varepsilon\right)$ fraction of adversarial erasures.*

1.2 Related Work

Two-way codes. For binary adversarial erasures, [GKZ22] constructed a two-way code of quadratic block length with resilience approaching $\frac{3}{5}$, surpassing the optimal $\frac{1}{2}$ resilience of one-way codes. They also proved an upper bound of $\frac{2}{3}$ on the resilience of any binary two-way code. Subsequently, [GZ23b] obtained a computationally efficient, constant-rate code, albeit with a lower resilience approaching $\frac{6}{11}$.

In a very recent work, [Sax27] studied two-way coding against adversarial erasures over the related *adaptive* binary channel. In this setting, the communication order need not be fixed in advance: the parties can adapt their speaking turns based on their inputs and received transcripts. They give a computationally efficient, constant-rate adaptive two-way code resilient to a $\frac{2}{3}$-fraction of adversarial erasures. While the $\frac{2}{3}$ upper bound of [GKZ22] does not apply to the adaptive setting, [Sax27] also prove an upper bound of $\frac{4}{5}$ on the erasure resilience of any adaptive two-way code.

For binary adversarial corruptions (bit flips), the Plotkin bound limits the maximum resilience of any one-way code to $\frac{1}{4}$. [EKSZ22] bypassed this limit by constructing a (non-adaptive) two-way code resilient to a $\frac{1}{4} + 10^{-5}$ fraction of corruptions. An upper bound of $\frac{2}{7}$ on the resilience of any binary two-way corruption code was given by [GKZ22] and subsequently improved to $\frac{13}{47}$ by [GZ23b].

In a very recent work, [EKK+27] studied two-way codes in the quantum setting. Specifically, they considered the transmission of an unknown quantum state from Alice to Bob over a quantum channel of large constant dimension. They showed that allowing Bob to transmit back to Alice enables codes that surpass the $\frac{1}{2}$ upper bound on adversarial erasure resilience implied by the no-cloning theorem, even if Bob's backward channel is purely classical and subject to the shared adversarial erasure budget. In this setting, they showed that the optimal fraction of adversarial erasures that can be tolerated is $\frac{3}{5}$, achieved by a constant-rate, computationally efficient protocol.

Feedback channels. Shannon initiated the study of *feedback channels* [Sha56], in which Alice learns the symbol received by Bob after each channel use, and showed that feedback cannot increase the capacity of memoryless *stochastic* channels. In the adversarial regime, however, feedback can strictly improve error resilience. For adversarial binary corruptions with noiseless feedback, a code achieving the optimal resilience of $\frac{1}{3}$ dates back to [Ber64, Zig76], surpassing the $\frac{1}{4}$ barrier of the feedback-free setting. More recent work has studied variations with partial, highly constrained, or noisy feedback [HKV15, GGZ23, BEK+25, WQC17].

Conceptually, two-way codes can be viewed as codes over channels with noisy feedback. In contrast to standard feedback models, however, Bob's transmissions traverse the same adversarial channel as Alice's and share the same global adversarial budget. Furthermore, Bob's messages can be arbitrary functions of his past view, rather than merely echoing received channel symbols.

Interactive coding. The field of *interactive coding*, pioneered by [Sch92, Sch93, Sch96], considers a more general problem than two-way message transmission: rather than transmitting a single unidirectional message, the parties must robustly simulate an arbitrary interactive protocol over a noisy channel. For any channel model, the maximum error resilience of interactive coding schemes is naturally bounded between half the resilience of one-way codes and the resilience of two-way codes. Indeed, two-way message transmission is a special case of interactive coding, while any interactive protocol can be naively simulated by exchanging both parties' inputs using two executions of a one-way code.

Over binary channels subject to adversarial bit flips, the $\frac{1}{8}$ resilience achieved in the pioneering work of [BR11] was first surpassed by [EKS20]. The optimal resilience threshold of $\frac{1}{6}$ was subsequently proved by [GZ22], followed by computationally efficient, constant-rate schemes approaching this bound [GZ23a].

2 Proof Overview

In this section, we give an overview of the main ideas behind our two-way code.

At a high level, in our two-way code, Alice first encodes her message using an error-correcting code over a large constant alphabet $[q]$ for $q = O_\varepsilon(1)$, with relative distance close to 1. She then sends each symbol of the codeword using Inner; see Algorithm 1.

The Inner protocol. To send $z \in [q]$, Alice maintains a binary vector, initially all zeros. At first, the entire vector forms a single block. The protocol maintains the following invariant: at the beginning of every round, each block is monochromatic, meaning that it consists entirely of zeros or entirely of ones.

In each round, the parties process the current blocks of the vector one by one. Alice sends the bits of a block one by one. Bob then reports whether he received any of them: he sends repeated 1's if he did, and repeated 0's otherwise. His reply contains half as many bits as the block. Since the block is monochromatic, receiving even one bit reveals the entire block to Bob.

To finish processing the block, Alice splits it equally into $2q$ contiguous subblocks. If she receives any 1 from Bob's reply, she knows that Bob learned the block and flips all bits in subblocks $2z - 1$ and $2z$. If she receives any 0, she knows that Bob received no bits from the block and leaves it unchanged. If the entire reply is erased, she does not know whether Bob received any bits and flips only subblock $2z - 1$. The parties then move on to the next block in the current round.

After processing all the current blocks, the parties treat the resulting subblocks as the blocks of the next round and process them from the start of the vector. Thus, each new round has $2q$ times as many blocks as the previous round, each shorter by a factor of $2q$. The new blocks are monochromatic because Alice either leaves an entire subblock unchanged or flips all its bits.

Inner illustration. Figure 1 follows Alice's vector S for two possible inputs, $z = 2$ and $z = 3$, over the alphabet $[4]$. It shows the vector after each of the first three rounds. We assume that no bits are erased in the first round, which starts from the all-zero vector known to both parties. In each later round, the adversary erases exactly the forward blocks on which the two vectors differ. Bob's replies are never erased.

This erasure pattern keeps the two inputs indistinguishable to Bob. If Bob knows that Alice's input is either 2 or 3, receiving even one bit from a block on which the two vectors differ would identify her input. Bits from blocks on which the vectors agree do not distinguish the two inputs. Thus, under the illustrated erasure pattern, Bob has the same received transcript in both executions.

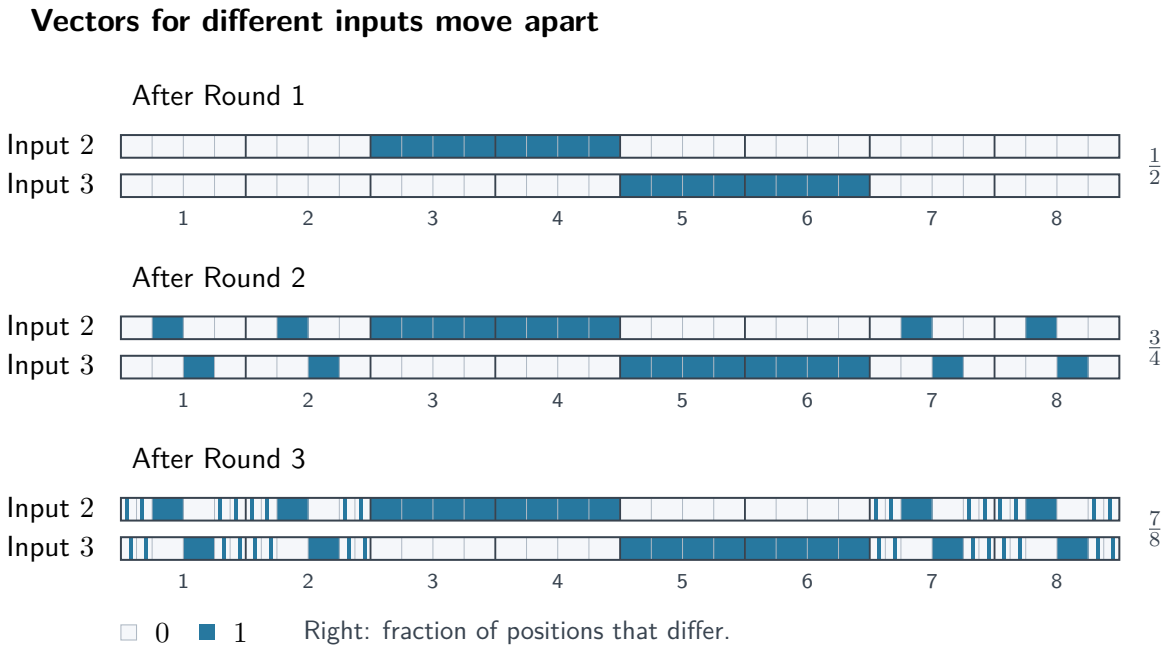


Figure 1: Alice's vectors after each of the first three rounds of Inner, with $q = 4$ and inputs $z = 2$ and $z = 3$. Blue denotes 1, and white denotes 0. Dark boundaries and the labels $1, \ldots, 8$ mark the eight blocks sent in round 2.

Resilience when Bob's replies are intact. Figure 1 illustrates how the Hamming distance between Alice's vectors for distinct inputs $z, z' \in [q]$ grows over the rounds of Inner. To keep the two inputs indistinguishable to Bob, the adversary must erase every bit transmitted by Alice from a block where the vectors differ. We first consider the illustrated erasure pattern, in which the adversary erases exactly these transmissions and never erases Bob's replies. By the similarity of two vectors, we mean the fraction of positions where they agree.

We next show that $r = O(\frac{q}{\varepsilon})$ iterations suffice to make the average similarity of the vectors transmitted for z and z' less than ε. Assuming this, the adversary then erases more than a $1 - \varepsilon$ fraction of Alice's transmissions. Since Alice sends $\frac{2}{3}$ of the bits in each round, this

requires erasing more than a $\frac{2}{3}(1-\varepsilon) > \frac{2}{3} - \varepsilon$ fraction of the total communication, which exceeds the adversary's budget.

To show that the average similarity is small, we work inductively. Initially, both vectors are all zeros, so their similarity is 1. By the end of the first round, the vectors differ on exactly four of the $2q$ blocks: $2z-1$, $2z$, $2z'-1$, and $2z'$. For input z, blocks $2z-1$ and $2z$ are flipped to all ones, while for input z', blocks $2z'-1$ and $2z'$ are flipped to all ones. All remaining blocks contain only zeros for both inputs. Thus, the similarity is $1 - \frac{4}{2q} = 1 - \frac{2}{q}$.

In the second round, each of the $2q-4$ common all-zero blocks is partitioned into $2q$ subblocks. These blocks and Bob's replies are delivered, so Alice flips the pair of subblocks selected by her input. The two inputs again select disjoint pairs, creating disagreement on exactly four of the $2q$ subblocks within each such block. The previously disagreeing blocks remain unchanged because they are entirely erased and Bob replies with 0's. Thus, the similarity after the second round is $(1 - \frac{2}{q})^2$.

The same process continues in subsequent rounds: existing disagreements persist, and each agreeing block creates disagreement on four of its $2q$ subblocks. Therefore, after round t, the similarity is $(1 - \frac{2}{q})^t$. The relative Hamming distance thus approaches 1 and stays close to 1 in all sufficiently late rounds. By choosing r large enough, the initial rounds account for an arbitrarily small fraction of Alice's transmissions. Even if we count the distance in every initial round as 0, the average relative Hamming distance over all transmitted vectors is greater than $1 - \varepsilon$.

Resilience when Bob's replies are erased. Now consider erasures of Bob's replies. Since each reply repeats a single bit, partial erasures consume budget without changing Alice's update. We may therefore assume that the adversary either erases an entire reply or leaves it unerased.

For a common all-zero block of length L, erasing both replies costs $\frac{L}{2}$ erasures in each execution. Alice then flips subblock $2z-1$ for input z and subblock $2z'-1$ for input z'. The vectors therefore differ on two of the $2q$ subblocks, leaving similarity $1 - \frac{1}{q}$ instead of $1 - \frac{2}{q}$. Thus, erasing both replies halves the disagreement created within this block. Erased replies can also undo earlier disagreements. However, compared with delivering a reply, erasing it changes Alice's update on only one subblock of length $\frac{L}{2q}$. The adversary therefore pays $\frac{L}{2}$ erasures for a possible decrease of at most $\frac{L}{2q}$ in the Hamming distance.

In the actual proof, we use a potential argument to formally show that it does not favor the adversary in such case. Similar arguments also apply if Bob's reply is erased for only one of the two executions.

Efficient decoding. Bob decodes using his received transcripts from the executions of Inner. For each coordinate j and possible symbol $a \in [q]$, he computes a cost $c_j(a)$. This cost is essentially the minimum number of erasures in both directions needed to explain his received transcript from the execution of Inner for the j-th codeword symbol, assuming Alice's input to that execution were a.

To compute these costs efficiently, we enumerate all inputs and erasure patterns for Inner once during public setup and store the resulting costs in a table. Both the alphabet size q and the number of bits exchanged in one execution of Inner depend only on ε, not on Alice's message length n. Thus, the number of inputs and erasure patterns also depends only on ε. For fixed ε, constructing the table therefore takes constant time, and each subsequent cost evaluation is a constant-time lookup.

Given these costs, we adapt *generalized minimum-distance* (GMD) decoding [FJ65, BGG20], a method for decoding concatenated codes. At each coordinate j, Bob selects a symbol $a \in [q]$ that minimizes $c_j(a)$. He then tries different cost cutoffs, erasing positions whose selected symbols have high cost, treating these symbols as less reliable. Each trial produces a received word that may contain both corruptions and erasures. To allow Bob to decode these words, we choose Alice's outer code over $[q]$ from [GI05], which supports linear-time decoding from a mixture of corruptions and erasures.

3 Preliminaries

Notation. For $n \geq 1$, $[n]$ is a shorthand for $\{1, \ldots, n\}$. For any tuple x and integer $i \geq 0$, define $x_{\leq i} = (x_1, \ldots, x_i)$. Similarly define $x_{<i}$, $x_{>i}$, etc., as well as corresponding superscript versions. The Hamming distance is denoted by $\Delta(\cdot, \cdot)$.

Binary two-way codes. A protocol over the two-way binary erasure channel is defined by

$$\Pi = (n, T, p, f_A, f_B, g),$$

where (1) n is the number of input bits; (2) T is the length of the protocol; (3) $p \in \{A, B\}^T$ specifies the order of speaking; (4) $f_A : \{0,1\}^n \times \{0,1,\perp\}^{<T} \to \{0,1\}$ and $f_B : \{0,1,\perp\}^{<T} \to \{0,1\}$ are the message functions computed by Alice and Bob, respectively; (5) $g : \{0,1,\perp\}^T \to \{0,1\}^n$ is the output function computed by Bob. An (erasure) adversary is defined by $\mathsf{Adv} \subseteq [T]$.

The execution of Π in the presence of Adv starts with Alice initially holding $x \in \{0,1\}^n$. Bob has no input. In round $i \in [T]$, if $p_i = A$, then Alice sends $f_A(x, \pi_A^{(<i)})$ to Bob, where $\pi_A^{(<i)}$ denotes Alice's view of the partial transcript before round i. Bob receives the same bit $f_A(x, \pi_A^{(<i)})$ if $i \notin \mathsf{Adv}$, or $\perp$ otherwise. Similarly, if $p_i = B$, then Bob sends $f_B(\pi_B^{(<i)})$ to Alice, where $\pi_B^{(<i)}$ denotes Bob's view of the partial transcript before round i. Alice receives the same bit $f_B(\pi_B^{(<i)})$ if $i \notin \mathsf{Adv}$, or $\perp$ otherwise. After all T rounds, Bob outputs $g(\pi_B^{(\leq T)})$.

We say an adversary Adv has budget $\delta \in [0,1]$ if $|\mathsf{Adv}| \leq \delta T$. Π has erasure resilience δ if for every input $x \in \{0,1\}^n$ and every adversary Adv with budget δ, execution of Π in the presence of Adv ends with Bob outputting $g(\pi_B^{(\leq T)}) = x$.

One-way coding primitives. We use the following efficient one-way coding primitive from [GI05]. It ensures distance approaching 1 and can decode in the presence of both corruptions and erasures.

Lemma 3.1 ([GI05])**.** *For $\delta > 0$ and $n \geq 1$, there exists an efficient code $\mathsf{Enc} : \{0,1\}^n \to [q]^N$ with relative distance $1-\delta$, where $q = O_\delta(1)$ and $N = O_\delta(n)$, such that*

1. *The encoder Enc encodes in linear time in n.*
2. *The decoder Dec (uniquely) decodes in linear time in n, so long as $2a + b < (1-\delta) \cdot N$, where a, b are the number of corruptions and erasures respectively.*

Lemma 3.2 follows by applying the generalized minimum distance decoding technique; see for example [FJ65, BGG20].

Lemma 3.2. *For $\delta > 0$ and $n \geq 1$, there exists an efficient code $\mathsf{Enc} : \{0,1\}^n \to [q]^N$ with relative distance $1-\delta$, where $q = O_\delta(1)$ and $N = O_\delta(n)$, such that*

1. *The encoder Enc encodes in linear time in n.*
2. *Let $B \geq 1$ be a constant integer. Suppose we are given cost functions $c_j : [q] \to \{0, \ldots, B\}$ for $j \in [N]$. Assume that $c_j(z_1) + c_j(z_2) \geq B$ for every $j \in [N]$ and all distinct $z_1, z_2 \in [q]$. Then there is a decoder Dec that uniquely recovers any $x \in \{0,1\}^n$ satisfying $\sum_{j \in [N]} c_j(\mathsf{Enc}(x)_j) < \frac{1-\delta}{2} \cdot BN$. Decoding runs in linear time in n, provided each cost $c_j(z)$ can be computed in constant time. The encoder does not depend on the cost functions.*

Proof. Use the code Enc promised by Lemma 3.1, which has minimum distance at least $d = (1-\delta) \cdot N$ between any pair of codewords. Let Dec' be the decoder for Enc. We show how to decode using the cost functions.

The decoding proceeds in B iterations. Each iteration $\tau \in [B]$ may return a candidate message x', and x' is accepted as output only if $\sum_{j \in [N]} c_j(\mathsf{Enc}(x')_j) < (1-\delta)/2 \cdot BN$. We first verify that such x' is unique. Indeed, if distinct $x, x' \in \{0,1\}^n$ satisfy the condition, then we have

$$\sum_{j \in [N]} (c_j(\mathsf{Enc}(x)_j) + c_j(\mathsf{Enc}(x')_j)) \geq B \cdot d = (1-\delta) \cdot BN,$$

because $\mathsf{Enc}(x)_j \neq \mathsf{Enc}(x')_j$ for at least d coordinates. This is a contradiction, however.

To recover x, iteration τ constructs $y_\tau \in \{1, \ldots, q, \perp\}^N$ as follows. For $j \in [N]$, the j-th symbol of y_τ is the unique $z \in [q]$ such that $c_j(z) < \tau/2$, and is $\perp$ if no such z exists. The uniqueness is guaranteed due to the assumption $c_j(z_1) + c_j(z_2) \geq B$ for $z_1 \neq z_2 \in [q]$. The candidate message of the current iteration is $\mathsf{Dec}'(y_\tau)$. It suffices to show that at least one of the iterations yields $x = \mathsf{Dec}'(y_\tau)$ such that $\sum_{j \in [N]} c_j(\mathsf{Enc}(x)_j) < (1-\delta)/2 \cdot BN$, if such x exists.

To see this, fix the unique x satisfying the cost condition. Let a_τ, b_τ denote the number of corruptions and erasures in y_τ with respect to $w = \mathsf{Enc}(x)$. By Lemma 3.1, $\mathsf{Dec}'(y_\tau) = x$ if $2a_\tau + b_\tau < (1-\delta) \cdot N$. We claim that

$$\sum_{\tau \in [B]} (2a_\tau + b_\tau) < (1-\delta) \cdot BN.$$

This will conclude the proof by an averaging argument.

We prove the claim by bounding the contribution of each coordinate $j \in [N]$ to the sum. Let $c_j^* = c_j(w_j)$. If $c_j^* \le B/2$, any other symbol $z \neq w_j \in [q]$ has cost at least $B - c_j^* \ge B/2 \ge \tau/2$ and thus can never go into y_τ. Moreover, the j-th coordinate of y_τ is $\perp$ for $\tau \le 2c_j^*$, and is w_j otherwise. The total contribution is thus $2c_j^*$.

In the meantime, if $c_j^* > B/2$, w_j never goes into y_τ. Moreover, any other symbol $z \neq w_j \in [q]$ has cost at least $B - c_j^*$, and can go into y_τ only if $\tau > 2(B - c_j^*)$. This happens for at most $B - 2(B - c_j^*) = 2c_j^* - B$ iterations. y_τ has an erasure in coordinate j in all other iterations. Overall, the total contribution is at most $B + 2c_j^* - B = 2c_j^*$.

Summing up the contribution from all coordinates, we have that

$$\sum_{\tau \in [B]} (2a_\tau + b_\tau) \le 2 \cdot \sum_{j \in [N]} c_j^* < (1-\delta) \cdot BN,$$

as claimed. Since evaluation of cost functions only takes constant time, the decoding still runs in linear time as B is constant. □

4 Our Two-Way Code

In this section, we present our two-way code for proving Theorem 1.1. At a high level, Alice first encodes her message using the outer code in Lemma 3.2. Alice and Bob then communicate each symbol of the outer encoding using an inner code that we show below. Finally, Bob computes the cost function for every symbol from his received transcript and applies the decoder for the outer code, as guaranteed by Lemma 3.2.

Parameters. Fix sufficiently small $\varepsilon > 0$. Let Enc be the encoder promised by Lemma 3.2 for $\delta = \varepsilon/10$, with codewords in $[q]^N$. Note that q is constant for fixed ε and that $N = O_\varepsilon(n)$. The inner code consists of $R = \lceil 10q/\varepsilon \rceil$ rounds. In each round, Alice sends $M = (2q)^R$ bits in total, while Bob sends $M/2$ bits in total. Thus, each execution of the inner code has length $R \cdot (M + M/2) = 3MR/2$, and the overall length of the protocol is

$$T = N \cdot 3MR/2 = 3MNR/2 = O_\varepsilon(n).$$

Blocks. The inner code starts with an all-zero bit string of length M. For $r \in [R]$, iteration r partitions the bit string into blocks of length $L_r = M/(2q)^{r-1}$ each. That is, the i-th block

Algorithm 1 The inner code to transmit $z \in [q]$.

Alice initializes $S \leftarrow 0^M$.
for $r = 1, \ldots, R$ **do**
  for $i = 1, \ldots, M/L_r$ **do**
    Alice sends block i of S using L_r bits.
    Bob sends $1^{L_r/2}$ if any bit of the current block is received, and $0^{L_r/2}$ otherwise.
    Alice subdivides block i of S into $2q$ subblocks of length $L_r/(2q)$ each.
    if Alice receives at least one "1" **then**
      Alice flips all bits in subblocks $2z-1$ and $2z$.
    else if Alice receives only $\perp$ **then**
      Alice flips all bits in subblock $2z-1$.
    end if
  end for
end for

in iteration r consists of bits indexed at $(i-1) \cdot L_r + 1, \ldots, i \cdot L_r$, for $i \in [M/L_r]$. Each block is guaranteed to be either all "0" or all "1", and is sent from Alice to Bob. Bob replies to each block with either $0^{L_r/2}$ or $1^{L_r/2}$, depending on whether he receives any bit of the current block from Alice. Upon receiving the feedback from Bob, Alice subdivides each block into $2q$ subblocks of equal length and updates accordingly. Algorithm 1 gives the formal description of the inner code.

Decoding. Let $\pi_j \in \{0, 1, \perp\}^{MR}$ be Bob's received transcript in the j-th execution of the inner code, for $j \in [N]$. For each $j \in [N]$, Bob can compute the table of the cost function c_j, whose value at $z \in [q]$ is the minimum of

$$B = (2R - q) \cdot M \geq \left(1 - \frac{\varepsilon}{10}\right) \cdot 2MR,$$

and the minimum number of erasures in both directions needed to produce π_j when Alice's input is z. In case that π_j is inconsistent with z regardless of the number of erasures, $c_j(z)$ is also defined to be B. Let Dec denote the procedure that, given $(\pi_1, \ldots, \pi_N)$, computes the cost functions $c_1, \ldots, c_N$ and applies the decoder promised by Lemma 3.2. We will show in Section 5 that the cost functions c_j indeed satisfy the conditions required by the decoder. The full protocol is given in Algorithm 2.

5 Analysis of Our Two-Way Code

This section constitutes the proof of Theorem 1.1. We first show that the cost functions c_j as defined in Section 4 do satisfy the conditions of Lemma 3.2.

To this end, fix $j \in [N]$ and $z_1 \neq z_2 \in [q]$. Consider two executions of Algorithm 1 that are consistent with π_j and start with z_1, z_2 respectively. For $r \in [R]$, define $S_1^{(r)}$ to be the

Algorithm 2 The two-way code to transmit $x \in \{0,1\}^n$.

Alice initializes $w \leftarrow \mathsf{Enc}(x)$.
for $j = 1, \ldots, N$ **do**
    Alice and Bob run Algorithm 1 with Alice holding input w_j. Let π_j be Bob's received transcript.
end for
Bob outputs $\mathsf{Dec}(\pi_1, \ldots, \pi_N)$.

state of S at the end of iteration r when Alice starts with input z_1. Similarly define $S_2^{(r)}$ for z_2. For notational convenience, also let $S_1^{(0)} = S_2^{(0)} = 0^M$.

Define the potential $\Phi^{(r)}$ to be the sum of $q \cdot \Delta(S_1^{(r)}, S_2^{(r)})$ and the total number of erasures up to iteration r in both executions. At the beginning, we have $\Phi^{(0)} = 0$. The following technical claim ensures that $\Phi^{(r)}$ increases steadily over iterations.

Claim 5.1. *For $r \in [R]$, it holds that*

$$\Phi^{(r)} - \Phi^{(r-1)} \geq 2M.$$

Proof. Consider any block $i \in [M/L_r]$ in iteration r. By definition, it has length L_r. We show that its contribution to the potential increase $\Phi^{(r)} - \Phi^{(r-1)}$ is at least $2L_r$. This suffices for proving the claim by summing over all blocks.

Observe that the potential increase comes from either erasures in the current iteration or increase in the Hamming distance, which is weighted by q. First consider the case that Bob does receive at least one bit in the current block, so $S_1^{(r-1)}$ and $S_2^{(r-1)}$ must have the same block i. This is because Algorithm 1 guarantees that all bits within each block are always the same. Bob sends $1^{L_r/2}$ in this case. If Alice receives at least one "1", then she flips $2 \cdot L_r/(2q) = L_r/q$ bits, increasing the Hamming distance by L_r/q. On the other hand, if Alice receives only erasures, then she flips $L_r/(2q)$ bits while all $L_r/2$ bits of Bob's feedback are erased. In both cases, the contribution to the potential increase is at least L_r. Moreover, $z_1 \neq z_2$ means that the flipped bits in the two executions are disjoint, implying $2L_r$ total contribution as claimed.

Now suppose that Bob receives none of the bits in the current block, so he sends $0^{L_r/2}$. Note that this already implies L_r erasures in each of the two executions. Regarding the Hamming distance, if Alice receives at least one "0", she simply keeps the block unchanged. Otherwise, all $L_r/2$ bits of Bob's feedback are erased, and Alice flips $L_r/(2q)$ bits. This decreases the Hamming distance by at most[1] $L_r/(2q)$. As a result, the negative contribution resulting from the decrease in the Hamming distance is always compensated by the additional erasures. Again note that the flipped bits in the two executions are disjoint. Consequently, the net contribution is at least $2L_r$ in total, concluding the proof. □

[1]The Hamming distance decreases only if $S_1^{(r-1)}$ and $S_2^{(r-1)}$ have different blocks i.

Since the Hamming distance $\Delta(S_1^{(r)}, S_2^{(r)})$ is always upper bounded by M, Claim 5.1 implies that eventually, the potential $\Phi^{(r)}$ will be mostly coming from erasures. This leads to the desired properties of the cost functions.

Lemma 5.2. *For $j \in [N]$ and $z_1 \neq z_2 \in [q]$, it holds that*

$$c_j(z_1) + c_j(z_2) \geq B.$$

Proof. By Claim 5.1 for all $r \in [R]$, together with the fact that $\Phi^{(0)} = 0$, we have that $\Phi^{(R)} \geq 2MR$. In the meantime, the contribution from the Hamming distance is at most qM. Thus, the total number of erasures in the two executions must be at least $\Phi^{(R)} - qM \geq B$. This concludes the proof because the executions are arbitrary so long as they are consistent with π_j. Capping the costs at B does not affect the inequality. □

Now, we are ready to prove Theorem 1.1.

Proof of Theorem 1.1. By Lemma 5.2, the cost functions satisfy the pairwise distance requirement for all pairs of distinct inputs z_1, z_2 to the inner code. Therefore, it suffices to show that the actual codeword $w = \mathsf{Enc}(x)$ transmitted by Alice satisfies $\sum_{j\in[N]} c_j(w_j) < (1-\delta)/2 \cdot BN$.

To see this, note that by definition, $c_j(w_j)$ is a lower bound on the actual number of erasures in the j-th execution of the inner code, for $j \in [N]$. In the meantime, we know that the total number of erasures is at most

$$\left(\frac{2}{3} - \varepsilon\right) \cdot T = \left(1 - \frac{3}{2}\varepsilon\right) \cdot MNR < \left(1 - \frac{\varepsilon}{10}\right) \cdot \left(1 - \frac{\varepsilon}{10}\right) \cdot MNR \leq \frac{1-\delta}{2} BN.$$

Consequently, the decoder $\mathsf{Dec}(\pi_1, \ldots, \pi_N)$ indeed recovers x by Lemma 3.2.

Regarding efficiency, note that all parameters q, M, R for the inner code are constant for any fixed ε. Therefore, evaluating the cost function $c_j(z)$ only takes constant time per entry, by enumerating over all possible erasure patterns. As B is also constant, Lemma 3.2 ensures that both encoding and decoding of the outer code run in linear time, as claimed. □

AI Methodology

The proof was discovered by ChatGPT-6 Astra. The authors simplified the proof and wrote the manuscript. The authors verified the correctness and originality of all content, including references.

References

[BEK+25] Mark Braverman, Klim Efremenko, Gillat Kol, Raghuvansh R. Saxena, and Zhijun Zhang. Round-vs-resilience tradeoffs for binary feedback channels. In Raghu Meka, editor, *Innovations in Theoretical Computer Science Conference (ITCS)*, pages 22:1–22:23, 2025. 2

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

[BGG20] Ferdinand Blomqvist, Oliver W. Gnilke, and Marcus Greferath. On decoding of generalized concatenated codes and matrix-product codes. *CoRR*, abs/2004.03538, 2020. 6, 7

[BR11] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *Symposium on Theory of Computing (STOC)*, pages 159–166, 2011. 3

[EKK+27] Klim Efremenko, Gillat Kol, Sergey Komech, Raghuvansh R. Saxena, and Zhijun Zhang. Two-way quantum codes with optimal erasure resilience. In *Symposium on Discrete Algorithms (SODA)*, 2027. 2

[EKS20] Klim Efremenko, Gillat Kol, and Raghuvansh R. Saxena. Binary interactive error resilience beyond $\frac{1}{8}$ (or why $(\frac{1}{2})^3 > \frac{1}{8}$). In *Symposium on Foundations of Computer Science (FOCS)*, pages 470–481, 2020. 3

[EKSZ22] Klim Efremenko, Gillat Kol, Raghuvansh R. Saxena, and Zhijun Zhang. Binary codes with resilience beyond $\frac{1}{4}$ via interaction. In *Symposium on Foundations of Computer Science (FOCS)*, pages 1–12, 2022. 1, 2

[FJ65] George David Forney Jr. *Concatenated codes*. PhD thesis, Massachusetts Institute of Technology, 1965. 6, 7

[GGZ23] Meghal Gupta, Venkatesan Guruswami, and Rachel Yun Zhang. Binary error-correcting codes with minimal noiseless feedback. In *Symposium on Theory of Computing (STOC)*, pages 1475–1487, 2023. 2

[GI05] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Trans. Inf. Theory*, 51(10):3393–3400, 2005. 6, 7

[GKZ22] Meghal Gupta, Yael Tauman Kalai, and Rachel Yun Zhang. Interactive error correcting codes over binary erasure channels resilient to $> \frac{1}{2}$ adversarial corruption. In *Symposium on Theory of Computing (STOC)*, pages 609–622, 2022. 0, 1, 2

[GZ22] Meghal Gupta and Rachel Yun Zhang. The optimal error resilience of interactive communication over binary channels. In *Symposium on Theory of Computing (STOC)*, pages 948–961, 2022. 3

[GZ23a] Meghal Gupta and Rachel Yun Zhang. Efficient interactive coding achieving optimal error resilience over the binary channel. In *Symposium on Theory of Computing (STOC)*, pages 1449–1462, 2023. 3

[GZ23b] Meghal Gupta and Rachel Yun Zhang. Interactive error correcting codes: New constructions and impossibility bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM)*, pages 32:1–32:14, 2023. 1, 2

[HKV15] Bernhard Haeupler, Pritish Kamath, and Ameya Velingker. Communication with partial noiseless feedback. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, APPROX/RANDOM 2015, pages 881–897, 2015. 2

[Plo60] Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960. 1

[Sax27] Raghuvansh R. Saxena. Adaptive two-way codes for binary erasures. In *Symposium on Discrete Algorithms (SODA)*, 2027. 2

[Sch92] Leonard J. Schulman. Communication on noisy channels: A coding theorem for computation. In *Symposium on Foundations of Computer Science (FOCS)*, pages 724–733, 1992. 3

[Sch93] Leonard J. Schulman. Deterministic coding for interactive communication. In *Symposium on Theory of Computing (STOC)*, pages 747–756, 1993. 3

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. 3

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

[WQC17] Gang Wang, Yanyuan Qin, and Chengjuan Chang. Communication with partial noisy feedback. In *IEEE Symposium on Computers and Communications*, pages 602–607, 2017. 2

[Zig76] K. Sh. Zigangirov. On the number of correctable errors for transmission over a binary symmetrical channel with feedback. *Problems of Information Transmission*, 12(2):85–97, 1976. 2

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