On the Power of Feedback in Interactive Channels

Denis Pankratov*

Abstract

In classical information theory it is well-known that feedback does not improve the channel capacity. We demonstrate that this is not the case in the interactive setting by developing a new coding scheme for interactive communication over the binary symmetric channel (BSC) with feedback that allows recovery of the original communication with vanishing probability of error. More precisely, we show that the interactive channel capacity of BSC with feedback and bit-flip probability ϵ is at least $1 - O(\sqrt{\epsilon})$, while the upper bound on BSC without feedback was recently shown to be $1 - \Omega(\sqrt{H(\epsilon)})$ by Gillat Kol and Ran Raz.

1 Introduction

Alice and Bob communicate over a noisy channel. What is the best way to encode the communication, so that Alice and Bob can recover the intended conversation with a vanishing probability of error? The length of the intended conversation over the length of the encoded conversation is called *the rate*. The supremum over rates, at which the communication happens reliably, is known as *the channel capacity*. Much is known about the one-way version of channel capacity, as it is the central object of study in classical information theory established by Shannon [13] in 1948. In particular, the famous Shannon's noisy-channel coding theorem provides a complete mathematical characterization of the capacity of the channel in the one-way setting. Interactive setting is much less studied and no analogue of Shannon's theorem is known in this case.

The question of how to encode interactive communication to resist channel errors was first considered by Schulman [12] in 1996. The first difficulty in the interactive setting is that a single mistake in the earlier rounds of communication can completely derail the rest of the conversation. The second difficulty is that the encoding has to happen online, since each message depends on the history of communication. A standard way of simplifying a difficult problem in communication complexity is to restrict the number of rounds of communication. However, for the coding problem restricting the number of rounds reduces interactive setting to the one-way setting, as it is possible to encode the original message round-by-round. The interesting case occurs when the number of rounds is $\Omega(n)$ where n is the length of the intended conversation. Then round-by-round encoding has to provide strong guarantees (depending on n) on the error-tolerance of the earlier rounds. This necessarily increases communication by a factor depending on n. Thus, a priori, it is not even clear that the channel capacity is bounded away from zero. In spite of all these obstacles, Schulman showed how to encode conversations using tree codes so that the communication increases by a constant factor and the intended messages can be recovered safely even when constant fraction $\left(\frac{1}{240}\right)$ of communicated bits are corrupted *adversarially*. This implies that the interactive channel capacity is, indeed, bounded away from

*University of Chicago, Department of Computer Science. Email: pankratov@cs.uchicago.edu.

0. This approach to the problem of interactive coding against *adversarial noise* was further studied in [6, 11, 3, 2, 1].

The problem of encoding interactive communication in the *probabilistic-noise model* is subtly different from the problem of encoding conversation to resist adversarial noise. The desired goal for the former problem is obtaining noise-resilient encodings of length $n(1 + f(1/\epsilon))$, where n is the length of the original communication, ϵ is the noise parameter of the channel, and $f(1/\epsilon)$ is some function $o(1)$. The leading term in the length of the encoding has to be n. The known results from the adversarial-noise model provide encodings of length $\Theta(n)$ with hidden and often large constants in front of the n-term. Thus the results from the adversarial-noise model fall short of the desired bounds on the length of encoding in the probabilistic setting to achieve capacity.

We consider two models of probabilistic noise in this paper.

Binary symmetric channel (BSC). The channel has binary input and binary output $\{0, 1\}$, and the transmitted bit gets flipped with probability ϵ.

Binary erasure channel (BEC). The channel has binary input $\{0, 1\}$ and ternary output $\{0, 1, *\}$, and the transmitted bit gets replaced by $*$ with probability ϵ.

One-way channel capacity can be computed exactly using Shannon's noisy-channel coding theorem. The one-way channel capacity of BSC is $1 - H(\epsilon)$, where H is the binary entropy. The channel capacity of the BEC is $1 - \epsilon$. In 1996 Schulman [12] asked whether the interactive channel capacity can be smaller than the one-way channel capacity for some channels. This question remained unanswered until the recent breakthrough result due to Kol and Raz [8]. They prove tight bounds on the interactive channel capacity of the BSC, separating interactive channel capacity from its one-way analogue.

Theorem 1.1 (Kol and Raz [8])**.** *Interactive channel capacity of BSC with bit-flip probability ϵ is $\leq 1 - \Omega(\sqrt{H(\epsilon)})$. In the case where players take alternating turns the interactive channel capacity of BSC is lower bounded by $1 - O(\sqrt{H(\epsilon)})$.*

In many practical systems the transmitter receives *noiseless feedback* from the channel, i.e., how the transmitted bit was received by the receiver (see Figure 1). We denote channels with feedback by appending a letter "f" to the end of the corresponding acronym: BSCf and BECf.

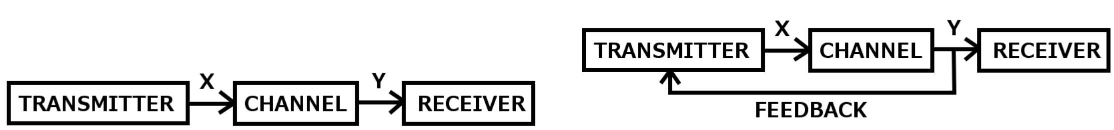

Figure 1: Regular channel vs channel with feedback.

The main conceptual difference between BSC, BEC, BSCf, and BECf lies in the awareness of mistakes during transmission. See the table below.

	Transmitter	Receiver
BSC	✗	✗
BSCf	✓	✗
BEC	✗	✓
BECf	✓	✓

Table 1: Who can detect an error in the transmission?

The simplest channel to deal with is BECf, since both players know exactly when the mistake happens, so the transmitter can keep resending the bit until it goes through the channel. This lower bound together with a matching upper bound in the interactive setting was first observed by Schulman [12].

Theorem 1.2 (Schulman [12])**.**

$$\mathrm{CAP}_{\mathrm{BECf}}(\epsilon) = 1 - \epsilon.$$

A surprising fact in classical information theory is that the feedback does not increase the capacity of the one-way channel (see, for example, [4]). Thus, the one-way channel capacity of BSCf is $1 - H(\epsilon)$ and the one-way channel capacity of BECf is $1 - \epsilon$.

Does feedback help in the interactive setting? We answer this question in the positive by providing a new coding theorem for the interactive communication over BSCf. Our main result is the following theorem.

Theorem 1.3 (Main Theorem)**.**

$$\mathrm{CAP}_{\mathrm{BSCf}} \geq 1 - O(\sqrt{\epsilon}).$$

Together with the recent upper bound $\mathrm{CAP}_{\mathrm{BSC}} \leq 1 - \Omega(\sqrt{H(\epsilon)})$ due to Kol and Raz [8] this demonstrates that the feedback can increase the channel capacity in the interactive setting.

Remark 1.4. The best upper bound on the interactive channel capacity of the BSCf is $1 - H(\epsilon)$ given by Shannon's theorem.

Remark 1.5. Contrast our unconditional lower bound (Theorem 1.3) on the $\mathrm{CAP}_{\mathrm{BSCf}}$ with the lower bound of Kol and Raz on $\mathrm{CAP}_{\mathrm{BSC}}$ (Theorem 1.1), where players are assumed to take alternating turns. Feedback allows the players to behave consistently while simulating the intended protocol, so that they never try to transmit at the same time even in the presence of errors in the conversation. This cannot be guaranteed for the BSC without feedback unless we restrict the intended protocol to have a certain structure. The errors are probabilistic, so any transcript can be turned into any other transcript without the knowledge of the players.

The rest of the paper is organized as follows. In Section 2 we provide definitions and the necessary background for the rest of the paper. In Section 3 the coding theorem is presented in four steps: (1) simulating protocol is described for *uniform protocols*, i.e., protocols with transcripts of equal length on every input, (2) the simulating protocol is reformulated as a random walk, (3) the simulating protocol is analyzed, (4) uniformity assumption is dropped. The paper ends with conclusions and discussion of open problems in Section 4.

2 Preliminaries

We shall require only the basic definitions of communication complexity, which we present in the current section. For a thorough treatment of communication complexity the reader is referred to the classical monograph due to Kushilevitz and Nisan [9]. For a basic introduction to the classical information theory the reader is referred to the textbook of Cover and Thomas [4] and another textbook of MacKay [10] (available online for free from MacKay's homepage).

In two-party interactive communication the parties are traditionally called Alice and Bob. The goal of the players is to compute a given function $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$. Alice is given input $X \in \mathcal{X}$, Bob is given input $Y \in \mathcal{Y}$, where (X, Y) are jointly distributed according to some distribution μ on $\mathcal{X} \times \mathcal{Y}$. The players have access to an infinite public string R of random bits.

FIrst consider the scenario, in which the players communicate over perfect channels. At each round, the protocol specifies the current speaker based on the communication history so far, i.e., *the transcript*. A protocol, in which only one player speaks at a time, is called *valid*. All protocols are valid over perfect channels, but this is not necessarily the case for imperfect channels. If Alice is the current speaker, she sends a message based on her input X, random string R, and the transcript. If Bob is the current speaker, he sends a message depending on Y, R, and the transcript so far. At the end of the protocol both players agree on an output Z'. The *average cost of the protocol*, denoted $\mathrm{CC}(\pi)$, is the expected number of bits exchanged during the execution of π, where the expectation is over μ and R. Protocol π solves f with error at most ϵ if for every input (x, y) the probability that π produces incorrect output on (x, y) is at most ϵ. Let $\mathrm{CC}(f)$ denote the average cost of the best protocol computing f with vanishing probability of error.

Now consider the scenario, in which the players communicate with each other over a binary symmetric channel with probability of bit-flip $\epsilon > 0$. The channel introduces errors probabilistically, so there is a non-zero probability of an intended transcript being transformed to any other transcript. We still require protocols to be valid. Let $\mathrm{CC}_{\mathrm{BSC}}(f, \epsilon)$ denote the average cost of the best valid protocol computing f with vanishing probability of error. When the channel is equipped with feedback, we denote this quantity by $\mathrm{CC}_{\mathrm{BSCf}}(f, \epsilon)$. One way of guaranteeing the validity of the protocol is to force the players specify before the execution of the protocol who is going to speak at each step. This is why Kol and Raz [8] introduce structure to the protocols in their lower bound. However, when the feedback is present we shall see that the players can maintain the validity constraint without specifying the speakers in advance.

Definition 1. The *interactive capacity of the BSC* with probability of bit-flip ϵ is defined as

$$\mathrm{CAP}_{\mathrm{BSC}}(\epsilon) := \lim_{n\to\infty} \min_{\{f:\mathrm{CC}(f)=n\}} \frac{n}{\mathrm{CC}_{\mathrm{BSC}}(f, \epsilon)}.$$

For BSCf with probability of bit-flip ϵ the capacity is defined similarly

$$\mathrm{CAP}_{\mathrm{BSCf}}(\epsilon) := \lim_{n\to\infty} \min_{\{f:\mathrm{CC}(f)=n\}} \frac{n}{\mathrm{CC}_{\mathrm{BSCf}}(f, \epsilon)}.$$

In the paper we shall work with the deterministic protocols only. Our result easily extends to the randomized protocols with public randomness by viewing such protocols as distributions over the deterministic protocols.

A deterministic protocol over a perfect channel can be represented as a communication tree. Each node v has an owner O_v, who can be either Alice or Bob, and a function f_v, which specifies a bit to transmit based on the owner's input. If $O_v =$Alice, the function f_v is of the form $f_v : \mathcal{X} \to \{0, 1\}$; otherwise, it is of the form $f_v : \mathcal{Y} \to \{0, 1\}$. Each internal node has exactly two outgoing edges: one labelled 0 and one labelled 1. Each leaf is labelled by some $z \in \mathcal{Z}$. To execute π in this representation, the players start at the root and proceed as follows. The owner of the current node v evaluates f_v on their input and transmits the resulting bit to the other party. Both players update the current node by following the corresponding edge out of v. When the players get to a leaf, the protocol terminates and the players output the label of the leaf.

3 Coding Theorem

A new coding theorem for interactive communication over the BSCf is proved in this section.

Lemma 3.1. *For all sufficiently small $\epsilon > 0$ for every deterministic communication protocol π with $n := \mathrm{CC}(\pi)$ there exists deterministic protocol τ such that*

- *communication in τ is over the binary symmetric channel with feedback and bit-flip probability $\epsilon > 0$,*
- $\mathrm{CC}(\tau) \le n(1 + O(\sqrt{\epsilon}))$,

- *at the end of τ both players output correct leaf of π with probability approaching 1 as n approaches infinity.*

Theorem 1.3 (Main Theorem) is an immediate consequence of Lemma 3.1. Lemma 3.1 is proved in three steps. First, it is proved under the *uniformity assumption* on π, i.e., that the transcript length of π is the same on every input. In Section 3.1 simulating protocol is presented. In Section 3.2 the simulating protocol is shown to be a random walk on a specific tree. The properties of the simulating protocol are analyzed and proved in Section 3.3. The uniformity assumption is dropped in Section 3.4.

3.1 Simulating Protocol

Let π be a deterministic protocol over the perfect channel, such that it terminates after exactly n steps. Fix inputs $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Let L be the unique leaf, which is reached by the players after executing π on inputs (x, y). We describe a protocol τ, in which the players communicate over the BSCf with bit-flip probability ϵ, and at the end of communication the players output L except with probability negligible in n.

Let $k, m \in \mathbb{N}$ be parameters to be specified later. The players augment the communication tree of π by replacing each leaf with an infinite complete binary tree, in which every node is labelled with the label of the replaced leaf. Call the augmented tree T.

Players keep track of the current node v in T. Initially v is set to the root of T.

The players repeat the following steps m times:

> The players repeat the following steps k times:
>
> > The current owner of v, evaluates f_v on the given input, transmits the output bit over the noisy channel to the other party.
> >
> > The players update the current node v to the new node according to the value of the bit after it was transmitted through the channel.
> >
> > The transmitter of the bit remembers if the bit got flipped by the channel.
>
> Alice sends a single verification bit B_A, where B_A is set to 0 if there were no mistakes in her transmissions leading from the root of T to the current node v and 1 otherwise.
>
> Bob sends a single verification bit B_B, where B_B is set to 0 if there were no mistakes in his transmissions leading from the root of T to the current node v and 1 otherwise.
>
> Alice takes a pair of bits - the received Bob's verification bit and her own verification bit, *as it was received by Bob (using feedback).*
>
> Bob Takes a pair of bits - the received Alice's verification bit and his own verification bit, *as it was received by Alice (using feedback).*
>
> Note that Bob computes exactly the same pair of bits as Alice does.
>
> If not both of the bits are 0 the players backtrack the current node v by $2k$ positions, where backtracking from the root is accomplished by staying put. While backtracking the players forget which simulated bits were transmitted incorrectly in the last $2k$ positions, so that they are not reused in the future calculations of verification bits.

If the current node v is labelled by a leaf of the original communication tree of π, the players output the label of that leaf. Otherwise, the players output an arbitrary label.

Protocol 1: Protocol τ simulating π over the BSCf.

3.2 Random Walk View

The simulating protocol from Section 3.1 can be viewed as a random walk on the tree T_R shown in Figure 2.

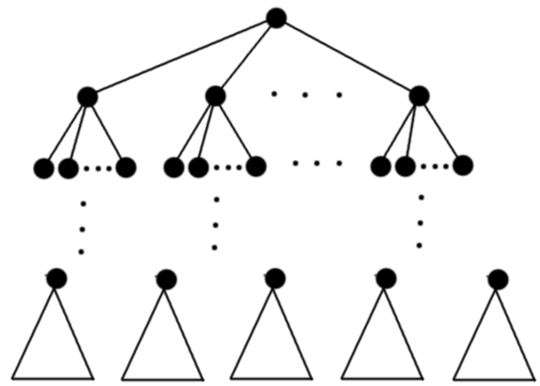

Figure 2: Tree T_R. The simulating protocol can be viewed as a random walk on this tree.

Each triangle represents an infinite binary subtree replacing leaves of the communication tree of π in T. Disregarding these triangles, T_R consists of at most $\frac{n}{k}$ levels. Each node in these levels has arity at most 2^k. Nodes at level i are precisely the nodes of T at level ki. The edges connecting nodes at level i and level $i+1$ are labelled by the concatenation of all messages exchanged on the path connecting the corresponding nodes in T. The tree T_R can be viewed as a collapsed version of T, where only the nodes corresponding to checkpoints are included. The checkpoint is a node, at which the verification bits can be exchanged in T. The simulating protocol performs a random walk on this tree for m steps. If after m steps, the random walk ends up in one of the triangles, the players output the corresponding label. In Section 3.3 we shall fix parameters m and k that guarantee small communication overhead and large probability of success.

3.3 Analysis

Proof. Let k be such that $\epsilon = \frac{1}{k^2}$. Choose $m = \frac{n}{k}(1 + 64\sqrt{\epsilon})$. Recall that protocol π terminates after exactly n steps on every input. Thus we can fix the input to be (x, y), which in turn determines the correct leaf L of π reached by players on the noiseless channel.

In the rest of the proof the random walk from Section 3.2 is analyzed. The random walk essentially travels along the path from the root of T_R to the infinite subtree labelled by L. Occasionally the random walk branches off this path, but rather soon it returns back to the node where the branching took place. Traveling downward from the root of T_R to L corresponds to correct simulation of k bits of π and correct verification bits. Traveling upward along the path from L to the root of T_R corresponds to either correct simulation of k steps of π and incorrect verification bits or incorrect simulation of k steps of π and correct verification bits. Lastly, branching off the path from the root of T_R to L corresponds to incorrect simulation of k bits of π followed by incorrect verification bits.

The probability that the random walk branches off the path from the root to L is bounded by the probability of incorrect verification bits, i.e., either Alice's verification bit got flipped or Bob's verification bit got flipped. This happens with probability at most 2ϵ. If the players branch off the path from the root of T_R to L, there was an error in the transcript. Since the players send verification bits for the entire transcripts up until the current node, the probability that the branched-off random walk will "dig deeper" into the tree T_R is also upper bounded by the probability of incorrect verification bits. A step of the random walk is called *bad* if either it branches off the path from the root of T_R to L or it is going deeper in the wrong branch. Otherwise a step is called *good*. Define random variable

$$Y_i = \begin{cases} 1 & \text{if } i\text{th step is bad} \\ 0 & \text{otherwise} \end{cases}$$

Thus $P(Y_i = 1) \leq 2\epsilon$ and by Chernoff bound $P(\sum Y_i \geq 8\epsilon m) \leq e^{-6\epsilon m} \stackrel{n\to\infty}{\to} 0$. Thus with high probability we have at most $8\epsilon m$ bad moves, which are undone by at most $8\epsilon m$ other moves and we have $m_c := m - 16\epsilon m = \frac{n}{k}(1 + 32\sqrt{\epsilon})$ moves along the path from the root of T_R to L (provided that $\epsilon \leq \frac{1}{64}$). It is left to show that with at least m_c moves along the path from the root of T_R to L, the random walk is likely to end up in the infinite subtree labelled with L.

Define the following random variable for the ith *good* step.

$$X_i = \begin{cases} 1 & \text{if backtracking occurs} \\ -1 & \text{otherwise} \end{cases}$$

In order for backtracking to happen, at least one of the $k + 2$ communicated during the step bits has to get flipped. Thus $p := P(X_i = 1) \leq (k + 2)\epsilon \leq 4\sqrt{\epsilon}$.

Observe that the protocol outputs the correct leaf as long as the number of correct steps along the path from root to L exceeds the number of incorrect steps by at least n/k. Thus the probability of outputting incorrect leaf given that the protocol makes at least m_c good steps is upper bounded by

$$P\left(\sum X_i > -\frac{n}{k}\right) \leq \exp(-m_c(1/2 - p - n/(2m_c k))^2/3p),$$

where the inequality follows from Chernoff bound. We have

$$\begin{aligned}
\tfrac{1}{2} - p - \tfrac{1}{2}\tfrac{n}{k}\tfrac{1}{m_c} &= \tfrac{1-2p}{2} - \tfrac{1}{2(1+32\sqrt{\epsilon})} \\
&\geq \tfrac{1}{2(1+32\sqrt{\epsilon})}((1 - 8\sqrt{\epsilon})(1 + 32\sqrt{\epsilon}) - 1) \\
&\geq \tfrac{1}{2(1+32\sqrt{\epsilon})}(24\sqrt{\epsilon} - 256\epsilon) \\
&\geq \tfrac{8\sqrt{\epsilon}(1-32\sqrt{\epsilon})}{2(1+32\sqrt{\epsilon})} \\
&\geq \tfrac{4}{3}\sqrt{\epsilon},
\end{aligned}$$

where the last step holds provided $\epsilon \leq \frac{1}{1024}$. Hence under these conditions the probability of incorrect output by τ is upper bounded by $e^{-\Omega_\epsilon(m)}$. In particular the probability of incorrect output approaches 0 as $n \to \infty$.

Lastly, the number of bits exchanged by the protocol is bounded by

$$m(k + 2) = \frac{n}{k}(1 + 64\sqrt{\epsilon})(k + 2) = n(1 + 64\sqrt{\epsilon})(1 + 2\sqrt{\epsilon}) = n(1 + O(\sqrt{\epsilon})).$$

□

3.4 Dropping Uniformity Assumption

If protocol π does not satisfy the uniformity assumption then the length of the transcript, which we denote by $C(X, Y)$, is a random variable depending on the sampled inputs X and Y. The terminating condition of Protocol 1 is modified as follows: *instead of performing the outer loop m times, the players perform the outer loop until they reach depth $100 \log n$ of some infinite subtree* [1] *for the first time. If the players communicate over $100n$ bits they terminate regardless of whether the required depth has been reached in an infinite subtree.*

[1] Recall: these subtrees replace leaves of the communication tree of original protocol π.

One can show that except with probability negligible in n the protocol terminates and outputs the correct leaf in $\frac{C(X,Y)}{k}(1+64\sqrt{\epsilon})+\frac{1024\log n}{k}$ steps. Thus, except with probability negligible in n, the protocol communicates

$$\mathbb{E}_{X,Y}\left(\frac{C(X,Y)}{k}(1+64\sqrt{\epsilon})+\frac{1024\log n}{k}\right)(k+2)=n(1+O(\sqrt{\epsilon}))$$

in expectation. And with negligible probability in n they communicate at most $100n$ bits. Therefore, the average cost of the modified τ is still $n(1+O(\sqrt{\epsilon}))$ and it outputs the correct answer with negligible probability of error.

4 Conclusions and Open Problems

A new coding theorem for the interactive communication over the binary symmetric channel with feedback was presented. This theorem together with the recent upper bound on the capacity of the BSC due to Kol and Raz [8] demonstrates that the feedback can increase the interactive channel capacity, unlike the classical channel capacity. In contrast with the coding theorem in [8], the coding theorem in this paper works unconditionally. This coding theorem was motivated by the idea of noisy comparison trees due to Feige et al. [5]. The summary of results about the interactive channel capacity of the binary symmetric channel is shown in Table 2

	Lower bound	Upper bound
BSC	$1-O(\sqrt{H(\epsilon)})$ (alternating turns) [8]	$1-\Omega(\sqrt{H(\epsilon)})$ [8]
BSCf	$1-O(\sqrt{\epsilon})$ (this paper)	$1-H(\epsilon)$ [13]

Table 2: Interactive channel capacity of the binary symmetric channel, summary of results.

In the rest of this section we list a few open problems related to this area of research, starting with:

Open Problem 4.1. *Close the gap between the upper bound and the lower bound on* $\mathrm{CAP}_{\mathrm{BSCf}}$.

It is possible that the technique of Kol and Raz [8] can be used to improve the upper bound. However, their technique is complicated. The area would benefit greatly from developing new techniques for giving strong upper bounds on the capacity of the interactive channel.

Open Problem 4.2. *Develop new techniques for proving strong upper bounds on the interactive channel capacity.*

As for improving the lower bound of BSCf, the coding scheme presented in this paper is limited by the overall overhead on communication of $(1+\Omega\left(\frac{1}{k}\right))(1+\Omega(k\epsilon))$, which is optimized for $k=\Theta\left(\frac{1}{\sqrt{\epsilon}}\right)$. Another possible way of attacking the problem would be to adapt the classical online coding scheme for the one-way communication over BSCf due to Horstein [7]. In that coding scheme, the transmitter encodes the message as a binary decimal number. The receiver keeps track of the current guess of the message, and the transmitter communicates whether the receiver's guess is to the left of the proper message or to the right of the proper message.

Another open problem is analyzing other types of interactive channels to better understand the power of feedback in the interactive channel capacity.

Open Problem 4.3. *Analyze other types of channels and give a characterization of how much the feedback helps for a given channel. Can anything be said in general?*

The binary erasure channel is an interesting starting point for the above question. The best known upper bound is $1-\epsilon$, and it follows from Shannon's theorem. It seems that the coding scheme of [8] can be used to show $1-O(\sqrt{\epsilon})$ lower bound[2] for the alternating-speakers regime. However, neither of these bounds seems to be the right answer for the interactive channel capacity of the BEC.

Open Problem 4.4. *Find the exact interactive channel capacity of the BEC with erasure-probability* ϵ*.*

Even improving the upper bound to $1-c\epsilon$ for some $c>1$ would be interesting, as it might shed some light on Open Problem 4.2. As for the lower bound, one should be looking for a coding scheme with $1+o(\sqrt{\epsilon})$ overhead in communication.

Shannon's theorem can be viewed as proving equivalence between the informational characterization of the one-way channel capacity (it is sometimes referred to as a "single-letter" characterization in the field of electrical engineering) and the operational definition of the channel capacity. The big goal is to find an analogue, if it exists, of Shannon's theorem in the interactive setting. Definition 1 gives the operational definition of the interactive channel capacity. What is the proper informational, i.e., single-letter, description of the same quantity?

Open Problem 4.5. *Find an analogue of Shannon's theorem in the interactive setting, i.e., find an informational characterization of the operational definition of the interactive channel capacity.*

Acknowledgements
The author thanks Mark Braverman for pointing out the paper of Gillat Kol and Ran Raz, which was the inspiration for this work. The author is also grateful to the anonymous reviewers. Their thoughtful comments significantly improved the quality of this paper.

References

[1] Zvika Brakerski and Yael Tauman Kalai. Efficient interactive coding against adversarial noise. In *Proc. of the 53rd FOCS*, pages 160–166, 2012.

[2] Mark Braverman. Towards deterministic tree code constructions. In *Proc. of the 3rd ITCS*, pages 161–167, 2012.

[3] Mark Braverman and Anup Rao. Towards coding for maximum errors in interactive communication. In *Proc. of the 43rd STOC*, pages 159–166, 2011.

[4] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006.

[5] Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM J. Comput.*, 23(5):1001–1018, 1994.

[6] R. Gelles and A. Sahai. Potent tree codes and their applications: Coding for interactive communication, revisited. In *Arxiv preprint arXiv:1104.0739*, 2011.

[7] M. Horstein. *Sequential Transmission of Digital Information with Feedback.* Group report. Massachusetts Institute of Technology, Department of Electrical Engineering, 1960.

[8] Gillat Kol and Ran Raz. Interactive channel capacity. *Electronic Colloquium on Computational Complexity (ECCC)*, 2013.

[2]This removes the logarithmic factor from the $O(\sqrt{H(\epsilon)})$ term in the bound for BSC. In the coding theorem of [8] the players *detect* errors using $O(\log k)$ hashes, but when communicating over BEC the players already *know* the errors making the hashes redundant.

[9] Eyal Kushilevitz and Noam Nisan. *Communication complexity.* Cambridge University Press, New York, NY, USA, 1997.

[10] David J. C. MacKay. *Information Theory, Inference & Learning Algorithms.* Cambridge University Press, New York, NY, USA, 2002.

[11] A. Moitra. Efficiently coding for interactive communication. In *Electronic Colloquium on Computational Complexity (ECCC)*, 2011.

[12] Leonard J. Schulman. Coding for interactive communication. In *Proc. of the 25th STOC*, pages 747–756, 1996.

[13] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.