Optimal Error Rates for Interactive Coding I: Adaptivity and Other Settings

Mohsen Ghaffari
MIT
ghaffari@mit.edu

Bernhard Haeupler
Microsoft Research
haeupler@cs.cmu.edu

Madhu Sudan
Microsoft Research
madhu@mit.edu

ABSTRACT

We consider the task of interactive communication in the presence of adversarial errors and present tight bounds on the tolerable error-rates in a number of different settings.

Most significantly, we explore *adaptive* interactive communication where the communicating parties decide who should speak next based on the history of the interaction. In particular, this decision can depend on estimates of the amount of errors that have occurred so far. Braverman and Rao [STOC'11] show that non-adaptively one can code for any constant error rate below $1/4$ but not more. They asked whether this bound could be improved using adaptivity. We answer this open question in the affirmative (with a slightly different collection of resources): Our adaptive coding scheme tolerates any error rate below $2/7$ and we show that tolerating a higher error rate is impossible. We also show that in the setting of Franklin et al. [CRYPTO'13], where parties share randomness not known to the adversary, adaptivity increases the tolerable error rate from $1/2$ to $2/3$. For list-decodable interactive communications, where each party outputs a constant size list of possible outcomes, the tight tolerable error rate is $1/2$.

Our negative results hold even if the communication and computation are unbounded, whereas for our positive results communication and computations are *polynomially* bounded. Most prior work considered coding schemes with *linear* communication bounds, while allowing unbounded computations. We argue that studying tolerable error rates in this relaxed context helps to identify a setting's intrinsic optimal error rate. We set forward a strong working hypothesis which stipulates that for any setting the maximum tolerable error rate is independent of many computational and communication complexity measures. We believe this hypothesis to be a powerful guideline for the design of simple, natural, and efficient coding schemes and for understanding the (im)possibilities of coding for interactive communications.

General Terms

Theory, Algorithms

STOC '14, May 31 - June 03 2014, New York, NY, USA.

ACM 978-1-4503-2710-7/14/05 ...$15.00.
http://dx.doi.org/10.1145/2591796.2591872.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Formal Models of Communication; F.2.0 [**Analysis of Algorithms and Problem Complexity**]: General

Keywords

Coding for Interactive Communication, Adversarial Errors, Error Rate, Adaptivity

1. INTRODUCTION

"Interactive Coding" or "Coding for Interactive Communication" studies the task of protecting an interaction between two parties in the presence of communication errors. This line of work was initated by Schulman [12] who showed, surprisingly at the time, that protocols with n rounds of communication can be protected against a (small) constant fraction of adversarial errors while incurring only a constant overhead in the total communication complexity.

In a recent powerful result that revived this area, Braverman and Rao [7] explored the maximal error rate that can be tolerated in an interactive coding setting. They showed the existence of a protocol that handles a $1/4 - \epsilon$ error rate and gave a matching negative result under the assumption that the coding scheme is *non-adaptive* in deciding which player transmits (and which one listens) at any point of time. They left open the questions whether the $1/4$ error rate can be improved by allowing adaptivity (see [4, Open Problem 7] and [7, Conclusion]) or by reducing the decoding requirement to list decoding (see [4, Open Problem 9] and [7, Conclusion]), that is, requiring each party only to give a small list of possible outcomes, of which one has to be correct.

In this work we answer both questions in the affirmative (in a somewhat different regime of computational and communication resources): We give a rate adaptive coding scheme that tolerates any error rate below $2/7$. We furthermore show matching impossibility result which strongly rules out any coding scheme achieving an error rate of $2/7$.

Moreover, we also consider the adaptive coding schemes in the setting of [9] in which both parties share some randomness not known to the adversary. While non-adaptive coding schemes can tolerate any error rate below $1/2$ this bound increases to $2/3$ using adaptivity, which we show is also best possible.

Lastly, we initiate the study of list decodable interactive communication. We show that allowing both parties to output a constant size list of possible outcomes allows non-adaptive coding schemes that are robust against any error rate below $1/2$, which again is best possible for both the adaptive and non-adaptive setting.

All our coding schemes are deterministic and work with communication and computation being polynomially bounded in the length of the original protocol. We note that most previous works considered the more restrictive setting of linear amount of communication (often at the cost of exponential time computations). Interestingly, our matching negative results hold even if the communication and computation are unbounded. We show that this sharp threshold behavior extends to many other computational and communication complexity measures and is common to all settings of interactive communication studied in the literature. In fact, an important conceptual contribution of this paper is the formulation of a strong working hypothesis that stipulates that maximum tolerable error rates are invariable with changes in complexity and efficiency restrictions on the coding scheme. This hypothesis suggests to first consider the simpler settings and then expand on the insights derived to get more general results. We believe that in this way, the working hypothesis yields a powerful guideline for the design of simple and natural coding schemes as also the search for negative results. This has been already partially substantiated by subsequent results (see[10] and Appendix B of the full version[11]).

Organization: In what follows, we provide the interactive communication model in Section 2. We also introduce the model for adaptive interaction there. Then, in Section 3, we explain our results as well as the underlying high-level ideas and techniques. In Section 4 we describe the simple Exchange problem and give an adaptive protocol tolerating a 2/7-fraction of errors in Section 4.1. In the remainder of Section 4, we prove that the error-rate of 2/7 is the best achievable for the Exchange problem and therefore also for the general case of interactive communication. In Section 5, we give interactive coding schemes over large alphabets tolerating a 2/7 error rate for general interactions.

2. INTERACTIVE CODING SETTINGS

In this section, we define the classical interactive coding setup as well as all new settings considered in this work, namely, *list decoding*, the *shared randomness* setting, and *adaptive protocols*.

An n-round *interactive protocol* Π between two players Alice and Bob is given by two functions Π_A and Π_B. For each *round* of communication, these functions map (possibly probabilistically) the history of communication and the player's private input to a decision of whether to listen or transmit, and in the latter case also to a symbol of the *communication alphabet*. All protocols studied prior to this work are *non-adaptive* in that the decision of a player to listen or transmit deterministically depends only on the round number, ensuring that exactly one party transmits in each round (in In [7] such protocols were called robust). In this case, the *channel* delivers the chosen symbol of the transmitting party to the listening party, unless the adversary interferes and alters the symbol arbitrarily. In the adversarial channel model with *error rate* ρ, the number of such errors is at most ρn. The outcome of a protocol is defined to be the transcript of the interaction.

A protocol Π' is said to *robustly simulate* a protocol Π for an error rate ρ if the following holds: Given any inputs to Π, both parties can *uniquely decode* the transcript of an error free execution of Π on these inputs from the transcript of any execution of Π' in which at most a ρ fraction of the transmissions were corrupted. This definition extends easily to *list decoding* by allowing both parties to produce a small (constant size) list of transcripts that is required to include the correct decoding, i.e., the transcript of Π. We note that the simulation Π' typically uses a larger alphabet and a larger number of rounds. While our upper bounds are all deterministic, we strengthen the scope of our lower bounds by also considering *randomized protocols* in which both parties have access to independent private randomness. We also consider the setting of [9] in which both parties have access to *shared randomness*. In both cases we assume that the adversary does not know the randomness and we say a randomized protocol robustly simulates a protocol Π with *failure probability* p if, for any input and any adversary, the probability that both parties correctly (list) decode is at least $1 - p$.

We now present the notion of an *adaptive* protocol. Defining a formal model for adaptivity leads to several subtle issues. We define the model first and discuss the issues later.

In an *adaptive* protocol, the communicating players are allowed to base their decision on whether to transmit or listen (probabilistically) on the communication history. In particular, this allows players to base their decision on estimates of the amount of errors that have happened so far (see Section 3.1 for why this kind of adaptivity is a natural and useful approach). This can lead to rounds in which both parties transmit or listen simultaneously. In the first case no symbols are delivered while in the latter case the symbols received by the two listening parties are chosen by the adversary, without it being counted as an error.

Discussion on the adaptivity model: It was shown in [7] that protocols which under no circumstances have both parties transmit or listen simultaneously are necessarily non-adaptive. Any model for adaptivity must therefore allow parties to simultaneously transmit or listen and specify what happens in either case. Doing this and also deciding on how to measure the amount of communication and the number of errors leads to several subtle issues.

While it seems pessimistic to assume that the symbols received by two simultaneously listening parties are determined by the adversary, this is a crucial assumption. If, e.g., a listening party could find out without doubt whether the other party transmitted or listened by receiving silence in the latter case, then uncorrupted communication could be arranged by simply using the listen/transmit state as an incorruptible one-bit communication channel. More subtle points arise when considering how to define the quantity of communication on which the adversaries budget of corruptions is based. The number of transmissions performed by the communicating parties, for example, seems like a good choice. This however would make the adversaries budget a variable (possibly probabilistic) quantity that, even worse, non-trivially depends on when and how this budgets is spent[1]. It would moreover allow parties to time-code, that is, encode a large number (even an encoding of all answers to all possible queries) in the time between two transmissions. While time-coding strategies do not seem to lead to efficient algorithms they would prevent strong lower bounds which show that even over an unbounded number of rounds no meaningful communication is possible (see, e.g., Theorem 3.2 which proves exactly this for an error rate of 2/7).

Our model avoids all these complications. For non-adaptive protocols that perfectly coordinate a designated sender and receiver in each round our model matches the standard setting. For the necessary case that adaptive parties fail to co-

[1]See [1] for an independent work exploring this direction.

ordinate our model prevents any signaling or time-sharing capabilities and in fact precludes any information exchange. This matches the intuition that in a conversation no advantage can be derived from both parties speaking or listening at the same time. It also guarantees that the product between the number of rounds and the bit-size of the communication alphabet is a clean and tight information theoretic upper bound on the total amount of information or entropy that can be exchanged (in either direction) between the two parties. This makes the number of rounds the perfect quantity to base the adversaries budget on. All this makes our model, in hindsight, the arguably cleanest and most natural extension of the standard setting to adaptive protocols. The strongest argument for our model however is the fact that it allows to prove both strong and natural positive *and* negative results, implying that our model does not overly restrict or empower the protocols or the adversary.

3. OVERVIEW

In this section we state our results and explain the high level ideas and insights behind them.

3.1 Adaptivity

A major contribution of this paper is to show that adaptive protocols can tolerate more than the 1/4 error rate of the non-adaptive setting:

THEOREM 3.1. *Suppose* Π *is an* n*-round protocol over a constant bit-size alphabet. For any* $\epsilon > 0$*, there is a deterministic computationally efficient protocol* Π' *that robustly simulates* Π *for an error rate of* $2/7 - \epsilon$ *using* $O(n)$ *rounds and an* $O(n)$*-bit size alphabet.*

The proof is given in Section 5. Section 6 of the full version [11] shows that the same error rate can be tolerated with a constant size alphabet but $O(n^2)$ rounds.

Furthermore, in Section 4.2, we show a matching impossibility result which even applies to the arguably simplest interactive protocol, namely, the *Exchange Protocol*. In the Exchange Protocol each party gets a one bit input which it has to communicate to the other party.

THEOREM 3.2. *There is no (deterministic or randomized) protocol that robustly simulates the Exchange Protocol for an error rate of* $2/7$ *with an* $o(1)$ *failure probability even when allowing computationally unbounded protocols that use arbitrarily large number of rounds and an unbounded alphabet.*

Why Adaptivity is Natural and Helpful: Next, we explain why it should not be surprising that adaptivity leads to a higher robustness. We also give some insights for why the 2/7 error rate is the natural noise tolerance threshold for adaptive protocols.

It is helpful to first understand why the 1/4 error rate was thought of as a natural barrier. The intuitive argument, presented in [7], for why one should not be able to cope with an error rate of 1/4 is as follows: During any N round interaction one of the parties, w.l.o.g. Alice, is the designated sender for at most half of the rounds. With an error rate of 1/4 the adversary can corrupt half of the symbols Alice sends out. This makes it impossible for Alice to (reliably) deliver even a single bit x because the adversary can always make the first half of her transmissions consistent with $x = 0$ and the second half with $x = 1$ without Bob being able to know which of the two is real and which is corrupted.

While this intuition is quite convincing at the first glance, it silently assumes that it is a priori clear which of the two parties transmits less often. This in turn essentially only holds for non-adaptive protocols for which the above argument can also be made into a formal negative result [7, Claim 10]. On the other hand, we show that assuming this a priori knowledge is not just a minor technical assumption but indeed a highly nontrivial restriction which is violated in many natural settings of interaction. For example, imagine a telephone conversation on a connection that is bad/noisy in one direction. One person, say Alice, clearly understands Bob while whatever Alice says contains so much noise that Bob has a hard time understanding it. In a non-adaptive conversation, Bob would continue to talk half of the time (even though he has nothing to say given the lack of understandable responses from Alice) while Alice continues to talk little enough that she can be completely out-noised. This is of course not how it would go in real life. There, Bob would listen more in trying to understand Alice and by doing this give Alice the opportunity to talk more. As soon as this happens, the adversary cannot completely out-noise Alice anymore and the conversation will be able to progress. Similar *dynamic rate adaptation* mechanisms that adapt the bitrate of a sender to channel conditions and the communication needs of other senders are common in many systems, a prominent example being IEEE 802.11 wireless networks.

Even if one is convinced that adaptive algorithms should be able to beat the 1/4 error rate, it is less clear at this point what the maximum tolerable error rate should be. In particular, 2/7 seems like a quite peculiar bound. Next, without going into details of the proofs, we want to give at least some insight why 2/7 is arguably the right and natural error rate for the adaptive setting.

We first give an intuitive argument why even adaptive protocols cannot deal with an error rate of 1/3. For this, the adversary runs the same strategy as above which concentrates all errors on one party only. In particular, given a $3N$ rounds conversation and a budget of N corruptions, the adversary picks one party, say Alice, and makes her first N transmissions sound like as if her input is $x = 1$. The next N transmissions are made to sound like Alice has input $x = 0$. During the first N responses, regardless of whether $x = 1$ (resulting in Alice talking herself) or $x = 0$ (resulting in the adversary imitating the same transmissions), the whole conversation will sound legitimate. This prevents any rate adaptation, in this case on Bob's side, to kick in before $2N$ rounds of back and forth have passed. Only then it becomes apparent to the receiver of the corruptions, in this case Bob, that the adversary is trying to fool him. Knowing that the adversary will only try to fool one party, Bob can then stop talking and listen to Alice for the rest of the conversation. Still, even if Bob listens exclusively from this point on, there are only N rounds left which is just enough for all of them to be corrupted. Having received N transmission from Alice claiming $x = 1$ and equally many claiming $x = 0$, Bob is again left puzzled. This essentially proves the impossibility of tolerating an error rate of 1/3. But even this 1/3 error rate is not achievable. To explain why even a lower fraction of errors, namely 2/7, leads to a negative result, we remark that the radical immediate back-off we just assumed for Bob is not possible. The reason is that if both parties are so sensitive and radical in their adjustment, the adversary can fool both parties simultaneously by simply corrupting a few transmissions of both parties after round $2N$. This would lead to both parties assuming that the transmissions

of the other side are being corrupted. The result would be both parties being silent simultaneously which wast es valuable communication rounds. Choosing the optimal tradeoff for how swift and strong protocols are able to adaptively react without falling into this trap naturally leads to an error rate between 1/4 and 1/3, and what rate in this range could be more natural than the mediant 2/7.

3.2 Other Settings

We also give results on other settings that have been suggested in the literature, in particular, list decoding and the shared randomness setting of [9]. We briefly describe these results next.

Franklin et al. [9] showed that if both parties share some random string not known to the adversary, then non-adaptive protocols can boost the tolerable error rate from 1/4 to 1/2. We show that also in this setting adaptivity helps to increase the tolerable error rate. In particular, in Section 7 of the full version[11], we prove that an error rate of $2/3 - \epsilon$ is achievable and best possible:

THEOREM 3.3. *In the shared randomness setting of [9], there exists a efficient robust coding scheme for an error rate of* $2/3 - \epsilon$ *while no such scheme exists for an error rate of* $2/3$. *That is, the equivalents of Theorem 3.1 and Theorem 3.2 hold for an error rate of* $2/3$. *The number of rounds of the robust coding scheme can furthermore be reduced to* $O(n)$ *if one allows exponential time computations.*

We also give the first results for list decodable coding schemes (see Section 2 for their definition). The notion of list decodability has been a somewhat recent but already widely successful addition to the study of error correcting codes. It is known that for error correcting codes such a relaxation leads to being able to efficiently [13] tolerate any constant error rate below 1, which is a factor of two higher than the $1/2 - \epsilon$ error rate achievable with unique decoding. It has been an open question whether list decoding can also lead to higher tolerable error rates in interactive coding schemes (see [4, Open Problem 9] and [7, Conclusion]). We show that this is indeed the case. In particular, for the non-adaptive setting the full factor of two improvement can also be realized in the interactive setting:

THEOREM 3.4. *Suppose* Π *is an* n*-round protocol over a constant-bit size alphabet. For any* $\epsilon > 0$ *there is an* $O(1)$*-list decodable non-adaptive deterministic computationally-efficient protocol* Π' *that robustly simulates* Π *for an error rate of* $1/2 - \epsilon$ *using* $O(n)$ *rounds and an* $O(n)$*-bit size alphabet.*

The proof of this theorem is presented in Section 5.2 and its extension to the constant alphabet size setting can be found in Section 6 of the full version [11]. We note that a subsequent but independent work by Braverman and Efremenko [6] also gave a list decodable coding scheme with the same tolerable error rate. Lastly, we show that this $1/2 - \epsilon$ error rate is best possible even for adaptive coding schemes. That is, no adaptive or non-adaptive coding scheme can achieve an error rate of 1/2. We prove these impossibility results formally in Appendix D of the full version[11].

Taken together, our results provide tight negative and matching positive results for any of the eight interactive coding settings given by the three Boolean attributes, {unique decoding / list decoding}, {adaptive / non-adaptive}, and {without shared randomness / with shared randomness} (at least when allowing a linear size alphabet or quadratic number of rounds in the simulation). Table 1 shows the maximum tolerable error rate for each of these settings:

	Unique Dec. (UD)	UD & shared rand.
Non-adaptive	1/4 ([7])	1/2 ([9])
Adaptive	2/7	2/3
	List Dec. (LD)	LD & shared rand.
Non-adaptive	1/2	1/2
Adaptive	1/2	2/3

Table 1: Unless marked with a citation all results in this table are new and presented in this paper. Matching positive and negative results for each setting show that the error rates are tight.

3.3 Invariability Hypothesis

In this section, we take a step back and propose a general way to understand the tolerable error rates specific to each setting and to design interactive coding schemes achieving them. We first formulate a strong working hypothesis which postulates that tolerable error rates are invariable regardless of what communication and computational resources are given to the protocol and to the adversary. We then use this hypothesis to determine the tight tolerable error rate for any setting by looking at the simplest setup. Finally, we show how the insights coming from these simpler setups can lead to designs for intuitive, natural, and easily analyzable interactive coding schemes for the general setup.

Invariability Hypothesis: In this section we formulate our invariability hypothesis.

Surveying the literature for what error rates could be tolerated by different interactive coding schemes, the maximum tolerable error rate appears to vary widely depending on the setting and more importantly, depending on what kind of efficiency one strives for. For example, even for the standard setting—that is, for non-adaptive unique decoding coding schemes using a large alphabet—the following error rates apply: for unbounded (or exponential time) computations, Schulman [12] tolerates a 1/240 error rate; Braverman and Rao [7] improved this to 1/4; for sub-exponential time computations, Braverman [5] gave a scheme working for any error rate below 1/40; for randomized polynomial time algorithms, Brakerski and Kalai [2] got an error rate of 1/16; for randomized linear time computations, Brakerski and Naor [3] obtained an unspecified constant error rate smaller than 1/32; lastly, assuming polynomially bounded protocols and adversaries and using a super-linear number of rounds, Chung et al. [8] gave coding schemes tolerating an error rate of 1/6 (with additional desirable properties).

We believe that this variety is an artifact of the current state of knowledge rather then being the essential truth. In fact, it appears that any setting comes with one tolerable error rate which exhibits a strong threshold behavior: For any setting, there seems to be one error rate for which communication is impossible regardless of the resources available, while for error rates only minimally below it simple and efficient coding schemes exist. In short, the tolerable error rate for a setting seems robust and completely independent communication resource or computational complexity restrictions made to the protocols or to the adversary.

Taking this observation as a serious working hypothesis was a driving force in obtaining, understanding, and structuring the results obtained in this work. As we will show, it helped to identify the simplest setup for determining the tolerable error rate of a setting, served as a good pointer to open questions, and helped in the design of new, simple, and natural coding schemes. We believe that these insights and schemes will be helpful in future research to obtain the optimal, and efficient coding schemes postulated to exist. In fact, we already have a number of subsequent works confirming this (e.g., the results of [5, 2, 8] mentioned above can all be extended to have the optimal 1/4 error rate). All in all, we believe that identifying and formulating this hypothesis is an important conceptual contribution of this work:

HYPOTHESIS 3.5 (INVARIABILITY HYPOTHESIS). *Given any of the eight settings for interactive communication (discussed above) the maximum tolerable error rates is invariable regardless:*

1. *whether the protocol to be simulated is an arbitrary n-round protocol or the much simpler (n-bit) exchange protocol, and*
2. *whether only $O(1)$-bit size alphabets are allowed or alphabets of arbitrary size, and*
3. *whether the simulation has to run in $O(n)$ rounds or is allowed to use an arbitrary number of rounds, and*
4. *whether the parties are restricted to polynomial time computations or are computationally unbounded, and*
5. *whether the coding schemes have to be deterministic or are allowed to use private randomness (even when only requiring an $o(1)$ failure probability), and*
6. *whether the adversary is computationally unbounded or is polynomially bounded in its computations (allowing simulation access to the coding scheme if the coding scheme is not computationally bounded)*

We note that our negative results are already as strong as stipulated by the hypothesis, for all eight settings. The next corollary furthermore summarizes how far these negative results combined with the positive results presented in this work (see Table 1) already imply and prove two weaker versions of the hypothesis:

COROLLARY 3.6. *The Invariability Hypothesis holds if one weakens point 3. to "3'. whether only $O(n)$-bit size alphabets are allowed or alphabets of arbitrary size". The Invariability Hypothesis also holds if one weakens point 4. to "4'. whether the simulation has to run in $O(n^2)$ rounds or is allowed to use an arbitrary number of rounds".*

We also refer the reader to [10, 6] for further (subsequent) results supporting the hypothesis, such as, a proof that the hypothesis holds if either point *5.* or point *6.* are omitted.

Understanding Tolerable Error Rates: Next, we explain how we use the invariability hypothesis to find and understand optimal tolerable error rates.

Suppose that one assumes, as a working hypothesis, the invariability of tolerable error rates to hold regardless of the computational setup and even the structure of the protocol to be simulated. Under this premise, the easiest way to approach determining the best error rate is in trying to design robust simulations for the simplest possible two-way protocol, the *Exchange Protocol*. This protocol simply gives each party n bits as an input and has both parties learn the other party's input bits by exchanging them (see also Section 4). Studying this setup is considerably simpler. For instance, for non-adaptive protocols, it is easy to see that both parties sending their input in an error correcting code (or for $n = 1$ simply repeating their input bit) leads to the optimal robust protocol which tolerates any error rate below 1/4 but not more. The same coding scheme with applying any ECC list decoder in the end also gives the tight 1/2 bound for list decoding. For adaptive protocols (both with and without shared randomness), finding the optimal robust 1-bit exchange protocol was less trivial but clearly still better than trying to design highly efficient coding schemes for general protocols right away. Interestingly, looking at simpler setup actually crystallized out well what can and cannot be done with adaptivity, and why. These insights, on the one hand, lead to the strong lower bounds for the exchange problem but, on the other hand, were also translated in a crucial manner to the same tradeoffs for robustly simulating general n-round protocols.

Natural Interactive Coding Schemes: The invariability working hypothesis was also helpful in finding and formalizing simple and natural designs for obtaining robust coding schemes for general protocols.

Before describing these natural coding schemes we first discuss the element of "surprise/magic" in prior works on interactive coding. The existence of an interactive coding scheme that tolerates a constant error rate is a fairly surprising outcome of the prior works, and remains so even in hindsight. One reason for this is that the simplest way of adding redundancy to a conversation, namely encoding each message via an error correcting code, fails dramatically because the adversary can use its budget non-uniformly and corrupt the first message(s) completely. This derails the interaction completely and makes all further exchanges useless even if no corruptions happens from there on. While prior works, such as [12] or [7], manage to overcome this major challenge, their solution remains a technically intriguing works, both in terms of the ingredients they involve (tree codes, whose existence is again a surprising event) and the recipe for converting the ingredients into a solution to the interactive coding problem. As a result it would appear that the challenge of dealing with errors in interactive communication is an inherently complex task.

In contrast to this, we aim to give an intuitive and natural strategy which lends itself nicely to a simple explanation for the possibility of robust interactive coding schemes and even for why their tolerable error rates are as they are. This strategy simply asserts that if there is no hope to fully trust messages exchanged before, one should find ways to put any response into the assumed common context by (efficiently) referring back to what one said before. Putting this idea into a high-level semi-algorithmic description gives the following outline for a robust conversation:

At first glance the algorithm may appear vague. In particular notions like "making sense", and "most relevant response", seem ambiguous and subject to interpretation. It turns out that this is not the case. In fact, formalizing this outline into a concrete coding scheme turns out to be straightforward. This is true specially if one accepts the invariability working hypothesis and allows oneself to not be overly concerned with immediately getting a highly efficient implementation. This permits to use the simplest (inefficiently) summary, namely referring back word by word to everything said before. This straightforward formalization leads to Algorithm 2. Indeed, a reader that compares the two algorithms side-by-side will find that Algorithm 2 is essentially a line-by-line formalization of Algorithm 1.

Algorithm 1 Natural Strategy for a Robust Conversation (Alice's Side)

```
1: Assume nothing about the context of the conversation
2: loop
3:     Listen
4:     E'_B ← What you heard Bob say last (or so far)
5:     E_A ← What you said last (or so far)
6:     if E_A and E'_B makes sense together then
7:         Determine the most relevant response r
8:         Send the response r but also include an (efficient) summary of what you said so far (E_A)
9:     else
10:        Repeat what you said last (E_A)
11: Assume / Output the conversation outcome(s) that seem most likely
```

In addition to being arguably natural, Algorithm 2 is also easy to analyze. Simple counting arguments show that the conversation outcome output by most parties is correct if the adversary interferes at most a $1/4 - \epsilon$ fraction of the time, proving the tight tolerable error rate for the robust (while somewhat still inefficient) simulation of general n-round protocols. Maybe even more surprisingly, the exact same simple counting argument also directly shows our list decoding result, namely, that even with an error rate of $1/2 - \epsilon$ the correct conversation outcome is among the $1/\epsilon$ most likely choices for both parties. Lastly, it is easy to enhance both Algorithm 1 and similarly Algorithm 2 to be adaptive. For this one simply adds the following three, almost obvious, *rules for a succesful adaptive conversation*:

Rules 3.7. *Be fair and take turns talking and listening, unless:*

1. *you are sure that your conversation partner already understood you fully and correctly, in which case you should stop talking and instead listen more to also understand him; or reversely*
2. *you are sure that you already understood your conversation partner fully and correctly, in which case you should stop listening and instead talk more to also inform him.*

Our algorithm Algorithm 3 exactly adds the formal equivalent of these rules to Algorithm 3. A relatively simple proof that draws on the insights obtained from analyzing the optimal robust exchange protocol then shows that this simple and natural coding scheme indeed achieves the optimal $2/7 - \epsilon$ tolerable error rate for adaptive unique-decoding. This means that Algorithm 3 is one intuitive and natural algorithm that simultaneously achieves the $1/4$ error rate (if the adaptivity rules are ignored), the $2/7 - \epsilon$ error rate for adaptive protocols and the $1/2 - \epsilon$ error rate with optimal list size when list decoding is allowed. Of course, so far, this result comes with the drawback of using a large ($O(n)$-bits) alphabet. Nonetheless, this result together with the invariability hypothesis hold open the promise of such a "perfect" algorithm that works even without the drastic communication overhead. Subsequent works have achieved this for algorithms using randomization or doing exponential computations [10, 6].

4. RESULTS FOR THE EXCHANGE PROB.

Here we study the *Exchange Problem*, which can be viewed as the simplest instance of a two-way (i.e., interactive) communication problem. In the Exchange Problem, each party is given a bit-string of n bits, that is, $i_A, i_B \in \{0,1\}^n$, and each party wants to know the bit-string of the other party.

Recall that the $1/4$ impossibility bound on tolerable error-rate for non-adaptive interactive protocols presented by Braverman and Rao [7] is this simple setting. In Section 4.1, we show that adding rate adaptivity to the exchange algorithms helps one break this $1/4$ impossiblity bound and tolerate an error-rate of $2/7 - \epsilon$, and in fact, this is done with a minimal amount of adaptivity-based decisions regarding whether a party should transmit or listen in each round. We show in Section 4.2 that the error-rate of $2/7$ is not tolerable even for the exchange problem, even if one is given infinite number of rounds, alphabet size, and computational power. Furthermore, the intuition used to achieve the $2/7 - \epsilon$ possibility result also extends to the more general *simulation* problem, discussed in Section 5.

4.1 Exchange with an Error-Rate of $2/7 - \epsilon$

Note that a simple solution based on error correcting codes suffices for solving exchange problem under error-rate $\frac{1}{4} - \epsilon$: parties use a code with relative distance $1 - \epsilon$. In the first half of the time, Alice sends its encoded message and in the second half of the time, Bob sends its encoded message. At the end, each party decodes simply by picking the codeword closest to the received string. As discussed before, the error rate $\frac{1}{4} - \epsilon$ of this approach is the best possible if no rate adaptation is used. In the following, we explain that a simple rate adaptation technique boosts the tolerable error-rate to $\frac{2}{7} - \epsilon$, which we later prove to be optimal.

THEOREM 4.1. *In the private randomness model with rate adaptation, there is an algorithm for the n-bit Exchange Problem that tolerates adversarial error rate of $2/7 - \epsilon \approx 0.2857 - \epsilon$ for any $\epsilon > 0$.*

PROOF. The algorithm runs in $N = 7n/\epsilon$ rounds, which means that the budget of adversary is $(2/7-\epsilon)7n/\epsilon = 2n/\epsilon - 7$. Throughout the algorithm, we use an error-correction code $\mathcal{C} : \{0,1\}^n \to \{1,\ldots,q\}^{\frac{n}{\epsilon}}$ that has distance $\frac{n}{\epsilon} - 1$. Also, for simplicity, we use $\mathcal{C}^\kappa$ to denote the code formed by concatenating κ copies of $\mathcal{C}$.

The first $6N/7$ rounds of the algorithm do not use any rate adaptation: Simply, Alice sends $\mathcal{C}^3(i_A)$ in the first $3N/7$ rounds and Bob sends $\mathcal{C}^3(i_B)$ in the second $3N/7$ rounds. At the end of this part, each party "estimates" the errors invested on the transmissions of the other party by simply reading the hamming distance of the received string to the closest codeword of code $\mathcal{C}^3$. If this estimate is less than $N/7 = n/\epsilon$, the party—say Alice—can safely assume that the closest codeword is the correct codeword. This is because the adversary's total budget is $2n/\epsilon - 7$ and the distance between two codewords of $\mathcal{C}^3(i_B)$ is at least $3n/\epsilon - 3$. In this case, in the remaining $N/7$ rounds of the algorithm, Alice will be sending $\mathcal{C}(i_A)$ and never listening. On the other hand, if Alice reads an estimated error greater than $N/7 = n/\epsilon$, then in the remaining $N/7$ rounds, she will be always listening. The algorithm for Bob is similar.

Because of the limit on the budget of the adversary, at most only one of the parties will be listening in the last $N/7$ rounds. Suppose that there is exactly one listening party and it is Alice. Then, throughout the whole algorithm, Alice has listened a total of $4N/7 = 4n/\epsilon$ rounds where Bob has been sending $\mathcal{C}^4(i_B)$. Since the adversaries budget is less than $2n/\epsilon - 7$, and because $\mathcal{C}^4$ has distance $\frac{4n}{\epsilon} - 4$, Alice can also decode correctly by just picking the codeword of $\mathcal{C}^4$ with the smallest hamming distance to the received string. □

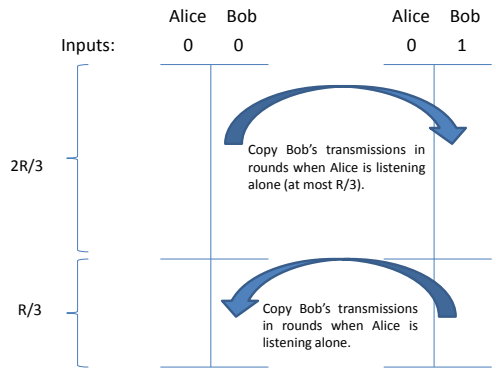

Figure 1: The adversary's strategy for the 1/3-impossibility proof

4.2 Impossibility of Tolerating Error-Rate 2/7

In this section, we turn our attention to impossibility results and particularly prove that the error-rate 2/7 is not tolerable. See the formal statement in Theorem 3.2.

Braverman and Rao [7] showed that it is not possible to tolerate error-rate of 1/4 with non-adaptive algorithms. For completeness, a (slightly more formal) proof in the style of distributed indistinguishably arguments is presented in the full version[11], which also covers random algorithms.

We first explain a simple (but informal) argument which shows that even with adaptivity, error-rate 1/3 is not tolerable. A formal version of this proof appears in the full version[11]. The informal version explained next serves as a warm up for the more complex argument used for proving the 2/7 impossibility, presented formally in Theorem 4.3.

LEMMA 4.2. *There is no (deterministic or randomized) adaptive protocol that robustly simulates the Exchange Protocol for an error rate of* 1/3 *with failure probability* $o(1)$*, even when allowing computationally unbounded protocols with arbitrarily large number of rounds and unbounded alphabet.*

PROOF. To simplify the discussion, here we only explain the argument for deterministic algorithms and we also ignore the rounds in which both parties listen. Note that by definition of the model, in those all-listening rounds, the adversary can deliver arbitrary messages to each of the parties at no cost. A complete proof appears in the full version[11].

For the sake of contradiction, suppose that there is an algorithm that solves the exchange problem under adversarial error-rate 1/3, in N rounds. We work simply with 1-bit inputs. Let $S_{X,Y}$ denote the setting where Alice receives input X and Bob gets input Y. The idea is to make either settings $S_{0,0}$ and $S_{0,1}$ look indistinguishable to Alice or settings $S_{0,0}$ and $S_{1,0}$ look indistinguishable to Bob.

Consider setting $S_{0,0}$ and suppose that for the first $2N/3$ rounds, the adversary does not interfere. Without loss of generality, we can assume that in this setting, Alice listens (alone) in less than $N/3$ of these $2N/3$ rounds. We next explain adversary's strategy for the case that this assumption holds. An pictorial illustration is presented in Figure 1.

First, we explain the adversary's strategy for setting $S_{0,1}$: Adversary creates a *dummy personality* $\widetilde{Bob}_0$ and simulates it with Alice in setting $S_{0,0}$ where adversary does not interfere. In the first $2N/3$ rounds of setting $S_{0,1}$, whenever Alice listens (alone), the adversary delivers transmission of $\widetilde{Bob}_0$ to Alice. As a shorthand for this, we say *Alice is connected to* $\widetilde{Bob}_0$. Since Alice listens less than $N/3$ of the time, the adversary will have enough budget to completely fake Bob as $\widetilde{Bob}_0$ (from Alice's viewpoint). Thus, the two settings look identical to Alice for the first $2N/3$ rounds. During the last $N/3$ rounds of the execution in setting $S_{0,1}$, the adversary lets Alice and the real Bob talk without no interference.

Now, we explain the adversary's strategy for setting $S_{0,0}$: The adversary generates another dummy personality $\widetilde{Bob}_1$ by simulating Bob in setting $S_{0,1}$ where alone-listening rounds of Alice in the first $2N/3$ rounds are connected to $\widetilde{Bob}_0$. In setting $S_{0,0}$, the adversary lets Alice and Bob talk freely during the first $2N/3$ rounds but for the last $N/3$ rounds, whenever Alice listens, the adversary connects her to $\widetilde{Bob}_1$.

To conclude, in each of the settings $S_{0,1}$ and $S_{1,0}$ at most $N/3$ rounds get corrupted by the adversary. Furthermore, the two settings look identical to Alice. This means she cannot know Bob's input, which completes the proof. □

THEOREM 4.3. *[A rephrasing of Theorem 3.2] There is no (deterministic or randomized) adaptive protocol that robustly simulates the Exchange Protocol for an error rate of* 2/7 *with an* $o(1)$ *failure probability even when allowing computationally unbounded protocols that use an arbitrarily large number of rounds and an unbounded alphabet.*

PROOF. Suppose that there is an algorithm that solves the exchange problem under adversarial error-rate 1/3, in N rounds. We study this algorithm simply with 1-bit inputs. Let $S_{X,Y}$ denote the setting where Alice receives input X and Bob gets input Y. We focus on settings $S_{0,0}$, $S_{0,1}$, and $S_{1,0}$. If a party has an input 1, it knows in which of these three settings we are. The idea is to present an adversarial strategy that changes the receptions of the party, or parties, that have a 0 input so as to make that party, or parties, unable to distinguish (between two of) the settings.

For simplicity, we first assume that the algorithm is deterministic and we also ignore the rounds where both parties listen. Note that by the definition of the model for adaptive algorithms (see Section 2), for these rounds, the adversary can deliver arbitrary messages to the parties at no cost.

For this lower bound, we need to define the party that becomes the base of indistinguishability (whom we confuse by errors) in a more dynamic way, compared to that in Theorem 4.2 or in [7, Claim 10]. For this purpose, we first study the parties that have input 0 under a particular pattern of received messages (regardless of the setting in which they are), without considering whether the adversary has enough budget to create this pattern or not. Later, we argue that the adversary indeed has enough budget to create this pattern for at least one party and make that party confused.

To determine what should be delivered to each party with input 0, the adversary cultivates *dummy personalities* $\widetilde{Alice}_0$, $\widetilde{Alice}_1$, $\widetilde{Bob}_0$, $\widetilde{Bob}_1$, by simulating Alice or Bob respectively in settings $S_{0,0}$, $S_{1,0}$, $S_{0,0}$, and $S_{0,1}$, where each of these settings is modified by adversarial interferences (to be specified). Later, when we say that in a given round, e.g. "the adversary *connects* dummy personality $\widetilde{Bob}_1$ to Alice", we mean that the adversary delivers the transmission of $\widetilde{Bob}_1$ in that round to Alice[2]. For each setting, the adversary uses one method of interferences, and thus, when we refer to a setting, we always mean the setting with the related adversarial interferences included.

[2]This is assuming $\widetilde{Bob}_1$ transmits in that round, we later discuss the case where both Alice and $\widetilde{Bob}_1$ listen later.

We now explain the said pattern of received messages. Suppose that Alice has input 0 and consider her in settings $S_{0,0}$ and $S_{0,1}$, as a priori these two settings are identical to Alice. Using connections to *dummy personalities*, the adversary creates the following pattern: In the first $2N/7$ rounds in which Alice listens alone, her receptions will be made to imply that Bob also has a 0. This happens with no adversarial interference in setting $S_{0,0}$, but it is enforced to happen in setting $S_{0,1}$ by the adversary via connecting to Alice the dummy personality $\widetilde{Bob}_0$ cultivated in setting $S_{0,0}$. Thus, at the end of those $2N/7$ listening-alone rounds of Alice, the two settings are indistinguishable to Alice. In the rest of the rounds where Alice listens alone, the receptions will be made to look as if Bob has a 1. That is, the adversary leaves those rounds of setting $S_{0,1}$ intact, but in rounds of setting $S_{0,0}$ in which Alice listens alone, the adversary connects to Alice the dummy personality $\widetilde{Bob}_1$ cultivated in setting $S_{0,1}$ (with the adversarial behavior described above). The adversary creates a similar pattern of receptions for Bob in settings $S_{0,0}$ and $S_{1,0}$. That is, the first $2N/7$ of his alone receptions are made to imply that Alice has a 0 but the later alone-receptions imply that Alice has a 1. The described reception pattens make Alice unable to distinguish $S_{0,0}$ from $S_{0,1}$ and also they make Bob unable to distinguish $S_{0,0}$ from $S_{1,0}$. However, the above discussions ignore the adversary's budget. We now argue that the adversary indeed has enough budget to create this reception pattern to confuse one or both of the parties.

Let x_A be the total number of rounds where Alice listens, when she has input 0 and her receptions follow the above pattern. Similarly, define x_B for Bob. If $x_A \leq 4N/7$, then the adversary indeed has enough budget to make the receptions of Alice in settings $S_{0,0}$ and $S_{0,1}$ follow the discussed behavior, where the first $2N/7$ alone-receptions of Alice are altered in $S_{0,1}$ and the remaining alone-receptions are altered in $S_{0,0}$. Thus, if $x_A \leq 4N/7$, the adversary has a legitimate strategy to make Alice confused between $S_{0,0}$ and $S_{0,1}$. A similar statement is true about Bob: if $x_B \leq 4N/7$, the adversary has a legitimate strategy to make Bob confused between $S_{0,0}$ and $S_{0,1}$.

Suppose that $x_A > 4N/7$ and $x_B > 4N/7$. Then, the number of alone-receptions of Alice is at most $x_A - (x_A + x_B - N) = N - x_B \leq 3N/7$ and similarly, the number of alone-receptions of Bob is at most $x_B - (x_A + x_B - N) = N - x_A \leq 3N/7$. This is because $x_A + x_B - N$ is a lower bound on the overlap of the rounds that the two parties listen. In this case, the adversary has enough budget to simultaneously confuse both Alice and Bob of setting $S_{0,0}$; Alice will be confused between $S_{0,0}$ and $S_{0,1}$ and Bob will be confused between $S_{0,0}$ and $S_{1,0}$. For that, in setting $S_{0,0}$, the adversary leaves the first $2N/7$ alone-receptions of each party intact but alters the remaining at most $N/7$ alone-receptions of each party, for a total of at most $2N/7$ errors. On the other hand, in setting $S_{0,1}$, $2N/7$ errors are used on the first $2N/7$ alone-receptions of Alice and in setting $S_{1,0}$, $2N/7$ errors are used on the first $2N/7$ alone-receptions of Bob. Note that these errors make the receptions of each party that has input 0 follow the pattern explained above.

We now go back to the issue of the rounds where both parties listen. The rounds of $S_{0,0}$ in which both parties listen are treated as follows: The adversary delivers the transmission of $\widetilde{Bob}_1$ (cultivated in setting $S_{0,1}$) to Alice and delivers the transmission of $\widetilde{Alice}_1$ (cultivated in setting $S_{1,0}$) to Bob. Recall that the adversary does not pay for these interferences. Furthermore, note that these connections make sure that these all-listening rounds do not help Alice to distinguish $S_{0,0}$ from $S_{0,1}$ and also they do not help Bob to distinguish $S_{0,0}$ from $S_{1,0}$.

Finally, we turn to covering the randomized algorithms. Note that for this case we only show that the failure probability of the algorithm is not $o(1)$ as just by guessing randomly, the two parties can have success probability of 1/4.

First suppose that $Pr[x_A \leq 4N/7] \geq 1/3$. Note that the adversary can easily compute this probability, or even simpler just get a $(1 + o(1))$-factor estimation of it. If $Pr[x_A \leq 4N/7] \geq 1/3$, then the adversary will hedge his bets on that $x_A \leq 4N/7$, and thus, it will try to confuse Alice. In particular, he gives Alice an input 0 and tosses a coin and gives Bob a 0 or a 1, accordingly. Regarding whether *Bob* gets input 0 or 1, the adversary also uses the dummy personalities $\widetilde{Bob}_0$ and $\widetilde{Bob}_1$, respectively. With probability 1/3, we will have that in fact $x_A \leq 4N/7$, and in this case the adversary by determining whether Alice hears from the real Bob or the dummy Bob, the adversary makes Alice receive the messages with the pattern described above. This means Alice would not know whether Bob has a 0 or a 1. Hence, the algorithm fails with probability at least 1/6 (Alice can still guess in this case which is correct with probability 1/2). Similarly, if $Pr[x_B \leq 4N/7] \geq 1/3$, then adversary will make Bob confused between $S_{0,0}$ and $S_{1,0}$. On the other hand, if $Pr[x_A \leq 4N/7] < 1/3$ and $Pr[x_B \leq 4N/7] < 1/3$, then just using a union bound we know that $\Pr[x_A > 4N/7 \& x_B > 4N/7] \geq 1/3$. In this case, the adversary gambles on the assumption that it will actually happen that $x_A > 4N/7$ and $x_B > 4N/7$. This assumption happens with probability at least 1/3, and in that case, the adversary makes Alice confused between $S_{0,0}$ and $S_{0,1}$ and Bob confused between $S_{0,0}$ and $S_{1,0}$, simultaneously, using the approach described above. Hence, in conclusion, in any of the cases regarding random variables x_A and x_B, the adversary can make the algorithm fail with probability at least 1/6. □

5. NATURAL INTERACTIVE CODING

We start by presenting a canonical format for interactive communication and then present our natural non-adaptive and adaptive coding schemes.

5.1 Interactive Protocols in Canonical Form

We consider the following canonical form of an n-round two party protocol over alphabet Σ: We call the two parties Alice and Bob. To define the protocol between them, we take a rooted complete $|\Sigma|$-ary tree of depths n. Each non-leaf node has $|\Sigma|$ edges to its children, each labeled with a distinct symbol from Σ. For each node, one of the edges towards children is *preferred*, and these *preferred* edges determine a unique leaf or equivalently a unique path from the root to a leaf. We say that the set $\mathcal{X}$ of the preferred edges at odd levels of the tree is owned by Alice and the set $\mathcal{Y}$ of the preferred edges at even levels is owned by Bob. This means that at the beginning of the protocol, Alice gets to know the preferred edges on the odd levels and Bob gets to know the preferred edges on the even levels. The knowledge about these preferred edges is considered as inputs $\mathcal{X}$ and $\mathcal{Y}$ given respectively to Alice and Bob. The output of the protocol is the unique path from the root to a leaf following only preferred edges. We call this path the *common path* and the edges and nodes on this path the *common edges* and the *common nodes*. The goal is to determine the com-

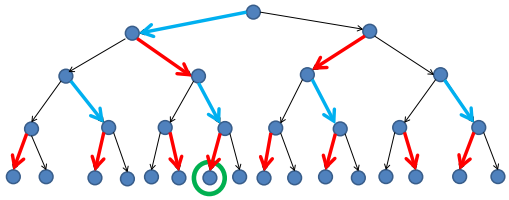

Figure 2: A Canonical Binary Interactive Protocol. Alice and Bob's preferred edges are indicated with blue and red arrows and the common leaf is indicated by a green circle.

mon path and we say a protocol succeeds if and only if both parties learn the common path.

This is easy if the channel is noiseless: Alice and Bob simply performing n rounds of communication, by moving down the tree together following the path of preferred edges. They take turns and exchange one symbol of Σ per round, where each symbol indicates the next common node. We call this exchange the execution of protocol P.

5.2 Natural Non-Adaptive Coding Schemes

In this section, we present a non-adaptive coding scheme which is a straightforward formalization of the natural approach presented in Section 3. This coding scheme tolerates the optimal error rate of $1/4 - \epsilon$ for unique decoding and simultaneously the optimal error rate of $1/2 - \epsilon$ when list decoding. The coding scheme is simple, intuitive, and computationally efficient, but uses a large $O(\frac{n}{\epsilon})$-bit size alphabet. We note that one can view this algorithm as a simplified version of the Braverman-Rao scheme [7] in which the use of tree codes is circumvented by utilizing a larger alphabet.

The algorithm, for which a pseudo code is presented in Algorithm 2, works as follows: In the course of the algorithm, Alice and Bob respectively maintain sets E_A and E_B which are a subset of their own preferred tree edges that are considered to be *important*. We call these *important edge-sets* or sometimes simple *edge-sets*. Initially these edge-sets are empty and in each iteration, Alice and Bob add one edge to their sets. In each iteration, when a party gets a turn to transmit, it sends its edge-set to the other party. The other party receives either the correct edge-set or a corrupted symbol which represents an edge-set made up by the adversary. In either case, the party combines the received edge-set with its own important edge-set and follows the common path in this set. Then, if this common path can be extended by the party's own set of *preferred edges* by a new edge e, the party adds this edge e to its edge-set, and sends this new edge-set in the next round. If, on the other hand, the common path already ends at a leaf, then the party registers this as a vote for this leaf and simply re-sends its old edge-set. In the end, both parties simply output the the leaf (respectively the $O(1/\epsilon)$ leaves) with the most votes for unique decoding (resp., for list decoding).

Analysis. We now prove that Algorithm 2 indeed achieves the optimal tolerable error rates for non-adaptive unique decoding and list decoding.

THEOREM 5.1. *For any $\epsilon > 0$, Algorithm 2 is a deterministic polynomial time non-adaptive simulator with alphabet size of $O(\frac{n}{\epsilon})$-bits and round complexity of $\frac{2n}{\epsilon}$ that tolerates an error-rate of $1/4 - \epsilon$ for unique decoding, and an error-rate of $1/2 - \epsilon$ for list decoding with list size $\frac{1}{\epsilon}$.*

Algorithm 2 Non-Adaptive Coding Scheme for Alice

$\mathcal{X} \leftarrow$ the set of Alice's preferred edges;
$E_A \leftarrow \emptyset$; ▷ E_A is Alice's set of *important* edges. We preserve that always $E_A \subseteq \mathcal{X}$
$N \leftarrow \frac{2n}{\epsilon}$;
for $i = 1$ to N **do**
  Receive edge-set E'_B; ▷ E'_B is the received version of Bob's important edge-set E_B
  $E \leftarrow E'_B \cup E_A$
  if E is a valid edgeset **then**
    $r \leftarrow \emptyset$
    follow the common path in E
    if the common path ends at a leaf **then**
      Add one vote to this leaf
    else
      $r \leftarrow \{e\}$ where e is the next edge in $\mathcal{X}$ continuing the common path in E (if any)
    $E_A \leftarrow E_A \cup r$
    Send E_A
  else
    Send E_A
Output the leaf with the most votes for unique decoding
Output the $O(1/\epsilon)$ leaves with the most votes for list decoding

PROOF. Clearly, both E_A and E_B grow by at most one edge per round. Furthermore, the edges always attach to an already present edge and therefore, each of these edge-sets always forms a subtree with size at most Nstarting at the root of the tree of the canonical form, which has depth n. One can easily see that each such subtree can be encoded using $O(N)$ bits, e.g., by encoding each edge of the breadth first search traversal of the subtree using alphabet of size 3 (indicating "left", "right" or "up"). Hence, parties can encode their edge-sets using $O(\frac{n}{\epsilon})$-bits symbols, which shows that the alphabet size is indeed as specified.

We now prove the correctness, starting with that of unique decoding. Note that any two consecutive rounds in which Bob and Alice's transmissions are not corrupted by adversary, one of the following two good things happens: Either the path in $E_A \cup E_B$ gets extended by at least one edge, or both Alice and Bob receive a vote for the correct leaf.

Now suppose that the simulation runs in $N = 2n/\epsilon$ rounds which can be grouped into n/ϵ round pairs. Given the error rate of $1/4-\epsilon$, at most a $1/2-2\epsilon$ fraction of these round pairs can be corrupted, which leaves $N/2(1/2 + 2\epsilon)$ uncorrupted round pairs. At most n of these round pairs grow the path while the remaining $N/2(1/2 + 2\epsilon) - n$ rounds vote for the correct leaf. This results in at least $N(1/2 + 2\epsilon) - \lceil n/2 \rceil = \frac{n}{2\epsilon} + 2n - n > N/4$ out of $N/2$ votes being correct.

For the list decoding, with error rate $1/2 - \epsilon$, we get that at most $1 - 2\epsilon$ fraction of round-pairs are corrupted, and thus at least $N\epsilon = 2n$ uncorrupted pairs exist. Hence, the correct leaf gets a vote of at least $2n - n$. Since the total number of votes that one party gives to its leaves is $N/2 = \frac{n}{\epsilon}$, the correct leaf has at least a ϵ fraction of all the votes. Therefore, if we output the $1/\epsilon$ leaves with the most votes, the list will include the correct leaf. □

5.3 Natural Adaptive Coding Scheme

In this section we show a simple way to introduce adaptation into the natural coding scheme presented in Algorithm 2. In particular, we use the rules specified in Rules 3.7 and show that this leads to a coding scheme tolerating an error rate of $2/7 - \epsilon$, the optimal error rate for this setting.

We first note that if in Algorithm 2 one party ends up with a leaf that has more than $(2/7 - \epsilon)N$ votes, it knows

Algorithm 3 Adaptive Coding Scheme for Alice

$\mathcal{X} \leftarrow$ the set of Alice's preferred edges;
$E_A \leftarrow \emptyset$;
$N \leftarrow \Theta(\frac{n}{\epsilon})$;
for $i = 1$ to $\frac{6}{7}N$ **do**
    Receive edge-set E'_B;
    $E \leftarrow E'_B \cup E_A$
    if E is a valid edgeset **then**
        $r \leftarrow \emptyset$
        follow the common path in E
        if the common path ends at a leaf **then**
            Add one vote to this leaf
        else
            $r \leftarrow \{e\}$ where e is the next edge in $\mathcal{X}$ continuing the common path in E (if any)
        $E_A \leftarrow E_A \cup r$
        Send E_A
    else
        Send E_A
Let s be number of votes of the leaf with the most votes
Let t be the total number of votes
if $s \geq t - \frac{N}{7}$ **then**
    for $i = 1$ to $\frac{N}{7}$ **do**
        Send E_A
else
    for $i = 1$ to $\frac{N}{7}$ **do**
        Receive edge-set E'_B; $E = E'_B \cup E_A$
        if E is a valid edge-set **then**
            follow the common path in E
            if the common path ends at a leaf **then**
                Add one vote to this leaf
Output the leaf with the most votes

that this leaf is the unique correct leaf, since adversary has only a budget of $(2/7 - \epsilon)N$. This party would then follow the second rule. Generalizing this idea, we use the rule that, if the party has a leaf v such that only at most $\frac{N}{7}$ votes are on leaves other than v, then the party can safely assume that this is the correct leaf. In our proof we show that this assumption is indeed safe and furthermore, at least one party can safely decode at the end of the first 6/7 fraction of the simulation. Since both parties know this in advance, if a party cannot safely decode after 6/7 fraction of the time, it knows that the other party has safely decoded—which corresponds to the condition in the first rule—and thus, this party only listens for the last 1/7 fraction of the protocol. The pseudocode is presented in Algorithm 3.

THEOREM 5.2. *Algorithm 3 is a deterministic adaptive coding scheme with alphabet size of $O(\frac{n}{\epsilon})$-bits, round complexity of $O(\frac{n}{\epsilon})$, and polynomial computational complexity that tolerates an error-rate of $2/7 - \epsilon$ for unique decoding.*

PROOF. First, we show that if at the end of $\frac{6N}{7}$ rounds, one party has t votes, $s \geq t - \frac{N}{7}$ of which are dedicated to one leaf v, then this party can safely assume that this leaf v is the correct leaf. We prove this by contradiction. If the party has s votes, then there are at least $\frac{3N}{7} - t$ that either stopped the growth of the path or turned an edge-set into a nonvalid edge-set. Furthermore, if v is not the correct leaf, then the votes v are created by errors of adversary which means that adversary has invested s errors on turning the edge-sets sent by the other party into other valid-looking edge-sets. Hence, in total, adversary has spent at least $\frac{3N}{7} - t + s \geq \frac{3N}{7} - t + t - \frac{N}{7} \geq \frac{2N}{7}$ errors which is a contradiction.

Now that we know that the rule for safely decoding at the end of $\frac{6N}{7}$ rounds is indeed safe, we show that at least one party will safely decode at that point of time. Suppose that no party can decode safely. Also assume that Alice has t_A votes, r_A of which are votes on the good leaf. That means at least adversary has turned at least $t_A - r_A$ edge-sets sent by Bob into other valid-looking edge-sets. Similarly, $t_B - r_B$ errors are introduced by the adversary on edge-sets sent by Alice. If neither Alice nor Bob can decode safely, we know that $t_A - r_A \geq \frac{N}{7}$ and $t_B - r_B \geq \frac{N}{7}$, which means that in total, adversary has introduced at least $\frac{2N}{7}$ errors. Since this is not possible give adversary's budget, we conclude that at the end of $\frac{6N}{7}$ rounds, at least one party decodes safely.

Now suppose that only one party, say Alice, decodes safely at the end of $\frac{6N}{7}$ rounds. Then, in the last $\frac{N}{7}$ rounds, Bob is listening and Alice is sending. In this case, we claim that Bob's leaf that gets the majority of the votes at the end is the correct leaf. The reason is, suppose that Bob has t_B votes from the first $\frac{6N}{7}$ rounds and t'_B votes from the last $\frac{N}{7}$ rounds. Furthermore, suppose that the correct leaf had r_B votes from the first $\frac{6N}{7}$ rounds and r'_B votes from the last $\frac{N}{7}$ rounds. Then, the adversary has introduced at least $(\frac{3N}{7} - t_B) + (\frac{N}{7} - t'_B) + (t_B - r_B) + (t'_B - r'_B) = \frac{4N}{7} - r_B + r'_B$ errors. Since adversaries budget is at most $(\frac{2}{7} - \epsilon)N$, we get that $r_B + r'_B > \frac{2N}{7}$. Hence, since clearly Bob has at most $\frac{4N}{7}$ votes in total, the correct leaf has the majority. □

6. REFERENCES

[1] S. Agrawal, R. Gelles, and A. Sahai. Adaptive protocols for interactive communication. In *arXiv:1312.4182*, 2014.

[2] Z. Brakerski and Y. Kalai. Efficient interactive coding against adversarial noise. In *FOCS*, pages 160–166, 2012.

[3] Z. Brakerski and M. Naor. Fast algorithms for interactive coding. In *SODA*, pages 443–456, 2013.

[4] M. Braverman. Coding for interactive computation: progress and challenges. In *Allerton*, pages 1914–1921, 2012.

[5] M. Braverman. Towards deterministic tree code constructions. In *ITCS*, pages 161–167, 2012.

[6] M. Braverman and K. Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *ECCC:TR14–007*, 2014.

[7] M. Braverman and A. Rao. Towards coding for maximum errors in interactive communication. In *STOC*, pages 159–166, 2011.

[8] K.-M. Chung, R. Pass, and S. Telang. Knowledge preserving interactive coding. *FOCS*, 2013.

[9] M. Franklin, R. Gelles, R. Ostrovsky, and L. J. Schulman. Optimal coding for streaming authentication and interactive communication. In *CRYPTO*, pages 258–276, 2013.

[10] M. Ghaffari and B. Haeupler. Optimal Error Rates for Interactive Coding II: Efficiency and List decoding. In *arXiv:1312.1763*, 2013.

[11] M. Ghaffari, B. Haeupler, and M. Sudan. Optimal Error Rates for Interactive Coding I: Adaptivity and other settings. In *arXiv:1312.1764*, 2013.

[12] L. J. Schulman. Coding for interactive communication. *Trans. on Inf. Theory*, 42(6):1745–1756, 1996.

[13] M. Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180 – 193, 1997.