Adaptive Protocols for Interactive Communication

Shweta Agrawal  Ran Gelles  Amit Sahai
I.I.T Delhi  UCLA  UCLA
shweta.a@gmail.com  gelles@cs.ucla.edu  sahai@cs.ucla.edu

Abstract

In this work, we ask the question: how much adversarial error can protocols for interactive communication tolerate? This question was examined previously by Braverman and Rao [STOC 2011] for the case of “robust” protocols, where intuitively each party has a fixed and predetermined order of speaking and termination time. All previous work in coding for interactive communication focused on robust or “non-adaptive” protocols.

We consider a new class of protocols for Interactive Communication, namely, adaptive protocols. Such protocols adapt structurally to the noise induced by the communication channel in the sense that at every step of the protocol, parties may choose whether to “talk” or be silent, and whether to terminate or not according to the (possibly erroneous) messages they receive.

We distill meaningful and increasingly powerful models of adaptivity. The strongest is a fully adaptive model $\mathcal{M}_{\text{adp}}$ in which each party dynamically decides when to talk and when to terminate. Next, we define a partially adaptive model $\mathcal{M}_{\text{term}}$ in which each party retains the ability to terminate adaptively at any time but may only speak at fixed and predetermined times. A further specialization of $\mathcal{M}_{\text{term}}$ is the $\mathcal{M}_{\text{QwA}}$ model, which forces the parties to use adaptive termination in a specific, restricted manner (called the ”Quit-while-Ahead” method), further curtailing adaptivity. Last is the non-adaptive or robust model $\mathcal{M}_{\text{na}}$, which has been the focus of all previous works. Thus, we obtain a hierarchy of adaptivity notions: $\mathcal{M}_{\text{na}} \subseteq \mathcal{M}_{\text{QwA}} \subseteq \mathcal{M}_{\text{term}} \subseteq \mathcal{M}_{\text{adp}}$, where each model is identified by the set of protocols it contains.

We study upper and lower bounds on the permissible noise rate for our models. In the $\mathcal{M}_{\text{adp}}$ model, we exhibit a protocol that crucially uses the power of dynamic talk slot allocation as well as adaptive termination to tolerate noise rate up to $2/3$. In addition we show that if the parties share common randomness then there exists a protocol that tolerates an optimal noise rate of $1 - \epsilon$.

For the $\mathcal{M}_{\text{term}}$ model, we demonstrate the power of adaptive termination by providing a protocol that tolerates noise rate up to $1/3$. In the $\mathcal{M}_{\text{na}}$ model, Braverman and Rao [STOC 2011] exhibited a protocol that tolerates noise rate of $1/4 - \epsilon$, and a matching upper bound of $1/4$. We provide an upper bound of $1/4$ for a broader class of protocols in $\mathcal{M}_{\text{QwA}}$. We stress that our upper bound is novel since the impossibility of Braverman and Rao in the $\mathcal{M}_{\text{na}}$ model does not carry over to the $\mathcal{M}_{\text{QwA}}$ model.

Another interesting setting we consider is when the adversarial channel is an erasure channel. We demonstrate an $\mathcal{M}_{\text{adp}}$ adaptive protocol that can resist an optimal erasure rate of $1 - \epsilon$ with constant dilation, even without shared randomness.
1 Introduction

One of the fundamental questions considered by Computer Science is “What is the best way to encode information in order to recover from channel noise”? This question was studied most notably by Shannon, in a pioneering work [Sha48] which laid the foundation of the rich area of information theory. Shannon considered this question in the context of one way communication, where one party wants to transmit a message “once and for all” to another. More recently, in a series of beautiful papers, Schulman [Sch92, Sch93, Sch96] generalized this question to subsume interactive communication, i.e. the scenario where one party wants to “converse with” another in an interactive manner, so that each subsequent message depends on all messages exchanged thus far. Surprisingly, Schulman showed that, analogous to the case of one way communication, it is indeed possible to embed any interactive protocol π within a larger protocol π′ so that π′ computes the same function as π but additionally provides the requisite error correction to tolerate noise introduced by the channel.

The noise in the channel may be stochastic, in which error occurs with some probability, or adversarial, in which the channel may be viewed as a malicious party Eve who disrupts communication by injecting errors in the worst possible way. In this work we focus on adversarial noise. Schulman [Sch96] provided a construction that turns a protocol π with communication complexity $T$, to a noise-resilient π′ which communicates at most $O(T)$ symbols, and can recover from an adversarial (bit) noise rate of at most $1/240$. This result was later improved by Braverman and Rao [BR11], who provide a protocol that can recover from a (symbol) noise rate up to $1/4 - \varepsilon$ and also communicates at most $O(T)$ symbols. The noise tolerance of a protocol can be doubled to $1/2 - \varepsilon$ if the parties have some shared randomness unknown to the adversary, as shown by Franklin, Gelles, Ostrovsky and Schulman [FGOS13].

The above bounds naturally raise the question of whether these protocols are optimal with respect to noise resilience, and for the plain model (i.e., without any shared randomness), Braverman and Rao [BR11] provide a partial solution. They show that for a large and natural class of protocols, which they call “robust” protocols, $1/4$ is an upper bound on the tolerable noise-rate. Intuitively speaking, robust protocols are those which have a fixed and predetermined “order of communication”. In this class of protocols, each party knows at every time step whose turn it is to speak and whether the protocol has terminated, since these properties are fixed and independent of the noise introduced by the adversary. However, one can imagine more powerful, general protocols where a party may choose whether or not to speak in a given time slot depending on the noise observed. Whether this flexibility affords greater noise resilience to the protocol than in the robust case is unclear, and was explicitly left open by [BR11].

In this work, we tackle this problem and ask the question:

If protocols are not required to be robust, can they tolerate more adversarial error?

Intuitively, if a protocol is not robust, then the structure of the protocol itself must be able to adapt to noise. As such, we will find it convenient to refer to a robust protocol as a non-adaptive protocol, since the structure $^1$ of such a protocol does not adapt to noise. All previous works on interactive communication (see Related Works below), whether for stochastic or adversarial noise, restrict attention to non-adaptive protocols. A superficial examination suggests that such a restriction is necessary, as the proofs in those works rely upon the robustness of the protocol.

$^1$ We note that of course, any interactive protocol that successfully deals with noise must adapt to the noise in terms of the behavior of the parties within the protocol. We stress that in our work, the word adaptive refers to whether the structure of the protocol itself, in terms of its length and who speaks at which round, can itself be adapted based on the presence of noise.
For example in [BR11], the adversarial strategy against non-adaptive protocols works along the following broad outline: the adversarial channel Eve picks the player who speaks less, say Alice, and corrupts half her messages, causing her channel to Bob to practically have zero capacity. Since the protocol is non-adaptive, Eve can pick her target with confidence, and the messages she will inject are well defined. However in adaptive protocols, it is possible that Bob somehow detects that Eve is targeting Alice and dynamically “donates” some of his own slots to her, which turns Alice into the party that speaks more. Moreover, Bob might realize that the noise level is too high for the protocol to recover from and “abort”, i.e. stop communicating altogether. This effectively changes the “length” of the protocol and with it the budget Eve has. If Alice is capable of realizing this event and aborting as well, the (relative) noise rate at the end of the protocol might be higher than the allowed threshold, causing Eve’s attack to fall short.

1.1 Adaptive Protocols

Thus, two aspects of adaptive behavior are relevant to us: the ability to dynamically choose slots in which the party speaks and the power to stop participating at the protocol at any time. We will systematically examine these in turn, and define meaningful models that capture those notions. The family of robust, or non-adaptive protocols will be denoted as the $M_{na}$ model.

Adaptive talk slot allocation. The desire to assign talk slots dynamically, practically means that parties are allowed to “remain silent” instead of using their slots. In our first model $M_{adp}$, the parties have complete adaptivity and may choose dynamically when to speak, and when to remain silent. In this model, silence has special status in that it does not count towards communication complexity, that is, towards the adversarial budget, albeit it does count towards round complexity. We note that the ability to remain silent implies the ability to terminate the protocol at any time (by keeping silent and ignoring all incoming communication until reaching the last round).

In a model that permits adaptive choice of talk-slots, the role of silence becomes crucial and must be modeled carefully. For example, if we don’t count silence towards the adversarial budget, and also don’t allow the adversary to alter silence transmissions by turning non-silence transmissions into silence and vice-versa, then the players effectively obtain a “magical” symbol in the alphabet that is perfectly noise-resistant. It is easy to see that this gives a trivial protocol $\pi'$ defined over some alphabet $\Sigma$ that simulates protocol $\pi$ defined over a binary alphabet and tolerates noise perfectly: run $\pi$ transmitting each “1” as $\sigma \in \Sigma$, and any “0” as silence. Note that the protocol succeeds even when the adversary corrupts all non silent symbols (i.e. error rate 1).

Hence, the adversary in the $M_{adp}$ model is given full control over the channel and she may change any symbol, including silence. Any change made by the adversary is counted towards her noise budget. The adversary “wins” if at the termination of the protocol the parties fail to output the correct value, while the noise is less than some pre-determined fraction of the (non-silence) symbols communicated at that instance. We use the term relative noise rate to refer to the fraction of injected noise in the number of communicated symbols (rather than in number of rounds).

---

2If silence is counted towards the communication complexity and thus towards the adversarial corruption budget, then parties are being charged for communication all the time and cannot dynamically ‘donate’ their slots to the other party as motivated above. This partially-adaptive case leads to the $M_{term}$ model described below.

3This idea is well justified by practical applications such as wireless communication in which transmitting silence does not consume any power (see also [DFO10] and references within). Additionally, in settings such as sensor networks, the sensor is a weak device with limited power which may only need to communicate with the server infrequently. In such settings, it may be perfectly reasonable to make extensive use of the “free” symbol of silence to communicate information, obtaining protocol with very sparse use of non-silent, power-consuming symbols, at the cost of increasing the time the protocol takes to conclude.
Adaptive termination. In our second model $\mathcal{M}_{\text{term}}$, we deny the players any special, no-cost symbol, treating silence as equal to any other symbol of the alphabet. Thus, if Alice uses silence to communicate information, it also gets counted towards her communication complexity until she terminates. As in the previous model, Eve may alter silence just as she alters any other symbol. Thus, in this model the talk-slots of each party remain fixed and non-adaptive. The adaptivity of the protocols in this model lies only in each player’s ability to change the protocol’s length by terminating in an adaptive way.

Recall that a protocol $\pi$ is only expected to behave correctly when the noise rate is restricted below some bound, say $\beta$. Now, suppose a party detects that the channel’s noise rate at some point has exceeded the allowed threshold $\beta$. In that case, the party can “abort”, and may cause the termination of the protocol while the noise rate $> \beta$, exempting it from the task of succeeding. We refer to this strategy as the “quit-while-ahead” method to use abort. Thus, the ability to change the length of the protocol in an adaptive way, puts some constraints on the noise rate in any prefix of the protocol.

The ability to use abort dynamically may also be used by parties to communicate information, as illustrated by the following simple example. Consider the scenario of one way communication, i.e. Alice wants to transmit a single message to Bob. Alice may encode her message into her time of termination: to send the message $i$, she transmits arbitrary symbols until time $n^i$ (for a suitably chosen constant $n$) and then terminates. Bob decodes in a straightforward manner: if he receives at least $\epsilon n^i$ symbols, he decodes $i$. It is easy to verify that for any $\epsilon > 0$ and large enough $n$, this “protocol” achieves noise resilience of $1 - \epsilon$ (with respect to the amount of transmissions Alice made).

To understand the power afforded by adaptive termination as a “quit-while-ahead” mechanism, we define a restriction of the $\mathcal{M}_{\text{term}}$ model which nullifies the motivation for players to use abort to communicate information. This model, called the $\mathcal{M}_{\text{QwA}}$ is exactly the same as $\mathcal{M}_{\text{term}}$ except that the output of the players is defined to be $\bot$ if they abandon the game prematurely. Hence, players only abort if they detect so much noise that they do not expect to be able to recover.

Thus, we obtain a hierarchy of adaptivity notions: $\mathcal{M}_{\text{na}} \subseteq \mathcal{M}_{\text{QwA}} \subseteq \mathcal{M}_{\text{term}} \subseteq \mathcal{M}_{\text{adp}}$, where each model is identified by the set of protocols it contains, and subset is interpreted as set containment.

1.2 Noise Resilience

We study the noise resilience of adaptive protocols in the the $\mathcal{M}_{\text{adp}}$, $\mathcal{M}_{\text{term}}$ and $\mathcal{M}_{\text{QwA}}$ models, and several variants thereof.

In the $\mathcal{M}_{\text{adp}}$ model, we construct an adaptive protocol which crucially uses both the ability to remain silent as well as the ability to prematurely terminate the protocol to withstand noise rate $< 2/3$. This protocol has excellent communication complexity, however its round complexity may be very large with respect to the length of the optimal noiseless protocol. If restricted to protocols whose round complexity is linear in the length of the optimal noiseless protocol, we can tolerate up to $1/2 - \epsilon$ fraction of errors. The reason we can withstand twice the number of errors as in [BR11] is due to the special role played by silence in the model $\mathcal{M}_{\text{adp}}$. Because it comes for free, we develop a strategy to use silence extensively which we call “silence encoding”, thereby forcing the adversary to “pay twice” for each error she wishes to make.

In the case where the parties share common randomness, we devise a protocol that withstands an optimal $1 - \epsilon$ fraction of errors; this protocol also features a round complexity which is linear

\footnote{Note that for terminating the protocol, the other party needs to abort as well, before the relative noise drops below $\beta$.}
in the length of the optimal noiseless protocol. In addition, we study the power of protocols in the $\mathcal{M}_{\text{adp}}$ model against a restricted “rolling budget” adversary (an adversary whose relative corruption rate at any given time, is always within her budget) as well as a weakening of the model where the players have a special, incorruptible “abort” signal which lets them be in-sync about termination decisions. In both these variants we show a tight bound on the noise resilience by providing protocols that resist noise rates up to $1-\varepsilon$.

An interesting observation is that we can apply our methods to the setting of erasure channels and obtain a protocol with linear round complexity (constant dilation) and erasure resilience of $1-\varepsilon$ without the need for a shared randomness. It is instructive to consider the simpler case of erasure channel as this model is usually easier to analyze and may give insight into the more general noisy setting. We note that for non-adaptive protocols over erasure channels, $1/2$ is a tight bound on the noise: A noise of $1/2-\varepsilon$ is achievable via the Braverman-Rao protocol (see [FGOS13]), and a noise of $1/2$ is enough to completely erase all the communication of a single party, disallowing any interaction.

Our protocol for adaptive settings hints that adaptivity can double the resilience to noise (same as done by shared randomness, etc.). Our bounds for the $\mathcal{M}_{\text{adp}}$ model are summarized in Table 1.

| Model                  | Noise Resilience | Ref.  |
|------------------------|------------------|-------|
| $\mathcal{M}_{\text{na}}$ | $1/4$            | [BR11]|
| $\mathcal{M}_{\text{adp}}$ | $2/3$            | §3.1  |
| $\mathcal{M}_{\text{adp}}$ + shared randomness | $1$         | §3.3  |
| $\mathcal{M}_{\text{adp}}$ + constant dilation | $1/2$      | §3.2  |
| $\mathcal{M}_{\text{adp}}$ + rolling budget adversary | $1$       | §3.4  |
| $\mathcal{M}_{\text{adp}}$ + special abort signal | $1$       | §3.4  |
| $\mathcal{M}_{\text{adp}}$ over an erasure channel | $1$   | §3.3  |

Table 1: Summary of the noise resiliency of our protocols in the $\mathcal{M}_{\text{adp}}$ model. For any function $f$, and for any constant $c$ less than the resiliency, there exists a protocol that correctly computes $f$ over any channel relative noise rate $c$.

For the $\mathcal{M}_{\text{term}}$ model we show a protocol with noise resilience $1/3$, exceeding the impossibility result of [BR11]. Although our protocol has large round complexity (with respect to the noiseless case), it serves as an important proof of concept for the strength of this model. Indeed, the $1/4$ bound in the non-adaptive $\mathcal{M}_{\text{na}}$ model holds regardless of the round complexity of the protocol.

Furthermore, we show a new upper bound of $1/2$ on tolerable noise in that model. Previous proofs do not hold due to parties’ ability to communicate information about their input by choosing the time of termination, as mentioned above. We prove this bound by devising an attack that
corrupts both parties with rate $1/2$, and carefully showing that at least one of the parties must terminate before it learns the input of the other party.

Finally, we study the $\mathcal{MQwA}$ model which prevents the parties from using termination time in order to communicate information. In this case, the non-adaptive protocol of [BR11] suffices to withstand error rate up to $1/4 - \epsilon$, but the argument for impossibility of $1/4$ provided in [BR11] completely breaks down. We provide a new proof for an upper bound of $1/4$ for this case, albeit for the case of symmetric talk-slot allocation (as in [Sch93, BR11], etc.). The proof is obtained by devising an attack that corrupts half of the messages sent by a specific party. As opposed to the proof in [BR11] which also corrupts half of the messages of a specific party (e.g., the first half of the protocol), our attack changes the massages in a “smooth” way so that the adversary never goes above its budget, hindering the parties’ advantage in terminating prematurely. Our bounds for the $\mathcal{M}_{\text{term}}$ model are summarized in Table 2.

### 1.3 Overview of Our Constructions

Our starting point is a simple protocol in which first Alice sends her input to Bob encoded by some error-correction code, and then Bob sends his input back to Alice using the same code. It is clear that in the non-adaptive case, the protocol fails if Eve changes one of the codewords so that it decodes to a different message. Such an attack requires changing about half of the symbols of a codeword, leading to a bound of $1/4$, since both Alice and Bob use the same code parameters (otherwise, Eve attacks the shorter codeword and may fail the protocol with even less noise).

However, in the adaptive case Bob needs not encode his input with exactly the same parameters as Alice. Bob may adapt his codeword’s length according to the noise he sees in Alice transmissions: the more noise he sees, the smaller his codeword will be. This means that when Eve doesn’t change Alice’s transmissions (Bob sees no noise), his codeword will be long which forces Eve to make more corruptions. On the other hand, if Bob sees some noise in Alice transmissions, he knows that Eve has already “wasted” some of her budget. According to the amount of noise Bob sees, he may either abort or use a shorter codeword, limiting the additional budget Eve can get. With the right selection of Bob’s codeword length, we are able to show that the above protocol resists noise of up to $1/3$ (in the $\mathcal{M}_{\text{term}}$ model).

In the $\mathcal{M}_{\text{adp}}$ model we can do even better, taking advantage of the power of silence via a simple method of encoding we call silence encoding: each symbol is mapped to its characteristic vector, replacing each zero with a silence. Due to this encoding, Eve needs to corrupt twice the amount of communication in order to cause the receiver decoding a wrong message. If Eve makes less corruptions, her attack is noticed by the receiver and the transmission is marked as an “erasure”. It is well known that erasures are “twice as easy” to correct than errors (see, e.g., [FGOS13]), which again leads to doubling the amount of corruptions the protocol can resist.

Consequently, combining silence encoding with the idea of varying-length codewords gives a protocol that resists noise rates up to $2/3$ (with double exponential round complexity with respect to the noiseless case). Concatenating silence encoding with the Braverman-Rao protocol leads to a protocol that resists noise rate up to $1/2$ (with a linear round complexity).

Under the assumption of shared randomness we can do even better, and get a protocol with a linear round complexity that resists noise up to $1 - \epsilon$. The idea is to use a variant of the Braverman-Rao protocol and add, after each label transmission, a “control message” that indicates whether or not the last label was received. In case the label wasn’t received, the sender will re-transmit it again and again, forcing Eve to waste a large amount of budget in order to corrupt a single label. The problem now is that Eve can abuse these control messages and request re-transmissions of labels that were not corrupted (which increases the communication and thus her budget). We avoid this
attack by using a weak message authentication code (MAC) known as the Blueberry code [FGOS13]. This randomized code effectively prevents Eve from choosing the message a corrupted codeword decodes to, rendering the corruption of control messages useless.

1.4 Related Work

Closest to our work is concurrent and independent work by Ghaffari, Haeupler, and Sudan [GHS13] which simultaneously initiates the study of adaptive protocols. They model adaptivity differently from us: in their setting the length of the protocol is fixed, yet at each round a party can decide whether to speak or listen. If both parties speak at the same round no symbol is transferred, and if both listen at the same round, they receive some adversarial symbol not counted towards the adversary’s budget. Ghaffari et al. provide a protocol which suffers at most a quadratic dilation and succeeds as long as at most fraction 2/7 of the rounds is corrupted. They also show that in their model, no protocol can resist noise rates above 2/7. Allowing the parties to pre-share randomness increases the admissible noise to 2/3. Ghaffari and Haeupler [GH13] further improve the round complexity of their protocols to near linear (instead of quadratic) while tolerating optimal noise rates in their model.

As mentioned above, the study of coding for interactive communication was initiated by Schulman [Sch92, Sch93, Sch96] who provided protocols for interactive communication using tree codes (see Appendix A for a definition and related works). In his original work, Schulman considered both the stochastic as well as adversarial noise model, and for the latter provided a protocol that resists (bit) noise rate 1/240. Braverman and Rao [BR11] improved this bound to 1/4 by constructing a different tree-code based protocol (which is efficient except for generation of tree codes). Very recently, Braverman and Efremenko [BE14] considered the case where α fraction of the symbols from Alice to Bob are corrupted and β fraction of the symbols in the other direction are corrupted. For any point (α, β) ∈ [0, 1] they determine whether or not a constant rate interactive communication is possible. This gives a complete characterization of the noise bounds for the non-adaptive case.

Over the last years, there has been a large interest in interactive protocols, considering various properties of such protocols such as their efficiency [BK12, BN13, GH13], their noise resilience under different assumptions and models [FGOS13, BTT13], their communication rate [KR13] and other properties, such as privacy [CPT13, GSW14] or list-decoding [GHS13, GH13, BE14]. We stress that all the above works assume robust (non-adaptive) setting.

Interactive (noiseless) communication using a special ‘silence’ symbol was introduced by Dhulipala, Fragouli, and Orlitsky [DFO10], who considered the communication complexity of computing symmetric functions in the multiparty setting. In their general setting, each symbol σ in the channel’s alphabet has some weight wσ ∈ [0, 1] and the weighted communication complexity, both in the average and worst case, is analyzed for a specific class of functions. Impagliazzo and Williams [IW10] also consider communication complexity given a special silence symbol for the two-party case. They establish a tradeoff between the communication complexity and the round complexity. Additionally, they relate these two measures to the “standard” communication complexity, i.e., without a special silence symbol.

2 Models for Adaptive Protocols

In this section, we provide formal definitions for our models.

We assume Alice and Bob wish to compute some function \( f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathcal{Z} \) where Alice holds some input \( x \in \mathcal{X} \) and Bob holds \( y \in \mathcal{Y} \). The sets \( \mathcal{X}, \mathcal{Y} \) and \( \mathcal{Z} \) are assumed to be of finite size. To compute \( f \), the parties run a protocol \( \pi = (\pi_A, \pi_B) \), over a channel which is controlled by a
malicious Eve. At every step of the protocol, \( \pi \) defines a message over some alphabet \( \Sigma \) to be transmitted by each party as a function of the party’s input, and the received messages so far. Unless stated otherwise, we assume that \( \Sigma \) is a finite set of size \( O(1) \), that is, its size is independent of \( f \).

2.1 The \( \mathcal{M}_{\text{adp}} \) Model

We now describe the “fully adaptive” model \( \mathcal{M}_{\text{adp}} \) in which players dynamically choose at every time step whether to speak or remain silent. It is important to emphasize that in this model silence is not counted towards communication complexity nor towards the adversary’s budget (but is counted towards round complexity).

The salient features of the model are as follows (described for party \( A \); the case for \( B \) is symmetric):

- In a given round \( i \), \( A \) decides whether to speak or remain silent. If \( A \) speaks then we say Alice sends a message \( a_i \in \Sigma \), and if \( A \) is silent, \( a_i = \emptyset \).

- The adversary may corrupt any symbol, including silence, transmitted by either party. Thus, the channel acts upon transmitted symbols via the function \( \text{Ch} : \Sigma \cup \{\emptyset\} \to \Sigma \cup \{\emptyset\} \),

\[ \text{conditioned on the parties input, Eve’s random coins and the transcript so far.} \]

- The corresponding symbol received by \( B \) is thus \( \tilde{a}_i = \text{Ch}(a_i) \).

- We assume the protocol terminates after a finite time. There exists a number \( R_{\text{max}} \) at which both parties terminate and output a value as a function of their input and the communication.

For a specific instance of the protocol we denote the messages sent by the parties \( M = (a_1, b_1, a_2, b_2, \ldots) \) in that instance, and the Noise Pattern \( E = (e_{a_1}, e_{b_1}, \ldots) \) so that \( e_{a_i} = \perp \) if \( \text{Ch}(a_i) = a_i \) and otherwise, \( e_{a_i} = \text{Ch}(a_i) \), and similarly for \( e_{b_i} \). We will treat \( E \) and \( M \) as strings of length \( 2R_{\text{max}} \) and sometimes refer to the \( i \)-th character in that string as \( E_i \) and \( M_i \). Next we formally define some important measures of the protocol.

**Definition 2.1.** For any protocol \( \pi \) in the \( \mathcal{M}_{\text{adp}} \) model, and for any specific run of the protocol with noise pattern \( E \) we define:

**Communication Complexity:**

\[
\text{CC}_{\pi}^{\text{adp}}(E) \overset{\Delta}{=} |\{i \leq 2R_{\text{max}} | M_i \neq \emptyset\}|,
\]

where \( M \) is the message string assuming the noise pattern \( E \).

**Noise Complexity:**

\[
\text{NC}_{\pi}^{\text{adp}}(E) \overset{\Delta}{=} |\{i \leq 2R_{\text{max}} | E_i \neq \perp\}|.
\]

**Relative Noise Rate:**

\[
\text{NR}_{\pi}^{\text{adp}}(E) \overset{\Delta}{=} \frac{\text{NC}_{\pi}^{\text{adp}}(E)}{\text{CC}_{\pi}^{\text{adp}}(E)}.
\]

To simplify notation we write \( \text{Ch} \), yet we implicitly use a different \( \text{Ch} \) for each channel instantiation.

Note that the above definition is per a single instance, where the inputs \( x, y \) are implicit. We can then define the communication complexity of the protocol as \( \text{CC}_{\pi}^{\text{adp}} = \max_{(x,y)} \max_E \text{CC}_{\pi}^{\text{adp}}(E) \).
Although we try to keep the above definition general, all of our protocols actually use a unary alphabet. In this case, we can denote silence as 0 and the non-silence pulse as 1. The messages sent by the parties are now $M \in \{0,1\}^{2R_{\text{max}}}$ and for any noise pattern $E \in \{0,1\}^{2R_{\text{max}}}$ the transcript (the received messages) can be written as the bitwise xor $M \oplus E$. If we define $|S|$ as the number of ones in the string $S$ (its Hamming weight), we have $\text{NR}_E^\pi(E) = |E|/|M|$. Note the the relative noise rate may exceed 1.

### 2.2 The $\mathcal{M}_{\text{term}}$ Model

The $\mathcal{M}_{\text{term}}$ model captures the power of a protocol in which players have fixed talk-slots but may terminate at different times depending on observed error, thus adapting the length of the protocol to the observed noise by either extending it or cutting it short. Within this model, silence is treated like an ordinary symbol and counts towards communication complexity as well as towards the adversary’s budget. However, silence is still special due to its role after a player, say Alice, terminates while Bob has still not terminated. Post termination, Alice’s input is set to the silence symbol, which may still be corrupted by Eve, adding to her error count, but it no longer counts towards Alice’s communication complexity.

Formally, we denote by $I_A, I_B \subseteq \mathbb{N}$ the fixed sets of talk-slots assigned to party A and B respectively. Note that $I_A$ and $I_B$ may be overlapping but we may assume that there are no “gaps” in the protocol, i.e. $I_A \cup I_B = \mathbb{N}$. The channel expects an input from party $P \in \{A,B\}$ only during its talk slots $I_P$. The $\mathcal{M}_{\text{term}}$ model can be defined in the following way (described for party $A$; the case for party $B$ is equivalent).

- In a given round $i \in I_A$, if $A$ has not terminated, it transmits $a_i \in \Sigma \cup \{\emptyset\}$.
- At the beginning of any round $i \in \mathbb{N}$, $A$ may decide to terminate. In that case it outputs some $z \in Z \cup \{\bot\}$, sets $\text{TER}_A = i$ and stops participating in the protocol. This is an irreversible decision.
- In every round $i \in I_A$ where $i \geq \text{TER}_A$, $A$’s input to the function $\text{Ch}(\cdot)$ is defined as $a_i = \emptyset$.
- The adversary may corrupt any symbol, including silence, transmitted by either party. Thus, the channel acts upon transmitted symbols via the function $\text{Ch}: \Sigma \cup \{\emptyset\} \rightarrow \Sigma \cup \{\emptyset\}$, conditioned on the parties input, the channel’s random coins, and the transcript so far. Note that even after $A$ has terminated, $a_i = \emptyset$ is sent over the channel and still might be corrupted by Eve.

**Definition 2.2.** For any protocol $\pi$ in the $\mathcal{M}_{\text{term}}$ model, and any noise pattern $E$ we define:

**Communication Complexity:**

$$\text{CC}_\pi^\text{term}(E) \overset{\Delta}{=} |[\text{TER}_A - 1] \cap I_A| + |[\text{TER}_B - 1] \cap I_B|,$$

where $[n]$ is defined as the set $\{1, 2, \ldots, n\}$.

**Round Complexity:**

$$\text{RC}_\pi^\text{term}(E) \overset{\Delta}{=} \max(\text{TER}_A, \text{TER}_B).$$

**Noise Complexity:**

$$\text{NC}_\pi^\text{term}(E) \overset{\Delta}{=} \left| \left\{ i \in I_A \mid i < \text{RC}_\pi^\text{term}, \text{Ch}(a_i) \neq a_i \right\} \right| + \left| \left\{ i \in I_B \mid i < \text{RC}_\pi^\text{term}, \text{Ch}(b_i) \neq b_i \right\} \right|.$$

**Relative Noise Rate:**

$$\text{NR}_\pi^\text{term}(E) \overset{\Delta}{=} \frac{\text{NC}_\pi^\text{term}(E)}{\text{CC}_\pi^\text{term}(E)}.$$
As before, we consider only protocols of bounded length. That is, we assume there exists a global constant $N_{\text{max}}$ and that for any input and noise pattern $RC \leq N_{\text{max}}$. We may consider a symmetric variant of the above model, called symmetric $M_{\text{term}}$ in which $I_A = I_B = N_{\text{max}}$.

2.3 The $M_{\text{QwA}}$ Model: the power of aborting

As discussed in Section 1, aborting may be used by a player in order to quit the protocol when there is no hope of success, just to defeat the adversary. This scenario can be seen as a “game” between the two parties against the channel: the channel tries to maintain the noise rate below the limit while the parties wish to force the noise rate above the limit. Since the noise rate is the fraction of injected errors to communicated symbols, we have that one player (Eve) controls the numerator of this ratio while the other (Alice and Bob collectively) control the denominator, making it hard to predict who has the bigger advantage! To understand this question and quantify the power of abort as a “quit while ahead” mechanism, we study a variant of the $M_{\text{term}}$ model, in which the players are forced to output $\perp$ (and fail the protocol) if they abort before a predetermined round. This restricts the party to abort only when indeed the noise level has exceeded the allowed threshold.

We will refer to this restriction of $M_{\text{term}}$ as $M_{\text{QwA}}$, which is exactly the same as $M_{\text{term}}$ as defined above, except that if $\min\{\text{TER}_A, \text{TER}_B\} \neq N_{\text{max}}$ then the parties’ output is defined as $\perp$.

3 Noise Resilience in the $M_{\text{adp}}$ Model

In this section we discuss the power of adaptive slot-allocation. We show that every function can be computed by an $M_{\text{adp}}$ protocol that can suffer noise-rate less than $2/3$, yet this protocol has a large round complexity. For the setting of efficient round complexity we show that a protocol can resist noise rate less than $1/2$. Additionally, we show that if the parties share randomness unknown to the adversary, any protocol of length $T$ in the noiseless model can be simulated in the $M_{\text{adp}}$ model and resist noise rate of up to $1 - \varepsilon$ while keeping the round complexity $O(T)$.

3.1 A protocol for relative noise rates $< 2/3$

The main technique used in this section is a simple code that takes advantage of the ‘silence’ symbols, which we call silence encoding.

**Definition 3.1.** Let $X = \{x_1, x_2, \ldots, x_n\}$ be some finite, totally-ordered set. The silence encoding is a code $SE_1 : X \to (\Sigma \cup \{\emptyset\})^n$ that encodes $x_i$ into a string $y_1, \ldots, y_n$ where $\forall j \neq i \ y_j = \emptyset$ and $y_i \neq \emptyset$.

The $k$-silence encoding is a code $SE_k : X \to (\Sigma \cup \{\emptyset\})^{kn}$ that encodes $x_i$ into a string $y_1, \ldots, y_{kn}$ where all $y_j = \emptyset$ except for the $k$ indices $y_{(i-1)k+1}, y_{(i-1)k+2}, \ldots, y_{ik} \in \Sigma^k$.

Decoding a $k$-silence-encoded codeword is straightforward. The receiver tries to find a message $x_i$ whose encoding minimizes the Hamming distance to the received codeword. If the string that minimizes the distance is not unique, the decoder marks this event as an erasure and outputs $\perp$. The event where the decoder decodes $x_j \neq x_i$ is called an error. Both encoding and decoding can be done efficiently.

We note the following interesting property of $k$-silence encoding: in order to cause ambiguity in the decoding (i.e., an erasure), the adversary must change at least $k$ indices in the codeword. Moreover, in order to make the decoder output an incorrect value (i.e., an error), the adversary must make at least $k+1$ changes to the codeword. Specifically for $k = 1$, a single corruption always...
causes an *erasure* (i.e., ambiguity), while in order to make a decoding *error*, at least 2 transmissions must have been changed.

We are now ready to prove our main theorem of this section.

**Theorem 3.2.** Let $X, Y, Z$ be some finite sets. For any function $f : X \times Y \to Z$ there exists an adaptive protocol $\pi$ that correctly computes $f$ as long as the relative noise rate is below $2/3$.

**Proof.** Our protocol is composed of two parts: in the first part Alice communicates her input to Bob and in the second part Bob communicates his input to Alice. However, after the first part, the protocol estimates the error injected and proceeds to the second part only if the noise-rate is low enough to correctly complete the protocol, or is high enough so that the adversary will surely exceed its budget by the time the protocol ends. In addition, Bob’s message crucially depends on the amount of error Eve introduced in the channel.

Assume the channel is defined over some alphabet $\Sigma$ and denote one of the alphabet’s symbols by ‘$\sigma$’. For any $k \in \mathbb{N}$ define $\pi$ on inputs $x_i, y_j \in X \times Y$ in the following way:

1. Alice communicates a $k$-silence encoding of her input, namely, she waits $k \cdot (i - 1)$ rounds and then sends the symbol $\sigma$ for $k$ consecutive rounds.

2. Bob waits until round $k |X|$ and decodes the codeword sent by Alice. Bob adaptively chooses his actions according to the following cases:
   (a) if there is ambiguity regarding what $x_i$ is, Bob aborts.
   (b) otherwise, Bob decodes some $x_i'$. Let $t$ be the difference between the number of $\sigma$ symbols Bob received during those rounds that “belong” to $x_i'$ and the number of $\sigma$’s received during the rounds that “belong” to a value $x_i''$, where $x_i''$ is the 2nd best decoding of the received codeword (when decoding by minimizing Hamming distance)
   Bob communicates his input $y_j$ using the following $2t$-silence encoding: he waits $2k \cdot (j - 1)$ rounds and then sends the symbol $\sigma$ for $2t$ consecutive rounds.
   Then, Bob outputs $f(x_i', y_j)$ and terminates.

3. Alice waits until round $R_{\text{max}} \triangleq k |X| + 2k |Y|$, and decodes the codeword sent by Bob. If there is ambiguity regarding the value of $y_j$, Alice aborts. Otherwise, she obtains some $y_j'$. Alice then outputs $f(x_i, y_j')$ and terminates.

Let us analyze what happens at round $k |X|$. As mentioned above, in order to cause ambiguity at that round, Eve must change at least $k$ transmissions. In this case Bob aborts at round $k |X|$; observe that neither of the parties communicates any symbol after round $k |X|$, thus their total communication for this instance is $k$ symbols. This implies noise rate of at least 1.

If, on the other hand, at round $k |X|$ there was no ambiguity, one of two things must have happened: either Bob correctly decodes Alice’s input, or he decodes a wrong input. First assume the latter, which implies that at least $k + 1$ corruptions were done. Since there is no ambiguity, we know that $t > 0$ and it must hold that Eve made $e \geq k + t$ corruptions. Then, by the end of the protocol, the relative noise rate is at least $\frac{e}{k+2(e-k)}$. This value decreases as $e$ increases, up till the point where $e = 2k$ at which it gets a minimal value of $2/3$. Eve has no incentive to perform more than $e = 2k$ corruptions, this will only increase the relative noise rate without changing the actions of Bob.

Now assume Bob decodes the correct value, thus Eve wins only if Alice decodes a wrong value from Bob or aborts. We consider two cases. (i) If Eve has corrupted $e < k$ symbols by round $k |X|$, then Bob will send his input via $2t$-silence encoding, where $t \geq k - e$. Thus, in order for Alice to
decode a wrong value (or abort), Eve must perform at least additional $2t$ corruptions, yielding a relative noise rate of at least $\frac{e + 2t}{k + 2t}$. Under the constraints that $0 \leq e \leq k - 1$ and $k - e \leq t \leq k$, it is easy to verify that

\[ \frac{e + 2t}{k + 2t} \geq 1 - \frac{t}{k + 2t} \geq \frac{2}{3}. \]

(ii) If Eve has made $e \geq k$ corruptions by round $k|X|$, yet Bob decoded the correct value, Eve will have to corrupt additional $2t$ symbols to cause confusion at Alice’s side. This implies a relative noise rate of at least

\[ \frac{e + 2t}{k + 2t} \geq \frac{k + 2t}{k + 2t} = 1. \]

\[ \square \]

### 3.2 A protocol with linear round complexity for relative noise $< 1/2$

While the above protocol obtains noise rate resilience of $2/3$ and excellent communication complexity, it has double exponential round complexity with respect to the round complexity of the best noiseless protocol. One way to ensure that the protocol have respectable round complexity is to restrict the number of “silence” rounds that a protocol may use. Our next theorem considers this setting. It shows that for any $\varepsilon > 0$ we can emulate any protocol $\pi$ of length $T$ (defined in the noiseless model) by a protocol $\Pi$ in the $M_{adp}$ model, which takes at most $O(T)$ rounds and resists noise rate of $1/2 - \varepsilon$.

**Theorem 3.3.** For any constants $\varepsilon > 0$ and for any function $f$, there exists an interactive protocol in the $M_{adp}$ model with round complexity $O(CC_f)$, that correctly computes $f$ if the adversarial relative corruption rate is at most $1/2 - \varepsilon$.

The protocol follows the emulation technique set forth by Braverman and Rao [BR11], and requires a generalized analysis for channels with errors and erasures as performed in [FGOS13] albeit for a completely different setting. The key insight is that silence encoding forces the adversary to pay twice for making an error (or otherwise causing “only” an erasure). This allows doubling the maximal noise rate the protocol resists. The proof appears in Appendix A.

### 3.3 A Protocol for relative noise $< 1$ with linear round complexity assuming shared randomness

We now extend the model by allowing the parties to share a random string, unknown to the adversary. We show that shared randomness setup allows the noise to go as high as $1 - \varepsilon$. Formally, we show

**Theorem 3.4.** For any small enough constants $\varepsilon > 0$ and for any function $f$, there exists an interactive protocol in the $M_{adp}$ model with round complexity $O(CC_f)$ such that, if the adversarial relative corruption rate is at most $1 - \varepsilon$, the protocol correctly computes $f$ with overwhelming success probability over the choice of the shared random string.

Before proving the theorem, let us begin with a short motivation for our construction. Our starting point is the protocol of Theorem 3.3, i.e., concatenating [BR11] with silence encoding. We need to deal with two issues: deletion of labels (erasures) and altering labels (errors). First we take care of the errors, which is done by the technique of the so called Blueberry code [FGOS13]. A Blueberry code with parameter $q$ encodes each symbol in $\Sigma$ into a random symbol in $\Gamma$, where $|\Sigma|/|\Gamma| < q$. Since the mapping $\Sigma \rightarrow \Gamma$ is unknown to the adversary, any change to the coded label
will be detected with probability $1 - q$ and the transmission will be considered as an erasure. By choosing $q$ to be small enough (as a function of $\varepsilon$), we can guarantee that the adversary cannot do much harm by changing symbols.

Next, we need to deal with the more problematic issue of erasures. The problem is that the [BR11] protocol is symmetric, that is, Alice and Bob speak the same amount of symbols. Thus, a successful corruption strategy with relative noise rate $1/2$ just deletes all Alice’s symbols. To overcome this issue we need to “break” the symmetry. We will do that by sending indication of deleted labels: If Alice’s label was deleted, Bob will tell her so, and she will send more copies of the deleted label. If all of these repeated transmissions are deleted again, Bob will indicate so and Alice will send again more and more copies of that label. This continues until the total amount of re-transmissions surpasses the amount of transmissions in the noiseless scenario. This breaks the symmetry: if Eve wishes to delete all the copies she will end up causing Alice to speak more, which forces Eve to delete the additional communication as well, which forces her into increasing the average relative noise rate she introduces.

The remaining issue is to prevent Eve from causing the parties to communicate many symbols without her making many corruptions, e.g., by forging Bob’s feedback to make Alice send unnecessary copies of her labels. This is prevented by the Blueberry code: such an attack succeeds with very small probability that makes it unaffordable.

Proof. (Theorem 3.4.) The protocol is based on the protocol of Theorem 3.3 (i.e., on the scheme of [BR11]), yet replacing each label transmission with an adaptive subprotocol that allows re-transmissions of deleted symbols. Furthermore, each transmission is encoded via a Blueberry code [FGOS13], which allows the parties to notice Eve’s attack most of the times.

Fix $\varepsilon > 0$ to be some small enough constant. For any noiseless protocol $\pi$ of length $T$ we will simulate $\pi$ in the $\mathcal{M}_{\text{adp}}$ model using the following procedure, defined with parameters $k = \varepsilon^{-1}, t = k\varepsilon^{-1}, q < (kt)^{-2}$.

1. The parties perform the scheme [BR11] for $N = O(T/\varepsilon)$ rounds.

2. In each round, each label is transmitted via the following process:

   (a) The sender encodes the label via a Blueberry code with parameter $q$, and sends it encoded with a $k$-silence encoding.

   (b) Repeat for $t$ times:

   • if the receiver hasn’t received a valid label, he sends back a “repeat-request” encoded using the Blueberry code and a 1-silence encoding. [Otherwise, he does nothing.]

   • each time the sender gets a valid repeat-request, he sends the original message transmission again encoded with a (fresh instance of) Blueberry code, and $k$-silence encoding.

   (c) The receiver sets that round’s output to be the first valid label he received during step (b), or ⊥ if all the $t$ repetitions are invalid (either deleted, or marked invalid by the Blueberry code).

First we note that indeed the protocol takes at most $O(N) = O(T)$ rounds, since the BR protocol takes $O(N)$ rounds, each of which is expended by at most $O(kt|\Gamma|) = O(1)$ rounds.

We now show that the above protocol achieves noise rate up to $1 - O(\varepsilon)$. We split the protocols into epochs, where each epoch corresponds to a single label transmission of the [BR11] simulation (i.e., the label and its repetitions are a single epoch).
Next, we divide the epochs into two disjoint sets: deleted and undeleted epochs. The former consists of any epoch in which the receiver outputs ⊥. The set of undeleted epochs is split again into two disjoint sets: “lucky” and “non-lucky”. A lucky epochs is any epoch in which Eve’s corruption is not detected by the Blueberry code.

For each epoch \( e \) we define \( c(e) \) as the communication made by the parties in this epoch, and \( r(e) \) as the rate of noise made by Eve in this epoch, that is, the number of corruptions Eve makes in the epoch is \( r(e)c(e) \). The global noise rate is a weighted average of the noise rate per epoch, where each epoch is weighted by the communication in that epoch.

Fix a noise pattern \( E \) for the adversary, and assume that the simulation process fails with noise pattern \( E \). Lemma A.3 tells us that the number of deleted epochs plus twice the number of incorrect epochs (where the receiver outputs a wrong label) must be at least \( (1 - \varepsilon)N \). Note that incorrect epoch must be lucky, and the probability for an epoch to be lucky is at most \( q \cdot 2^t \), since there are at most \( 2^t \) messages in each epoch and a probability at most \( q \) to break the Blueberry code for a single message. Hence, for our choice of \( q \), the number of deleted epochs is at least \( (1 - 3\varepsilon)N \) with overwhelming probability.

First, we analyze deleted epochs. The following is immediate.

**Claim 3.5.** In each deleted epoch \( e \), the noise rate is

\[
  r(e) \geq \frac{k(t + 1)}{(k + 1)t + k} = 1 - \frac{t}{kt + t + k} \geq 1 - \varepsilon
\]

while the communication is \( c(e) \geq k + t > \varepsilon^{-2} + \varepsilon^{-1} \).

Next we analyze the non-deleted epochs. We note that \( c(e) \) in this case ranges between \( \varepsilon^{-1} \) to \( \varepsilon^{-3} \), and we now relate \( r(e) \) to \( c(e) \). It is crucial that whenever \( c(e) \) exceeds \( \varepsilon^{-2} \), the amount of noise will be high enough, to maintain a global noise rate of almost one.

First, we deal with “lucky” epochs. For simplification, we assume that if \( e \) is “lucky”, then the noise rate is 0, and the communication is the maximal possible \( kt + t + 1 \). We will choose \( q \) to satisfy \( 2qt \cdot (kt + t + 1) \ll 1 \), so that the effective noise added by such epochs is negligible.

Then, we need to relate \( c(e) \) and \( r(e) \) for the rest of the epochs.

**Claim 3.6.** If \( e \) is a non-lucky non-deleted epoch, then

\[
  r(e)c(e) \geq \max(0, c(e) - 2k - 1).
\]

**Proof.** If the epoch is not lucky, then it must have concluded correctly (Bob has eventually received the correct label). Thus the only attack Eve can perform in order to increase \( c(e) \), is to delete Bob’s reply-request and Alice’s answers up to some point (in addition to corrupting Alice original label).

Thus, Eve must block the first \( k \) symbols sent by Alice (as long as \( c(e) > k \)), but she must not block the last \( k \) symbols made by Alice. Assume Eve only blocks Bob’s reply-requests. Then she can block \( x \leq t - 1 \) such requests and make \( x \) corruptions out of total communication \( x + 2k + 1 \).

Another possible attack is to let \( y \) of Bob’s repeat-request go through (as long \( y + x \leq t - 1 \)) but delete Alice’s replies (again, except for the last one). This will cause \( yk + x \) corruptions out of communication \( (y + 2)k + x + 1 \).
The relative noise rate caused by any noise pattern $E$ that results in a failed instance of the simulation is thus bounded by

$$\text{NR}_{\text{adp}}(E) \geq \frac{\sum_{e: \text{deleted}} r(e)c(e) + \sum_{e: \text{ lucky}} 0 + \sum_{e: \text{ correct non-lucky}} \max(0, c(e) - 2k - 1)}{\sum_{e} c(e)}$$

$$\geq \frac{(1 - \varepsilon) \sum_{e: \text{ deleted}} c(e) + \sum_{e: \text{ correct non-lucky}} \max(0, c(e) - 2k - 1)}{\sum_{e: \text{ deleted}} c(e) + \sum_{e: \text{ lucky}} (kt + t + 1) + \sum_{e: \text{ correct non-lucky}} c(e)}.$$  

Recall that with very high probability $> (1 - 3\varepsilon)N$ epochs are deleted and at most $2qtN$ epochs are lucky. Thus $\sum_{e: \text{ lucky}} (kt + t + 1)$ is upper bounded by $2qt(kt + t + 1)N$ with high probability. We take $q \ll 1/2t(kt + t + 1)$ and neglect this term in the denominator.

Now split the correct non-lucky epochs to two sets: $B_0 = \{e \mid c(e) \leq \varepsilon^{-1.5}\}$ contains epochs with “low” communication and $B_1$ contains all the other correct non-lucky epochs (with “high” communication). For small enough $\varepsilon$,

$$\text{NR}_{\text{adp}}(E) \geq \frac{(1 - \varepsilon) \sum_{e: \text{ deleted}} c(e) + \sum_{B_1} (c(e) - 2\varepsilon^{-1})}{\sum_{e: \text{ deleted}} c(e) + |B_0|\varepsilon^{-1.5} + \sum_{B_1} c(e)}$$

$$\geq \frac{(1 - \varepsilon) \sum_{e: \text{ deleted}} c(e) - |B_1|2\varepsilon^{-1}}{\sum_{e: \text{ deleted}} c(e) + |B_0|\varepsilon^{-1.5}}$$

$$\geq 1 - O(\varepsilon),$$

since $\sum_{e: \text{ deleted}} c(e) \geq (1 - 3\varepsilon)(\varepsilon^{-2} + \varepsilon^{-1})N$ and $|B_0| + |B_1| \leq (1 + 3\varepsilon)N$, with very high probability. \qed

It is immediate that the same result holds for protocols over the erasure channel: such channels can only make “erasures” to begin with, so there is no need for using Blueberry codes, nor for sharing randomness.

**Corollary 3.7.** For any small enough constant $\varepsilon > 0$ and for any function $f$, there exists an interactive protocol in the $M_{\text{adp}}$ model over an erasure channel that has round complexity $O(CC_f)$ and that correctly computes $f$ as long as the adversarial relative erasure rate is at most $1 - \varepsilon$.

### 3.4 Relaxations of the $M_{\text{adp}}$ model

In this section we consider several relaxations of the fully adaptive model as described below.

**Restricted “rolling budget” adversary.** To begin, we consider the power of the “rolling budget adversary” — a restricted adversary who is constrained not to go over his relative error budget at any time. Thus, at any given step $t$, the relative noise rate at $t$ is below the allowed noise threshold. We show that a rolling budget adversary is too weak: i.e., for any function there exists a protocol that resists relative noise-rate of up to 1, assuming a rolling budget adversary. This implies that any adversary that deploys a successful attack in the $M_{\text{adp}}$ model must exceed its allowed relative noise-rate.

**Proposition 3.8.** For any $\varepsilon > 0$ and any function $f : X \times Y \to Z$ there exists a protocol $\pi$ in the $M_{\text{adp}}$ model that correctly computes $f$ if the relative noise rate, at any given round, is at most $1 - \varepsilon$.

**Proof.** Consider the protocol where Alice 1-silence-encodes her input, and then Bob $k$-silence-encodes his input, where $k > (1 - \varepsilon)/\varepsilon$. Note that Eve is not allowed to corrupt any transmission until Bob sends his first symbol, thus Alice message is correctly decoded. Moreover, Eve’s budget in this case is at most $(1 - \varepsilon)(k + 1) < k$ thus Eve cannot prevent Alice from correctly decoding Bob’s message, due to the properties of the $k$-silence encoding. \qed
Synchronous Abort. Assume the parties hold a special, incorruptible “abort” signal that is triggered when either one of the parties terminates, so that the other party is immediately notified of this event. Once this signal is triggered, the channel becomes inactive, the parties cannot communicate anymore, and both are required to output some value. In the context of distributed processors, where the communication channel models a noisy bus, such a signal models an interrupt signal that directly connects the processors, and is less prone to noise. We show that such a synchronous abort signal is too strong.

Proposition 3.9. For any \( \varepsilon > 0 \) and any function \( f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z} \) there exists a protocol \( \pi \) in the \( \mathcal{M}_{\text{adv}} \) model that correctly computes \( f \) if the relative noise rate is at most \( 1 - \varepsilon \), assuming both parties are connected by an abort signal.

Proof. In some sense, the abort signal trivializes the problem since it allows to communicate information without increasing the communication complexity: Indeed, if Alice silence encodes her input (sending 1 symbol), and then Bob silence-encodes his input, but instead of communicating a symbol he aborts at the corresponding round, Alice still learns his input, and since only a single symbol is sent in this case, the protocol resists any noise rate < 1. We now show another protocol that does not rely on using the abort signal for communication — that is, the abort signal is triggered only when one of the parties “aborts” and outputs \( \perp \).

Consider the following protocol on inputs \( x_i, y_j \in \mathcal{X} \times \mathcal{Y} \).

1. Alice 1-silence encodes her input \( x_i \), namely, she waits \((i - 1)\) rounds and then sends the symbol \( \sigma \).

2. Bob, waits till round \(|\mathcal{X}|\), tries to decode Alice’s input and adaptively continues according to the following cases.

   (a) If Alice’s input cannot be uniquely determined, Bob aborts.

   (b) Otherwise, Bob obtains some input \( x_{i'} \). Bob encodes the string \((x_{i'}, y_j)\) using \( k \)-silence encoding. Specifically, he waits \(((i' - 1)|\mathcal{Y}| + (j - 1))k\) rounds, and then communicates the symbol \( \sigma \) for \( k \) consecutive rounds. We can think about this encoding as if the value \( x \) selects a “block” size \( k|\mathcal{Y}| \) and the \( y \) value selects a region of \( k \) symbols within that block.

   (c) After Bob completes transmitting \( k \) symbols he terminates and outputs \( f(x_{i'}, y_j) \).

3. Alice expects to see exactly \( k \) symbols between round \(|\mathcal{X}| + (i - 1)k|\mathcal{Y}|\) and \(|\mathcal{X}| + ik|\mathcal{Y}|\). If she receives a transmission outside this region, she aborts. By round \(|\mathcal{X}| + ik|\mathcal{Y}| + 1 \) Alice decodes Bob’s codeword and obtains \( y_{j'} \) or an erasure mark, if the codeword cannot be uniquely decoded. Then Alice terminates and outputs \( f(x_i, y_{j'}) \) or \( \perp \), respectively.

Whenever the abort signal is triggered, the other party, if not already terminated, outputs \( \perp \).

Let us now analyze the maximal relative noise rate tolerable by this protocol. Eve can perform either one of the following attacks:

- Eve causes ambiguity in Alice’s message: Bob then aborts at round \(|\mathcal{X}|\). Alice has communicated 1 symbol, and Eve has corrupted at least 1 rounds, thus the noise rate \( \geq 1 \).

- Eve changes Alice’s message \( x_i \) so that Bob decodes an incorrect value \( x_{i'} \). Recall that this costs Eve two corruptions. If \( i' < i \) then Bob sends his symbols prematurely, and Alice aborts after seeing the first symbol. The obtained noise rate is \( \geq 1 \). [If Eve deletes this message, then the noise rate exceeds 1.] If \( i' > i \) then Alice aborts at round \(|\mathcal{X}| + ik|\mathcal{Y}| + 1 \). Eve made
2 corruptions, Alice sent one symbol, and Bob might have sent a single symbol at round $|X| + ik|Y| + 1$, thus the noise rate $\geq 1$.

- Eve injects symbols before round $|X| + (i-1)k|Y|$: Alice aborts upon the first received symbol, and the noise rate $= 1$.
- Eve deletes Bob’s message (i.e., creates ambiguity): this costs her at least $k$ corruptions yielding noise rate $> \frac{k}{k+1}$.

Setting $k$ to be large enough so that $\frac{k}{k+1} > 1 - \varepsilon$ completes this proof. $\square$

**Characterizing functions with $\text{CC}^{\text{adp}}(\bot) = 1$.** If we do not care about round-complexity, then it is clear that as long there is no noise (i.e., the noise pattern is $E = \bot^{2R_{\text{max}}}$, which we simply denote $\bot$ to ease notations), then any function in the $\mathcal{M}_{\text{adp}}$ model has $\text{CC}^{\text{adp}}(\bot) \leq 2$ by the trivial protocol in which each party silence-encodes its input. It is interesting to ask which functions have $\text{CC}^{\text{adp}}(\bot) \leq 1$ in this model, as such functions might naturally resist higher noise rate.

It is clear that functions with $\text{CC}^{\text{adp}}(\bot) = 0$ are constant functions. In Appendix B we give a full characterization of functions with $\text{CC}^{\text{adp}}(\bot) = 1$. We show that this family of functions is not empty and it contains several “non-trivial” functions, such as $\min(x, y)$. Out of this class, any boolean monotonic function can resist error rate up to 1.

### 4 Noise Resilience in the $\mathcal{M}_{\text{term}}$ and $\mathcal{M}_{\text{QwA}}$ Models

In this section, we study the $\mathcal{M}_{\text{term}}$ model, which allows parties to terminate at any round, thus annulling the robustness of the protocol. The $\mathcal{M}_{\text{term}}$ model places no restriction on how the ability to terminate is used, whereas its specialization—the $\mathcal{M}_{\text{QwA}}$ model—restricts the players to only use abort as a “quit-while-ahead” mechanism by forcing the output of the protocol to be $\bot$ if any player terminates early. We proceed to analyze upper and lower bounds on the noise rate in these settings.

#### 4.1 Achievability

First, note that for any constant $\varepsilon > 0$, a non-adaptive protocol that correctly computes $f$ for noise rate less than $1/4 - \varepsilon$ was given by Braverman and Rao [BR11]. Since non-adaptive protocols are a strict subset of adaptive protocols, this yields a protocol that resists noise rate $1/4 - \varepsilon$ in both the $\mathcal{M}_{\text{term}}$ as well as the more restricted $\mathcal{M}_{\text{QwA}}$ model. Moreover, this protocol is symmetric, i.e. both parties communicate a single symbol at each round, and terminate after a fixed number of rounds.

The question is whether we can do better with adaptive termination and asymmetric talk slots. We answer this question in the affirmative. Below, we provide a protocol that resists noise rate $1/3 - \varepsilon$ in the $\mathcal{M}_{\text{term}}$ model.

The main idea for the protocol is as follows. Alice encodes her input using a very redundant encoding. Next, Bob will estimate the noise and transmit a coded version of his input with parameters that correspond to the noise seen so far, so that the more noise he sees, the smaller his codeword will be. Immediately after Bob completes transmitting his codeword, he terminates. Thus, he aborts prematurely in case he estimates high noise.

**Theorem 4.1.** For any function $f$ and any $\varepsilon > 0$, there exist a protocol $\pi$ in the $\mathcal{M}_{\text{term}}$ model, that correctly computes $f$ as long as the relative noise rate is below $1/3 - \varepsilon$. 

16
proof sketch. We assume parties’ inputs are in \( \{0,1\}^n \). We will use a family of good error correcting codes \( \text{ECC}_i : \{0,1\}^n \rightarrow \Sigma^n \) with \( i = 1, \ldots, i_{\text{max}} \). Each such code corrects up to \( 1/2 - \varepsilon \) fraction of errors while having a constant rate \( 1/c_i \) and using a constant alphabet \( \Sigma \), both of which depend on \( \varepsilon \). The redundancy of each code increases with \( i \), i.e., \( c_{i+1} > c_i \). Moreover, these codes will have the property that for any \( x \), \( \text{ECC}_i(x) \) is a prefix of \( \text{ECC}_j(x) \) for any \( j > i \). This can easily be done with random linear codes, e.g., by randomly choosing a large generating matrix of size \( n \times c_{\text{max}}n \) and encoding \( \text{ECC}_i \) by using a truncated matrix using only the first \( c_i \) columns.

Formally, for any \( n \) and \( \varepsilon > 0 \), let \( \{\text{ECC}_i\} \) be a family of error correcting codes as described above and let \( j \) be such that \( c_j \cdot 4\varepsilon > c_1 \). Set \( I_A = \{1,\ldots,c_jn\} \) and \( I_B = \{c_jn+1, c_jn+2,\ldots\} \).

1. Alice encodes her input using \( \text{ECC}_j \), and sends the codeword over to Bob in the first \( c_jn \) rounds of the protocol.

2. After \( c_jn \) rounds, Bob decodes Alice’s transmission to obtain \( \tilde{x} \). Let \( t \) be the Hamming distance between the codeword Bob receives and \( \text{ECC}_j(\tilde{x}) \).

3. Bob continues in an adaptive manner:
   
   (a) if \( t < (1/2 - \varepsilon)c_jn \) Bob encodes his input using a code \( \text{ECC} : \{0,1\}^n \rightarrow \{0,1\}^{2c_jn-4t} \). Note that the maximal value \( t \) can get is \( (1/2 - \varepsilon)c_jn \) which makes \( 2c_jn - 4t > 4\varepsilon c_jn \geq c_1n \), so a suitable code can always be found.

   (b) otherwise, Bob aborts.

4. After completing his transmission, Bob terminates and outputs \( f(\tilde{x},y) \).

5. Alice waits until round \( 3c_jn \) and then decodes Bob’s transmission to obtain \( \tilde{y} \) and outputs \( f(x,\tilde{y}) \).

Let us now analyze the noise rate in a given failed instance of the protocol. First, note that if Bob aborts at step 3b, the noise rate is clearly larger than \( 1/3 \). Next, assume Bob decodes a wrong value \( \tilde{x} \neq x \). Note, that the minimal distance of the code is \( 1 - 2\varepsilon \), thus given that Bob measures Hamming distance \( t \), Eve must have made at least \( (1 - 2\varepsilon)c_jn - t \) corruptions. The total communication in this scenario is \( c_jn + 2(c_jn - 2t) \) which yields a relative noise rate \( \frac{(1-2\varepsilon)c_jn-t}{3c_jn-4t} \), with a minimum of \( 1/3 - O(\varepsilon) \).

On the other hand, if Bob decodes the correct value \( \tilde{x} = x \), and measures Hamming distance \( t \), Eve must have made \( t \) corruptions at Alice’s side. To corrupt Bob’s codeword, she must perform at least \( (1/2 - \varepsilon)(2c_jn - 4t) \) additional corruptions, yielding a relative noise rate at least \( \frac{t+(1/2-\varepsilon)(2c_jn-4t)}{3c_jn-4t} = \frac{(1-2\varepsilon)c_jn+t}{3c_jn-4t} \) which also obtains a minimal value of \( 1/3 - O(\varepsilon) \).

There is still a remaining subtlety of how Alice knows the right code to decode. Surely, if there is no noise, Bob’s transmission is delimited by silence. However, if Eve turns the last couple of symbols transmitted by Bob into silence, she might cause Alice to decode with the wrong parameters. This is where we need the prefix property of the code, which keeps a truncated codeword a valid encoding of Bob’s input for smaller parameters. Eve has no advantage in shortening the codeword: if Eve tries to shorten a codeword of \( \text{ECC}_i \) into \( \text{ECC}_j \) with \( j < i \) and then corrupt the shorter codeword, she will have to corrupt \( (c_i - c_j + (1/2 - \varepsilon)c_j)n \geq (1/2 - \varepsilon)c_i n \) symbols, which only increases her noise rate. Similarly, if she tries to enlarge \( \text{ECC}_i \) into \( \text{ECC}_j \) with \( j > i \), in order to cause Alice to decode the longer codeword incorrectly, Eve will have to perform at least \( (1/2 - \varepsilon)c_j \) corruptions which is again more than needed to corrupt the original message sent by Bob. 

\( \Box \)
4.2 Impossibility

In this section we study upper bounds on the admissible noise in the $\mathcal{M}_{\text{term}}$ and $\mathcal{M}_{\text{QWA}}$ models. First, we show that in the $\mathcal{M}_{\text{QWA}}$ model, for symmetric protocols, we can match the bound of $1/4 - \varepsilon$, i.e. we provide an adversarial strategy that always wins with error rate $1/4$. Note that Braverman and Rao [BR11] showed a similar result for non-adaptive protocols. Informally speaking, their proof goes along the following lines: Eve picks the player, say Bob, who speaks for fewer slots, and changes half his messages so that the first half corresponds to input $y$ while the second half corresponds to $y'$. Now, Eve’s noise rate is at most $1/4$, and Alice cannot tell whether Bob’s input is $y$ or $y'$ and cannot output the correct value.

The above strategy does not carry over to the $\mathcal{M}_{\text{QWA}}$ model. Specifically, the above attack is not well defined. Indeed, Eve can inject messages in the first half of the attack, by running Bob’s part of $\pi$ on the input $y$. However, when Eve wishes to switch to $y'$, she now needs to run $\pi$ on input $y'$ given the transcript so far, say $\text{tr}(y)$. It is possible that $\pi(\cdot, y')$ conditioned on $\text{tr}(y)$ is not defined, for example upon occurrence of $\text{tr}(y)$ given input $y$, Bob may have already terminated and Eve cannot conduct the second part of the attack.

We address this issue by demonstrating a more sophisticated attack that does not abruptly switch $y$ to $y'$ after half the messages, but rather gradually moves from $y$ towards $y'$. Quite surprisingly⁷, the adversarial strategy that achieves noise rate $1/4$ is “rolling budget”, that is, at any time during the protocol the adversary’s relative noise rate is at most $1/4$. Therefore, the fact that the parties can prematurely terminate doesn’t give them any power in the $\mathcal{M}_{\text{QWA}}$ model.⁸

**Theorem 4.2.** There exists a function $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^{2n}$ such that for any adaptive protocol $\pi$ for $f$ in the symmetric $\mathcal{M}_{\text{QWA}}$ model, there exists an adversarial strategy that induces noise rate $< \frac{1}{4}$, and causes the protocol to be incorrect.

Before we prove the theorem we show the following technical lemma, which is the main idea of our proof. Denote the Hamming distance of two strings by $\Delta(\cdot, \cdot)$. In order to cause ambiguity when decoding a codeword from $\{x, y\}$, one needs to corrupt at most $(\Delta(x, y) + 1)/2$ symbols, and this can be done in a “rolling” manner. Formally,

**Lemma 4.3.** Assume $\mathbb{F}$ is some finite field. For any two strings $x, y \in \mathbb{F}^n$ there exists a string $z \in \mathbb{F}^n$ such that

$$\Delta(x + z, x) \geq \Delta(x + z, y)$$

and for any $j \leq n$, $w(z_1, \ldots, z_j) \leq \frac{j+1}{2}$, where $w(\cdot)$ is the Hamming weight function.

**Proof.** We prove by induction on the Hamming distance of the two strings, $d = \Delta(x, y)$. We begin by proving that for an even $d$, a more restricted form of the lemma holds, namely, that for any $j \leq n$, $w(z_1, \ldots, z_j) \leq \frac{j}{2}$. The case of $d = 2$ is easily obtained by setting $z$ to be all zero except for the second index where $x$ and $y$ differ. Now assume the hypothesis holds for an even $d$ and consider $d + 2$. Split $x = x_1 x_2$ and $y = y_1 y_2$ such that $|x_1| = |y_1|$ and $\Delta(x_1, y_1) = d$ (thus $\Delta(x_2, y_2) = 2$). Let $u, v$ be the strings guaranteed by the induction hypothesis for $x_1, y_1$ and $x_2, y_2$ respectively, and set $z = uv$.

Clearly, by $z$’s construction, it holds $\Delta(x + z, x) \geq \Delta(x + z, y)$. Moreover, for any $j < |x_1|$ we know that $w(z_1, \ldots, z_j) \leq \frac{j}{2}$, by the induction hypothesis. Note that $w(v)$ is at most 1, and that

---

⁷Compare with a rolling budget adversary in the $\mathcal{M}_{\text{adv}}$ model (Section 3.4).

⁸In fact, the bound we obtain is $1/4 - O(1/k)$ where $k$ is the round complexity of $\pi$. Therefore, the only hope to obtain protocols that resist any noise rate strictly less than $1/4$ is having infinite protocols. This is however beyond the scope of this work, and is left as an open question.

18
\(v_1 = 0\) by the construction of the base case. Then it is clear that the claim holds for \(j = |x_1| + 1\), and for any \(j > |x_1| + 1\), \(w(z_1, \ldots, z_j) = w(u) + w(v_1, \ldots, v_{j-|x_1|+1}) \leq \frac{|x_1|+1}{2} + 1 \leq \frac{j}{2} + 1\).

Completing the proof of the original lemma (where \(d\) can be odd and the weight is \(\leq \frac{j+1}{2}\)) is immediate. If \(d\) is odd we construct \(z\) by using the induction lemma (of the even case) over the prefix with hamming distance \(d-1\) and change at most a single additional index, which is located after that prefix. Assume that the prefix is of length \(n_{\text{prefix}}\). The claim holds for any \(j \leq n_{\text{prefix}}\) due to the induction hypothesis. For any \(j > n_{\text{prefix}}\) it holds that

\[
w(z_1, \ldots, z_j) = w(z_1, \ldots, z_{n_{\text{prefix}}}) + w(z_{n_{\text{prefix}}+1}, \ldots, z_j) \leq \frac{n_{\text{prefix}}}{2} + 1 \leq \frac{j+1}{2}.
\]

\[\square\]

We now continue to proving that 1/4 is an upper bound of the permissible noise rate.

**Proof. (Theorem 4.2.)** Let \(f\) be such that for any \(y, y', f(x, y) \neq f(x, y')\), for instance, the identity function \(f(x, y) = (x, y)\). Consider the transcripts of \(\pi\) up to round 10.\(^9\) By the pigeonhole principle for large enough \(n\), there must be \(y, y'\) that for some \(x\) produce the same transcript up to round 10. Let \(m = \min\{\text{TER}_A(x, y), \text{TER}_A(x, y')\}\).

The basic idea is the following. Assuming no noise, let \(t\) be Bob’s messages in \(\pi(x, y)\) up to round \(m\) and \(t'\) be Bob’s messages in \(\pi(x, y')\) up to round \(m\). Using Lemma 4.3 Eve can change \(t\) into \(t + z\) (starting from round 10), so that \(\Delta(t + z, t) \geq \Delta(t + z, t')\) and Eve’s relative noise rate never exceeds 1/4. Furthermore, Eve can change \(t'\) into \(t' + z' = t + z\) and also in this case Eve’s relative noise rate never exceeds 1/4: the string \(z_{11}, \ldots, z_m\) must satisfy, for any index \(10 < j \leq m\), that \(w(z_{11}, \ldots, z_j) \leq \frac{j+1}{2}\) (this follows from the way we construct \(z\) and the fact that \(\Delta(t' + z', t) = \Delta(t + z, t) \geq \Delta(t + z, t') = \Delta(t' + z', t')\)). Thus, the relative noise rate made by Eve up to round \(j\) is at most \(\frac{j-10+1}{2} < 1/4\). The same argument should be repeated until we reach the bound on the round complexity \(\text{TER}_{\pi}\), which we formally prove in Lemma 4.4, yet before getting to that we should more carefully examine the actions of both parties during this attack.

Consider Alice actions when the messages she receives are \(t + z = t' + z'\). She can either (i) abort (output \(\perp\)), (ii) output \(f(x, y)\) or (iii) output \(f(x, y')\), however, her actions are independent of Bob’s input (since her view is independent of Bob’s input). Assuming Eve indeed never goes beyond 1/4, it is clear that Eve always wins in case (i). For case (ii) Eve wins on input \((x, y')\) and for case (iii), Eve wins on input \((x, y)\).

However, while in the above analysis Alice’s actions must be the same between the two cases of \(t \rightarrow t + z\) and \(t' \rightarrow t' + z'\), this is not the case for Bob. We must be more careful and consider Bob’s possible adaptive reaction to errors made by Eve. In other words, Bob, noticing Alice replies, may either abort, or send totally different messages so that his transcript is neither \(t\) nor \(t'\). We now show that even in this case Eve has a way to construct \(z, z'\) and never exceed a relative noise rate of 1/4.

**Lemma 4.4.** Assume \(\pi\) takes \(k\) rounds. Eve always has a way to change (only) messages sent by Bob, so that Alice’s view is identical between an instance of \(\pi(x, y)\) and of \(\pi(x, y')\), while Eve corrupts no message up to round 10 and at most \((k-9)/2\) messages between rounds 11 and \(k\) (incl.).

\(^9\)10 is obviously arbitrary, and has the sole purpose of avoiding the edge case in which Eve corrupts the first couple of rounds, possibly causing (a relative) noise rate higher than 1/4.
Proof. We prove by induction. The base case where \( k \leq 10 \) is trivial.

Assume the lemma holds for some even \( k \), and we prove for \( k + 1 \) and \( k + 2 \). By the induction hypothesis, Eve can cause the run of \( \pi(x, y) \) and \( \pi(x, y') \) look identical in Alice’s eyes while corrupting at most \( (k - 9)/2 \) messages after round 10.

Let \( t \) denote the next two messages (rounds \( k + 1, k + 2 \)) sent by Bob in the instance of \( \pi(x, y) \) and \( t' \) in the instance of \( \pi(x, y') \). There are strings \( z, z' \) such that \( w(z), w(z') \leq 1 \) and \( t + z = t' + z' \).

Assume we construct \( z \) via the the construction of Lemma 4.3 then also \( z_1 = 0 \) and \( z'_2 = 0 \). Also note that \( t, t' \) are independent of errors made in rounds \( k + 1, k + 2 \) (Bob ‘sees’ that his message at \( k + 1 \) have been changed at round \( k + 2 \) the earliest, thus this information can affect only his messages at rounds > \( k + 2 \)).

At round \( k + 1 \) the amount of corrupt messages (in both cases) is at most

\[
\left\lfloor \frac{k - 9}{2} \right\rfloor + 1 = \begin{cases} \frac{k - 10}{2} + 1 & k \text{ is even} \\ \frac{(k + 1) - 9}{2} & k \text{ is odd} \end{cases}
\]

And the same holds for round \( k + 2 \) (for both cases).

With the above lemma, Eve can always cause Alice to be confused between an instance of \( \pi(x, y) \) and \( \pi(x, y') \) by inducing, at any point of the protocol, a relative noise rate of at most

\[
\frac{k - 9}{2} < \frac{1}{4}
\]

Therefore, unless one of the parties aborts\(^{11}\), Alice outputs a wrong output. In all these cases the protocol is incorrect while the noise rate is below 1/4.

Attempts to extend the above proof to work for the symmetric \( \mathcal{M}_\text{term} \) model run into a hurdle created by parties’ ability to communicate information about their inputs by the time of aborting. Indeed, in the above attack Alice learns Bob’s inputs (since they were never corrupted), and Bob might be able to distinguish \( x \) from \( x' \) by whether or not Alice has prematurely aborted (i.e., according to the number of silence symbol implicitly communicated by the channel after Alice terminates).

Next, we show that in the general \( \mathcal{M}_\text{term} \) case, no interactive protocol resist a noise rate of 1/2 or more. At a high level, the attack proceeds by changing 1/2 of both Alice and Bob’s messages so that whoever terminates first is completely confused about their partner’s input.

Theorem 4.5. There exists a function \( f \), such that any adaptive protocol \( \pi \) for \( f \) in the \( \mathcal{M}_\text{term} \) setting, cannot resist noise rate of 1/2.

Proof. Assume \( f \) is the identity function on input space \( \{0, 1\}^n \times \{0, 1\}^n \), and consider a resilient protocol \( \pi \). Fix two distinct inputs \( (x, y) \) and \( (x', y') \). Given any \( \xi \in \{(x, y), (x', y), (x, y'), (x', y')\} \) we can define an attack on \( \pi(\xi) \). The attack will change both parties’ transmissions in the following way: Alice’s messages will be changed to the “middle point” between what she should send given that her input is \( x \) and what she should send given that her input is \( x' \). At the same time, Bob’s messages are changed to the middle point between what he should send given that his input is \( y \) and what he should send given that his input is \( y' \).

\(^{10}\)Note that \( t, t' \) are conditioned on the noise Eve has introduced throughout round \( k \).

\(^{11}\)As before, Eve needs not corrupt any message after one of the parties aborts, since she is always within her budget.
Specifically, at each time step, Eve considers the next transmission of Alice on \(x\) and on \(x'\) (given the transcript so far). If Alice sends the same symbol in both cases, Eve doesn’t do anything. Otherwise, Eve alternates between sending a symbol from Alice’s transcript on \(x\) and on \(x'\). Note that the attack is well defined even if Alice has already terminated on input \(x\) but not on \(x'\),\(^{12}\) although we only use the attack until Alice aborts on one of the inputs. Corrupting Bob’s transmissions is done in a similar way.

Next, consider the input \(\xi^* \in \{(x, y), (x', y), (x, y'), (x', y')\}\) for which, given the above attack, the termination time of a party (wlog Alice) is minimal. I.e., when employing the above attack on any of the other three inputs, the termination time of the parties are not smaller than Alice’s termination time in the instance \(\pi(\xi^*)\) under the same attack. Without loss of generality, assume \(\xi^* = (x, y)\).

Finally note that when Alice terminates, she cannot tell whether Bob holds \(y\) or \(y'\). Indeed, up to the point she terminates, the attack on \(\xi^* = (x, y)\) and the attack on \((x, y')\) look exactly the same from Alice’s point of view. This is because Bob does not terminate before Alice (for both his inputs!), and our attack changes Bob’s messages in both instances in a similar way. Thus Alice’s view is identical for both Bob’s inputs, and she must be wrong at least on one of them. Note that such an attack causes at most \(1/2\) noise in each direction up to the point where Alice terminates (there’s no need to continue in the attack after that point). Thus, the total corruption rate is at most \(1/2\). \(\square\)

**Acknowledgements**

We would like to thank Hemanta Maji and Klim Efremenko for useful discussions.

Research supported in part from a DARPA/ONR PROCEED award, NSF grants 1228984, 1136174, 1118096, and 1065276, a Xerox Faculty Research Award, a Google Faculty Research Award, an equipment grant from Intel, and an Okawa Foundation Research Grant. This material is based upon work supported by the Defense Advanced Research Projects Agency through the U.S. Office of Naval Research under Contract N00014-11-1-0389. The views expressed are those of the author and do not reflect the official policy or position of the Department of Defense, the National Science Foundation, or the U.S. Government.

**References**

[BE14] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. Electronic Colloquium on Computational Complexity (ECCC), 2014.

[BK12] Zvika Brakerski and Yael Tauman Kalai. Efficient interactive coding against adversarial noise. *Foundations of Computer Science (FOCS), IEEE Annual Symposium on*, pages 160–166, 2012.

[BN13] Zvika Brakerski and Moni Naor. Fast algorithms for interactive coding. In *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA ’13, pages 443–456, 2013.

\(^{12}\)Recall that once Alice terminates, we assume the symbol \(\emptyset\) is being transmitted by the channel.
whose encoding minimizes the Hamming distance to the received word, namely, $\Delta(x, y) = \Delta(x, 0^n)$. Unless otherwise written, $\log()$ denotes the binary logarithm (base 2).

A $d$-ary tree-code [Sch96] over alphabet $\Sigma$ is a rooted $d$-regular tree of arbitrary depth $N$ whose edges are labeled with elements of $\Sigma$. For any string $x \in [d]^{\leq N}$, a $d$-ary tree-code $T$ implies an encoding of $x$, $TCenc(x) = w_1w_2\ldots w_{|x|}$ with $w_i \in \Sigma$, defined by concatenating the labels along the path defined by $x$, i.e., the path that begins at the root and whose $i$-th node is the $x_i$-th child of the $(i-1)$-st node.

For any two paths (strings) $x, y \in [d]^{\leq N}$ of the same length $n$, let $\ell$ be the longest common prefix of both $x$ and $y$. Denote by $\text{anc}(x, y) = n - |\ell|$ the distance from the $n$-th level to the least common ancestor of paths $x$ and $y$. A tree code has distance $\alpha$ if for any $k \in [N]$ and any distinct $x, y \in [d]^k$, the Hamming distance of $T\text{Cenc}(x)$ and $T\text{Cenc}(y)$ is at least $\alpha \cdot \text{anc}(x, y)$.

For a string $w \in \Sigma^n$, decoding $w$ using the tree code $T$ means returning the string $x \in [d]^n$ whose encoding minimizes the Hamming distance to the received word, namely,

$$T\text{Cdec}(w) = \arg\min_{x \in [d]^n} \Delta(T\text{Cenc}(x), w).$$

A theorem by Schulman [Sch96] proves that for any $d$ and $\alpha < 1$ there exists a $d$-ary tree code of unbounded depth and distance $\alpha$ over alphabet of size $d^{O(1/(1-\alpha))}$. However, no efficient construction of such a tree is yet known. Schulman argued the existence of tree codes but was unable to provide an efficient construction. For a given depth $N$, Peczarski [Pec06] gives a randomized construction for a tree code with $\alpha = 1/2$ that succeeds with probability at least $1 - \epsilon$, and requires alphabet of size at least $d^{O(\sqrt{\log \epsilon^{-1}})}$. Braverman [Bra12] gives a sub-exponential (in $N$) construction of a tree-code, and Gelles, Moitra and Sahai [GMS11] provide an efficient construction of a randomized relaxation of a tree-code of depth $N$, namely a potent tree code, which is powerful enough as a substitute for a tree code in most applications. Finally, Moore and Schulman [MS14] suggested an efficient construction which is based on a conjecture on some exponential sums.

Appendix

A Constant-Rate Protocol With Resiliency $1/2 - \varepsilon$ in the $M_{\text{adp}}$ Model

In this section we prove Theorem 3.3. First, let us recall some primitives and notations that will be used in our proof. We denote the set $\{1, 2, \ldots, n\}$ by $[n]$, and for a finite set $\Sigma$ we denote by $\Sigma^{\leq n}$ the set $\bigcup_{k=1}^{n} \Sigma^k$. The Hamming distance $\Delta(x, y)$ of two strings $x, y \in \Sigma^n$ is the number of indices $i$ for which $x_i \neq y_i$, and the Hamming weight of some string, is its distance from the all-zero string, $w(x) = \Delta(x, 0^n)$. Unless otherwise written, $\log()$ denotes the binary logarithm (base 2).

A $d$-ary tree-code [Sch96] over alphabet $\Sigma$ is a rooted $d$-regular tree of arbitrary depth $N$ whose edges are labeled with elements of $\Sigma$. For any string $x \in [d]^{\leq N}$, a $d$-ary tree-code $T$ implies an encoding of $x$, $TCenc(x) = w_1w_2\ldots w_{|x|}$ with $w_i \in \Sigma$, defined by concatenating the labels along the path defined by $x$, i.e., the path that begins at the root and whose $i$-th node is the $x_i$-th child of the $(i-1)$-st node.

For any two paths (strings) $x, y \in [d]^{\leq N}$ of the same length $n$, let $\ell$ be the longest common prefix of both $x$ and $y$. Denote by $\text{anc}(x, y) = n - |\ell|$ the distance from the $n$-th level to the least common ancestor of paths $x$ and $y$. A tree code has distance $\alpha$ if for any $k \in [N]$ and any distinct $x, y \in [d]^k$, the Hamming distance of $T\text{Cenc}(x)$ and $T\text{Cenc}(y)$ is at least $\alpha \cdot \text{anc}(x, y)$.

For a string $w \in \Sigma^n$, decoding $w$ using the tree code $T$ means returning the string $x \in [d]^n$ whose encoding minimizes the Hamming distance to the received word, namely,

$$T\text{Cdec}(w) = \arg\min_{x \in [d]^n} \Delta(T\text{Cenc}(x), w).$$

A theorem by Schulman [Sch96] proves that for any $d$ and $\alpha < 1$ there exists a $d$-ary tree code of unbounded depth and distance $\alpha$ over alphabet of size $d^{O(1/(1-\alpha))}$. However, no efficient construction of such a tree is yet known. Schulman argued the existence of tree codes but was unable to provide an efficient construction. For a given depth $N$, Peczarski [Pec06] gives a randomized construction for a tree code with $\alpha = 1/2$ that succeeds with probability at least $1 - \epsilon$, and requires alphabet of size at least $d^{O(\sqrt{\log \epsilon^{-1}})}$. Braverman [Bra12] gives a sub-exponential (in $N$) construction of a tree-code, and Gelles, Moitra and Sahai [GMS11] provide an efficient construction of a randomized relaxation of a tree-code of depth $N$, namely a potent tree code, which is powerful enough as a substitute for a tree code in most applications. Finally, Moore and Schulman [MS14] suggested an efficient construction which is based on a conjecture on some exponential sums.
We now prove Theorem 3.3. For any function \( f \) and any constant \( \varepsilon > 0 \), we construct a protocol that correctly computes \( f \) as long as the relative noise rate does not exceeds \( 1/2 - \varepsilon \).

Let \( \varepsilon > 0 \) be fixed, and let \( \pi \) be an interactive protocol in the noiseless model for \( f \), in which the parties exchange bits with each other for up to \( T \) rounds. We begin by turning \( \pi \) into a resilient version \( \pi_{BR} \) which resists noise rate of up to \( 1/4 - \varepsilon \), using techniques from [BR11]. The protocol takes \( N = O(T) \) rounds in each of which both parties send a message over some finite alphabet \( \Sigma \).

**Lemma A.1** ([BR11]). For every \( \varepsilon \) there is an alphabet \( \Sigma \) of size \( O(1/\varepsilon) \) such that any binary protocol \( \pi \) can be compiled to a protocol \( \pi_{BR} \) of \( N = O(|\pi|) \) rounds in each of which both parties send a symbol from \( \Sigma \). For any input \( x, y \), both parties output \( \pi(x, y) \) if the fraction of errors is at most \( 1/4 - \varepsilon \).

The conversion is described in [BR11]; we give more details about this construction in the proof of Lemma A.3.

Next, we construct a protocol \( \Pi \) that withstands noise rate of \( 1/2 - \varepsilon \). The parties run \( \pi_{BR} \), yet each symbol from \( \Sigma \) is silence-encoded. That is, every round of \( \pi_{BR} \) in which a party sends some symbol \( a \in \Sigma \) is expanded into \(|\Sigma|\) rounds of \( \Pi \) in which a single symbol ‘\( \sigma \)’ is sent at a timing that corresponds to the index of \( a \) in the total ordering of \( \Sigma \). The channel alphabet used in \( \Pi \) is thus unary. Decoding is performed by minimizing Hamming distance and the decoder obtains either a symbol of \( \Sigma \) or an erasure mark \( \perp \).

From this point and on, we regard only rounds of \( \pi_{BR} \) protocol, ignoring the fact that each such ‘round’ is composed of \(|\Sigma|\) mini-rounds. Denote by \( N(i, j) \) the ‘effective’ noise-rate between rounds \( i \) and \( j \), for which an erasure is counted as a single error and decoding the wrong symbol of \( \Sigma \) is counted as two errors. Formally, assume that at time \( n \), Alice sends a symbol \( a_n \in \Sigma \), and Bob receives \( \tilde{a}_n \in \Sigma \cup \{\perp\} \), possibly with added noise or an erasure mark (similarly, Bob sends \( b_n \in \Gamma \), and Alice receives \( b_n \)).

**Definition A.2.** Let the effective noise in Alice’s transmissions be

\[
N_A(i, j) = |\{ k \mid i \leq k \leq j, \tilde{a}_k = \perp \}| + 2|\{ k \mid i \leq k \leq j, \tilde{a}_k \notin \{a_k, \perp\}\}|
\]

and similarly define \( N_B(i, j) \) for the effective noise in Bob’s transmissions. The effective number of corruptions in the interval \([i, j]\) is \( N(i, j) = N_A(i, j) + N_B(i, j) \).

The following lemma states that if the \( \pi_{BR} \) fails, then \( N \) must be high.

**Lemma A.3** ([FGOS13]). Let \( \varepsilon > 0 \) be fixed and let \(|\pi_{BR}| = N\). If \( \pi_{BR} \) fails, then

\[
N(1, N) \geq (1 - \varepsilon)^2 N
\]

With this lemma, the proof of the theorem is immediate: recall that with silence encoding, causing an erasure costs at least one corruption and causing an error costs at least two corruptions. Observe that \( CC_{\Pi} = CC_{\pi_{BR}} = 2N \), then if the amount of corruptions is limited to \( 1/2 - \varepsilon \),

\[
\max N(1, N) = (1 - 2\varepsilon)N < (1 - \varepsilon)^2 N
\]

where the maximum is over all possible noise-patterns of at most \((1/2 - \varepsilon) \cdot 2N\) corruptions.

Finally, we give the proof for Lemma A.3. Parts of this analysis were taken as-is from [FGOS13] and we re-iterate them here (with the authors’ kind permission) for self containment.
Proof. (Lemma A.3.) Let us recall how to construct a constant-rate protocol \( \pi_{BR} \) for computing \( f(x, y) \) over a noisy channel out of an interactive protocol \( \pi \) for the same task that assumes a noiseless channel [BR11]. We assume that \( \pi \) consists of \( T \) rounds in which Alice and Bob send a single bit according to their input and previous transmissions. Without loss on generality, we assume that Alice sends her bits at odd rounds while Bob transmits at even rounds. We can view the computation of \( \pi \) as a root-leaf walk along a binary tree in which odd levels correspond to Alice's messages and even levels to Bob's, see Figure 1.

![Figure 1: A \( \pi \)-tree showing the path \( P \) (bold edges) taken by Alice and Bob for computing \( f(x, y) \). Dashed edges represent the hypothetical reply of Alice and Bob given that a different path \( P' \) was taken (when such replies are defined).](image)

In order to obtain a protocol that withstands (a low rate of) channel noise, Alice and Bob simulate the construction of path \( P \) along the \( \pi \)-tree. The users transmit edges of \( P \) one by one, where each user transmits the next edge that extends the partial path transmitted so far. This process is repeated for \( N = O(\varepsilon T) \) times. In [BR11] it is shown that unless the noise rate exceeds \( 1/4 \), after \( N \) rounds both parties will decode the entire path \( P \). We refer the reader to [BR11] for a full description of the protocol and correctness proof. We now extend the analysis for the case of channels with errors and erasures.

To simplify the explanation, assume that the players wish to exchange, at each round, a transmission over \( \Gamma' = \{0, \ldots, N\} \times \{0, 1\} \leq 2 \). Intuitively, the transmission \((e, s) \in \Gamma'\) means “extend the path \( P \) by taking at most two steps defined by \( s \) starting at the child of the edge I have transmitted at transmission number \( e \)”.

Since \( \Gamma \) is not of constant size, the symbol \((e, s)\) is not communicated directly over the channel, but is encoded in the following manner. Let \( \Gamma = \{<, 0, 1, >\} \) and encode each \((e, s)\) into a string \( < z > \in \Gamma^{\leq \log N + 2} \) where \( z \) is the binary representation of \((e, s)\). Furthermore, assume that \(|< z >| \leq c \log(e)\) for some constant \( c \) we can pick later. Next, each symbol of \( \Gamma \) is encoded via a \(|\Gamma|-ary tree-code with distance parameter 1 - \varepsilon\) and label alphabet \( \Sigma = O_{\varepsilon}(|\Gamma|)\).

At time \( n \) Alice sends \( a_n \in \Sigma \), the last symbol of \( TCenc((e, s)_1, \ldots, (e, s)_n) = a_1a_2 \cdots a_n \), and Bob receives \( \tilde{a}_n \in \Gamma \cup \{\perp\} \), possibly with added noise or an erasure mark (similarly, Bob sends \( b_n \in \Sigma \), and Alice receives \( \tilde{b}_n \)). Let \( TCdec(\tilde{a}_1, \ldots, \tilde{a}_n) \) denote the string Bob decodes at time \( n \) (similarly, Alice decodes \( TCdec(\tilde{b}_1, \ldots, \tilde{b}_n) \)). For every \( i > 0 \), we denote with \( m(i) \) the largest number such that the first \( m(i) \) symbols of \( TCdec(\tilde{a}_1, \ldots, \tilde{a}_i) \) equal to \( a_1, \ldots, a_{m(i)} \) and the first \( m(i) \) symbols of \( TCdec(\tilde{b}_1, \ldots, \tilde{b}_i) \) equal to \( b_1, \ldots, b_{m(i)} \).

---

13 On top of the tree-code encoding, we implicitly perform silence encoding of every symbol in \( \Sigma \).
Let $\mathcal{N}$ be as defined in Definition A.2. We begin by showing that if $m(i) < i$ then many corruptions must have happened in the interval $[m(i) + 1, i]$.

**Lemma A.4.** $\mathcal{N}(m(i) + 1, i) \geq (1 - \varepsilon)(i - m(i))$.

*Proof.* Assume that at time $i$ Bob decodes the string $a'_1, \ldots, a'_i$. By the definition of $m(i)$, $a'_1, \ldots, a'_{m(i)} = a_1, \ldots, a_{m(i)}$, and assume without loss of generality that $a'_{m(i)+1} \neq a_{m(i)+1}$. Note that the Hamming distance between $\text{Tcenc}(a_1, \ldots, a_i)$ and $\text{Tcenc}(a'_1, \ldots, a'_i)$ must be at least $(1 - \varepsilon)(i - m(i))$. It is immediate that for Bob to make such a decoding error, $\mathcal{N}_A \geq (1 - \varepsilon)(i - m(i))$.

Next, we demonstrate that if some party didn’t announce the $k$-th edge by round $i + 1$, it must be the case that the $(k - 1)$-th edge wasn’t correctly decoded early enough to allow completing the transmission of the $k$-th edge.

**Lemma A.5.** Let $t(i)$ be the earliest time such that both users announced the first $i$ edges of $P$ within their transmissions. For $i \geq 0$, $k \geq 1$, if $t(k(i)) > i + 1$, then either $t(k(i)) > i - c\log(i - (t(k(i))))$, or there exists $j$ such that $t(k(i)) > m(j)$ and $i - c\log(i - t(k(i))) < j \leq i$.

*Proof.* [The proof is taken from [BR11], as this claim is independent of the definition of $\mathcal{N}$.] Without loss of generality, assume that the $k$-th edge of $P$ describes Alice’s move. Suppose that for any $j$ that satisfies $i - c\log(i - t(k(i))) < j \leq i$ both $t(k(i)) \leq m(j)$ and $t(k(i)) \leq i - c\log(i - t(k(i)))$. Then it must be the case that the first $k - 1$ edges of $P$ have already been announced, and correctly decoded by Alice for any $j$ in the last $c\log(i - t(k(i)))$ rounds, yet the $k$th edge has not. However, by the protocol definition, Alice should announce this edge, and this takes her at most $c\log(i - t(k(i)))$ rounds, thus by round $i + 1$ she has completed announcing it, in contradiction to our assumption that $t(k) > i + 1$.

Finally, we relate the effective noise rate, with the protocol progress of the protocol.

**Lemma A.6.** For $i \geq -1, k \geq 0$, if $t(k) > i + 1$, then there exist numbers $\ell_1, \ldots, \ell_k \geq 0$ such that $\sum_{s=1}^{k} \ell_s \leq i + 1$ and $\mathcal{N}(1, i) \geq (1 - \varepsilon)(i - k + 1 - \sum_{s=1}^{k} c\log(\ell_s + 2))$.

*Proof.* We prove by induction. The claim trivially holds for $k = 1$ and for $i \leq 0$ by choosing $\ell_s = 0$. Otherwise, by Lemma A.5 there are two cases. The first case is that $t(k - 1) > i - c\log(i - t(k - 1))$. Let $i' = t(k - 1) - 1$ and $k' = k - 1$, thus by the induction hypothesis (on $i', k'$), there exist $\ell_1, \ldots, \ell_{k-1} \geq 0$ with $\sum_{s=1}^{k-1} c\log(\ell_s) \leq t(k + 1)$ such that

$$\mathcal{N}(1, i) \geq \mathcal{N}(1', i) \geq (1 - \varepsilon) \left( (t(k - 1) - 1) - (k - 1) + 1 - \sum_{s=1}^{k-1} c\log(\ell_s + 2) \right) \right.$$

$$= (1 - \varepsilon) \left( i - k + 1 - \sum_{s=1}^{k-1} c\log(\ell_s + 2) - (i - t(k - 1)) \right)$$

Set $\ell_k = i - t(k + 1)$ to complete this case.

In the other case there exists $j$ such that $m(j) < t(k - 1)$ and $i - c\log(i - t(k - 1)) < j \leq i$. In this case we can write

$$\mathcal{N}(1, i) = \mathcal{N}(1, m(j)) + \mathcal{N}(m(j) + 1, i).$$

The second term is lower bounded by $\mathcal{N}(m(j) + 1, j)$, which by Lemma A.4 is lower bounded by $(1 - \varepsilon)(j - m(j))$. We use the induction hypothesis to bound the first term (with $i' = m(j) - 1$ and
\(k' = k - 1\) to get

\[
N(1, m(j)) \geq N(1, m(j) - 1) \geq (1 - \varepsilon) \left( m(j) - 1 - (k - 1) + 1 - \sum_{s=1}^{k-1} c \log(\ell_s + 2) \right)
\]

\[
\geq (1 - \varepsilon) \left( j - k + 1 - \sum_{s=1}^{k-1} c \log(\ell_s + 2) - j + m(j) \right)
\]

for \(\ell_1, \ldots, \ell_{k-1} \geq 0\) such that \(\sum_{s=1}^{k-1} \ell_s < m(j)\). Take \(\ell_k = i - t(k - 1)\). Since \(t(k - 1) \geq m(j)\) we get that \(\sum_{s=1}^{k} \ell_s < m_j + (i - m(j)) < i + 1\) and

\[
N(1, i) \geq N(1, m(j)) + N(m(j) + 1, i)
\]

\[
\geq (1 - \varepsilon) \left( j - k + 1 - \sum_{s=1}^{k-1} c \log(\ell_s + 2) \right)
\]

\[
\geq (1 - \varepsilon) \left( i - k + 1 - \sum_{s=1}^{k-1} c \log(\ell_s + 2) - i + j \right)
\]

Which completes the proof since for this case \(i - j < c \log(i - t(k - 1)) = c \log(\ell_k)\).

We can now complete the proof of Lemma A.3. Suppose the protocol \(\pi_{BR}\) has failed, thus \(m(N) < t(T)\). By Lemma A.6 we have \(\ell_1, \ldots, \ell_T \geq 0\) that satisfy \(\sum_{s=1}^{T} \ell_s < m(N) < N\) and

\[
N(1, N) \geq N(1, m(N) - 1) + N(m(N) + 1, N)
\]

\[
\geq (1 - \varepsilon)(m(N) - T - \sum_{s=1}^{T} c \log(\ell_s + 2)) + (1 - \varepsilon)(N - m(N))
\]

\[
\geq (1 - \varepsilon) \left( N - T - cT \log \left( \frac{1}{T} \sum_{s=1}^{T} (\ell_s + 2) \right) \right)
\]

\[
\geq (1 - \varepsilon) \left( N - T - cT \log \left( \frac{m(N) + 2}{T} \right) \right)
\]

where the second transition is due to the concavity of the log function. Setting, for instance, \(N = T \frac{\varepsilon^2}{2} \log(e^{-1})\) gives

\[
N(1, N) \geq (1 - \varepsilon) \left( N - \varepsilon N/e^2 \log(e^{-1}) - \varepsilon N \log(3N/T)/c \log(e^{-1}) \right)
\]

\[
= (1 - \varepsilon) \left( N - \varepsilon N/e^2 \log(e^{-1}) - \varepsilon \log(3e^2 \varepsilon^{-1} \log(e^{-1}))/c \log(e^{-1}) \right)
\]

\[
= (1 - \varepsilon) \left( 1 - \varepsilon \frac{1 + \log(e^{-1}) + c \log(3e^2 \log(e^{-1}))}{c^2 \log(e^{-1})} \right) N
\]

\[
> (1 - \varepsilon)^2 N,
\]

for a large enough constant \(c\).
B Characterization of Functions With $CC^{\text{adp}}(\bot) = 1$

In this section we extend the discussion started by [DFO10] and [IW10] on the communication complexity of specific functions in the noiseless case, when the parties are allowed a special “silence” symbol which does not count towards the communication complexity. Specifically, we assume that a party either transmit a pulse (which increases the communication complexity) or remains silent (which doesn’t affect the communication complexity). We ask what functions have $CC^{\text{adp}}(\bot) = 1$, i.e., send only a single pulse assuming there is no noise. We remark that any function trivially has $CC^{\text{adp}}(\bot) \le 2$, and that the class of functions with $CC^{\text{adp}}(\bot) = 0$ is exactly all the constant functions.

**Definition B.1** (partitions). We call $S_1, \ldots, S_m$ a partition of a set $S$, if $\bigcup_i S_i = S$, and $\forall i \neq j$ it holds that $S_i \cap S_j = \emptyset$. A row-partition of a domain $X \times Y$ is the partitions $X_1 \times Y, \ldots, X_m \times Y$ such that $X_1, \ldots, X_m$ is a partition of $X$. Similarly, a column-partition of the domain $X \times Y$ is the partitions $X \times Y_1, \ldots, X \times Y_m$ such that $Y_1, \ldots, Y_m$ is a partition of $Y$.

For simplicity we consider only binary partitions, that is $m = 2$. We say that some subdomain $X' \times Y'$ is a restricted domain (of depth $d$) if it takes $d$ recursive decompositions to obtain $X' \times Y'$.

**Definition B.2.** We say that $f : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$ is All-But-One Partitionable (ABO) if:

1. $f$ is constant on $\mathcal{X} \times \mathcal{Y}$; or,
2. There is either a row-partition or a column partition of $\mathcal{X} \times \mathcal{Y}$ such that
   
   (a) $f$ is constant restricted to any of the partitions, except for at most a single partition.
   
   (b) $f$ restricted to any of its partitions, is also All-But-One partitionable.

**Proposition B.3.** The set of functions with $CC^{\text{adp}}(\bot) = 1$ is exactly all the functions that are All-But-One partitionable.

**Proof.** We give a protocol with $CC^{\text{adp}}(\bot) = 1$ for any ABO function. The protocol follows the partitioning of the function. If the current domain is ABO-row-partitionable, then Alice is the active player, and otherwise it is Bob. At every partition level, we order the partitions in some fixed order, where the non-constant partition is the last. If the active player’s input is in the $i$-th constant partition, the party is silent for $(i-1)$ rounds and then at the $i$-th round, it sends some symbol $\sigma$. If the active player’s input is within the last (non-constant) partition, the party remains silent. In that case, the parties continue the protocol by recursively computing $f$ restricted to the non-constant partition in the same manner.

We now show the other direction, namely that all the functions with $CC^{\text{adp}}(\bot) = 1$ are ABO. Assume towards contradiction that a non-ABO function $f$ can be computed with $CC^{\text{adp}}_f(\bot) = 1$. Since $f$ is not ABO, it must be that if we recursively partition it, we get some restricted domain $\mathcal{D} = X' \times Y'$ such that any row-partition of $\mathcal{D}$, say $P, Q$, satisfies that $f$ restricted to $P \times Y'$ is non-constant and neither does $f$ restricted to $Q \times Y'$. Similarly, $f$ is non-constant restricted to any column-partition of $\mathcal{D}$.

Let $R(u, v)$ be the round in which the single symbol is sent when running $\pi$ on $u, v$, and denote by $P(u, v) \in \{\text{Alice, Bob}\}$ the party that sends this symbol. Let $(u, v) := \min_{u', v' \in \mathcal{D}} R(u', v')$, and assume without loss of generality that $P(u, v)$ is Alice. Note that since this $(u, v)$ is the input that minimizes $R(\cdot)$, it must be that $\forall y R(u, y) = R(u, v)$. This is because Alice has no knowledge of Bob’s input, thus she must behave the same for all those cases (specifically, she gives the same output). Therefore, it must be that $\forall y f(u, y) = f(u, v)$, and we reached a contradiction since $X'_1 = \{u\}, X'_2 = X' \setminus \{u\}$ is a row-decomposition of $\mathcal{D}$ in which one of the partitions is constant. 

28
We note that some interesting functions are ABO. For instance, any function that depends only on one input is ABO, but these don’t require interaction and thus are somewhat trivial in the interactive setting. An example for an ABO function that depends on both inputs is the subdomain-indicator function $g : \mathcal{X} \times \mathcal{Y} \to \{0, 1\}$,

$$
\begin{align*}
g_{X' \times Y'}(x, y) &= \begin{cases} 
1 & (x, y) \in X' \times Y' \\
0 & \text{otherwise}
\end{cases},
\end{align*}
$$

parametrized by some subdomain $X' \times Y' \subseteq \mathcal{X} \times \mathcal{Y}$. Another interesting example is $f = \min(x, y)$ defined on $\mathcal{X} = \mathcal{Y} = Z = \{0, 1\}^n$; see Figure 2.

| $f$ | 0  | 1  | 2  | ... | $2^n - 1$ |
|-----|----|----|----|-----|-----------|
| 0   | 0  | 0  | 0  | ... | 0         |
| 1   | 0  | 1  | -  | -   | 1         |
| 2   | 0  | 1  | 2  |     | 2         |
| ... |    |    |    |     |           |
| $2^n - 1$ | 0  | 1  | 2  | ... | $2^n - 1$ |

Figure 2: The function $f = \min(x, y)$ with its ABO partitions.

Discussion. One might think that since these functions have $\text{CC}_{\text{adp}}(\bot) = 1$ they can resist relative noise rate up to one, however this is not necessarily correct. To see this, consider computing $f = \min(x, y)$ via the protocol of Proposition B.3, when Alice holds 0 and Bob holds $y > 0$. Alice would send one symbol at the first round to indicate her input is 0. However, if this symbol is deleted by Eve, Bob will communicate another symbol at the second round (since Bob doesn’t hear anything from Alice at the first round, he assumes $x > 0$). Bob outputs the wrong value and the error rate is $1/2$.

On the other hand, it is easy to see that if the function is boolean and monotone (e.g., the OR function on $n$ bits), then the above protocol indeed resists relative noise rate up to 1.