New Directions in Coding Theory: Capacity and Limitations

Ameya A. Velingker

CMU-CS-16-122

August 2016

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Venkatesan Guruswami, Chair
Bernhard Haeupler
Gary Miller
Emmanuel Abbe, Princeton University

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

This research was sponsored by the National Science Foundation under grant numbers CCF1065106, CCF1115525, CCF0963975 and Microsoft.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of any sponsoring institution, the U.S. government or any other entity.

Keywords: error-correcting codes, Shannon capacity, polar codes, communication channels, interactive communication, affine invariance, locally testable codes (LTCs), list decoding, restricted isometry property (RIP), entropy sumset inequalities

To Aayi, Baba, and Yogeshwar for their encouragement, love, and support.

Abstract

Error-correcting codes were originally developed in the context of reliable delivery of data over a noisy communication channel and continue to be widely used in communication and storage systems. Over time, error-correcting codes have also been shown to have several exciting connections to areas in theoretical computer science. Recently, there have been several advances including new constructions of efficient codes as well as coding in different settings. This thesis explores several new directions in modern coding theory. To this end, we:

1. Provide a theoretical analysis of polar codes, which were a breakthrough made by Arikan in the last decade [Arı09]. We show that polar codes over prime alphabets are the first explicit construction of efficient codes to provably achieve Shannon capacity for symmetric channels with polynomially fast convergence. We introduce interesting new techniques involving entropy sumset inequalities, which are an entropic analogue of sumset inequalities studied in additive combinatorics.
2. Consider the recent problem of coding for two-party interactive communication, in which two parties wish to execute a protocol over a noisy interactive channel. Specifically, we provide an explicit interactive coding scheme for oblivious adversarial errors and bridge the gap between channel capacities for interactive communication and one-way communication.
3. Study the problem of list decodability for codes. We resolve an open question about the list decodability of random linear codes and show surprising connections to the field of compressed sensing, in which we provide improved bounds on the number of frequency samples needed for exact reconstruction of sparse signals (improving upon the work of Candès and Tao [CT06] as well as Rudelson and Vershynin [RV08]).
4. Study locally-testable codes and affine invariance in codes. Specifically, we investigate the limitations posed by local testability, which has served as an important notion in the study of probabilistically checkable proofs (PCPs) and hardness of approximation.

Acknowledgments

First and foremost, I would like to thank my advisors Venkat Guruswami and Gary Miller for their guidance and support throughout my graduate school years at Carnegie Mellon University. Venkat introduced me to various interesting problems in coding theory from the time I first visited Carnegie Mellon as a prospective student, and I am grateful to him for teaching me a lot about error-correcting codes and serving as a wonderful mentor throughout graduate school. Venkat has been very patient with me and has been the source of many fruitful collaborations, many of which have made their way into this thesis. I am also grateful to Gary for providing me a lot of guidance early on in graduate school and taking the time to teach me about computational geometry, which led to some productive discussions and research collaborations. Gary would always make a lot of effort to be available for me and explain new ideas to me, and I feel that interacting with him helped my development as a researcher. This thesis would not have been possible without the excellent mentorship of Venkat and Gary.

I would also like to thank the rest of my thesis committee: Emmanuel Abbe and Bernhard Haeupler. They provided a lot of useful feedback and advice during my work on my thesis dissertation. Additionally, I am grateful to Bernhard for introducing me to the realm of coding for interactive communication, which led to some fruitful research collaborations over my last few years at Carnegie Mellon.

I have had the privilege of co-authoring papers with a number of people during my time in graduate school: Mitali Bafna, Karthekeyan Chandrasekaran, Mahdi Cheraghchi, Michael Cohen, Brittany Fasy, Pritish Kamath, Michael Kapralov, Sanjeev Khanna, Satyanarayana Lokam, Amir Nayyeri, Donald Sheehy, Madhu Sudan, Sébastien Tavenas, Carol Wang. I thank them for the numerous useful discussions and collaborations we had throughout my time as a graduate student.

I am indebted to Microsoft Research India and my mentors, Satyanarayana Lokam and Navin Goyal, for providing me the opportunity of an excellent summer internship. I would also like to thank Microsoft Research New England for hosting me for a semester and allowing me to broaden my horizons. I would specifically like to thank Madhu Sudan for taking the time to explain various topics about coding theory and probabilistically checkable proofs (PCPs) during my semester there. I also thank Jelani Nelson for allowing me to visit Harvard University and enlightening me about dimensionality reduction and subspace embeddings. I am grateful to the Simons Institute for the Theory of Computing and the organizers of their Information Theory program for hosting my visit, during which I attended a number of exciting workshops and got to expose myself to a number of new and exciting research topics. I also thank Swastik Kopparty for hosting my visit to Rutgers University and taking the time to discuss interesting research topics with me.

I would also like to thank colleagues, friends, and faculty at Carnegie Mellon as well as other institutions with whom I have had numerous useful research conversations: Mohammad Bavarian, Arnab Bhattacharyya, Vijay Bhattiprolu, Eshan Chattopadhyay, Alan Guo, Guru Guruganesh, Jakub Pachocki, Richard Peng, Sahil Singla, David Witmer, John Wright, Patrick Xia, Henry Yuen, Goran Žužić, and many others!

I thank Deb Cavlovich and Catherine Copetas, who made everything go smoothly throughout my time at Carnegie Mellon. There were many times when I needed their assistance, and Deb and Catherine always went well beyond what was required to make sure everything was in order.

I would also like to thank many friends at Carnegie Mellon who made the Ph.D. program a truly enjoyable time: Abhishek Bhardwaj, Alejandro Carbonara, Aram Ebtekar, Justin Freedman, Joshua Giltinan, Nika Haghtalab, Siddhartha Jain, Ashique Khudabukhsh, Dimitris Konomis, Kelvin Liu, Shriphani Palakodety, Shayak Sen, Preeti Singh, Dougal Sutherland, Lin Xiao, Manzil Zaheer, Veronica Zheng, and many others!

I would like to extend thanks to Don Davis, Joe Gallian, and Ken Ono, who provided mentorship during my high school and undergraduate years and motivated me to pursue research.

Finally, I would like to thank my family. In particular, I thank my parents, Avinash and Vina Velingker, who nurtured my passion for mathematics and computer science from a young age. Without their active interest, encouragement, and guidance, I would not have made it to Carnegie Mellon or have been able to complete this dissertation. My parents always encouraged me to pursue my dreams and passions. I also thank my younger brother Yogeshwar Velingker, who has been a source of much joy and inspiration over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The subject of this thesis is *coding theory*, or the study of *error-correcting codes*. Historically, error-correcting codes were developed in the context of reliable communication, in which a sender is trying to transmit a message consisting of symbols to a receiver over a noisy channel. While much research continues to focus on the initial motivation for the field, in recent years, there have emerged a number of new exciting connections of error-correcting codes to other areas. We explore several new directions in coding theory in this thesis.

1.1 Error-Correcting Codes

Error-correcting codes allow two parties to communicate reliably in the presence of noise by providing a method to add redundancy to a desired message.

Let us describe some of the basic notions and definitions involving error-correcting codes.

Definition 1. *An* $[n, k]$ error-correcting code *consists of an* encoding function $\mathsf{Enc} : \Sigma^k \to \Sigma'^n$ *and a* decoding function $\mathsf{Dec} : \Sigma'^n \to \Sigma^k$*. Here,* Σ *is the* alphabet *of the message, and* Σ' *is the alphabet of the received word. In most cases we consider, we will have* $\Sigma' = \Sigma$*. If we wish to highlight that* $|\Sigma'| = |\Sigma| = q$*, then we will often refer to the code as an* $[n, k]_q$ *code (with the subscript* q*).*

We now describe several key terms and properties of error-correcting codes:

- **Code**. The *code* generally refers to the image of the encoding function Enc.

- **Codewords**. Given an error-correcting code with encoding function Enc, we refer to each element of the image of Enc as a *codeword*.

- **Communication Rate**. The *communication rate*, or simply *rate*, of an $[n, k]_q$ error-correcting code is given by k/n. In other words, it is the ratio of the length of the message to the length of the encoding. The rate lies betwen 0 and 1 and essentially measures the amount of redundancy introduced by the encoding.

- **Minimum Distance**. The *minimum distance* (often referred to as simply *distance*) refers to the smallest Hamming distance between any two codewords. Thus, it provides a measure of how far apart the codewords are spaced out. If the minimum distance of an $[n, k]_q$ code $\mathcal{C}$ is d, then we will often refer to the code as an $[n, k, d]_q$ code.

1.1.1 Communication Channels and Reliable Communication

The primary use of error-correcting codes has been to enable reliable communication over a noisy communication channel. A mathematical theory of communication was proposed in the seminal work of Shannon [Sha48], which introduced the notion of communication rate and related it to the use of an encoder and decoder in order to add redundancy.

Specifically, Shannon introduced a probabilistic model of the *communication channel*, in which conditional probabilities specify the probability of any input symbol being outputted as a particular symbol. In a remarkable result that resulted in the birth of information theory and coding theory, Shannon showed that for any channel, there exists a certain real number called the *capacity* of the channel such that any communication rate below the capacity is achievable for reliable transmission over the given channel, while any communication rate above the capacity results in non-negligible loss of information.

Shannon's result was, however, *existential*—it only showed the existence of good coding schemes achieving any specified rate below the channel capacity. In particular, it did not show how to construct explicit error-correcting codes (with efficient encoding and decoding functions) that achieve the desired rate. Thus, one of the central challenges in coding theory over the past several decades following Shannon's work has been to find explicit constructions of codes that perform well.

1.1.2 Adversarial Errors and Minimum Distance

While Shannon's work viewed communication from a probabilistic viewpoint, there is another perspective of reliable communication that views error-correcting codes as a combinatorial and geometric object. The celebrated work of Hamming [Ham50] adopted this latter viewpoint and laid many of the foundations of error-correcting codes. The viewpoint is most natural in the context of reliable communication under *adversarial errors*.

Recall that Shannon's model of a communication channel treats errors as probabilistic (as determined by the underlying conditional probabilities) and requires that a coding scheme reliably communicate a message with high proability (over the randomness of the channel).

On the other hand, one can consider the case of adversarial errors, in which one wishes to have a reliable coding scheme that tolerates any error pattern of up to a certain number of errors. Intuitively, one wishes to use an error-correcting code with the property that pairs of codewords are *far* from each other. Then, if a few symbols in a transmitted codeword are distorted, the resulting string will still be closer in structure to the original codeword than any other codeword, meaning that the decoder will not confuse the two codewords.

This property of pairs of codewords being far is formalized as the minimum distance of a code. The *Hamming distance* between two codewords is the number of positions in which the two codewords differ. Recall that the minimum distance of a code is simply the minimum Hamming distance between any two codewords of the code. The minimum distance property of a code has an intuitive geometric property. If one represents codewords by points in space, then the property that the code has minimum distance d implies that closed Hamming balls of radius $(d-1)/2$ around each codeword are disjoint. Therefore, if a code has minimum distance d, then corrupting up to $(d-1)/2$ symbols in a codeword results in a string whose closest codeword is still the original codeword. This provides a decoding mechanism that can tolerate any pattern of up to $(d-1)/2$ errors in the transmission. In general, for a fixed rate or dimension of a code, one wishes to maximize the minimum distance d. This amounts to a sphere-packing problem in which the metric is given by Hamming distance. This interpretation has allowed the use of techniques from geometry, combinatorics, etc. to coding theory.

1.2 Overview of the Thesis

The primary focus of thesis thesis is on determining *capacity* and *limitations* of structures in coding theory. One of the fundamental tradeoffs in coding theory is between the amount of errors that can be tolerated and the amount of redundancy that one adds. The capacity of communication channels is a notion that quantifies the optimal tradeoff in the case of probabilistic errors, where the channel determines a particular error model and capacity detemines the redundancy. The main topics of this thesis concern understanding this tradeoff as well as constructing coding schemes that try to achieve optimal tradeoffs. Specifically, we discuss the following topics in this thesis:

Polar codes. One important question in the realm of coding theory has been the explicit construction of capacity-achieving error correcting codes over various channels. A major breakthrough in this area was made recently by Arikan [Arı09], who discovered a new class of capacity-achieving codes known as *polar codes*. Polar codes are efficiently encodable and decodable and have been shown to achieve capacity for symmetric channels.

Our contribution is to analyze the convergence properties of these codes to capacity [GV15]. In the process, we introduce an interesting technique involving *entropy sumset inequalities*. This contribution is discussed in Section 3.

Coding in the interactive setting. Classical coding theory has primarily dealt with the setting of one-way communication, in which a single party wishes to transmit a message. However, with the advent of such notions such as communication complexity and information complexity, there has recently been much interest in coding for interactive two-party communication protocols. In hopes of better understanding the limits of coding schemes in such a setting, we consider questions regarding the capacity of interactive channels in Section 4 [HV16].

List decodability and local testability. We also investigate limitations of error-correcting codes with additional structure. Two important structural notions that have gained importance in recent years are *list decodability* and *local testability*. The former allows decoding beyond the unique decoding radius by allowing a decoder to ouptut a small list of possible messages corresopnding to a received word. In Section 5, we investigate list decodability properties of random linear codes and introduce new techniques that exhibit surprising connections to the area of *compressed sensing* [CGV13].

Local testability is a notion of locality that allows one to test a received word for membership in the code by querying a small number of positions in the word and has connections to property testing and probabilistically checkable proofs (PCPs) in complexity theory. In Section 6, we investigate local testability for a class of codes known as *affine-invariant codes* [GSVW15].

Chapter 2

Preliminaries

In this chapter, we provide some background information about information theory and error-correcting codes and introduce notation and definitions.

2.1 Basic Information Theory

We now introduce some basic information theory, as we will make use of information theory concepts throughout this thesis. We are primarily concerned with discrete random variables. A discrete random variable X is specified by $\mathcal{X}$, the set of values that X can take, along with a probability distribution $\{p_X(x)\}_{x\in\mathcal{X}}$ satisfying the normalization condition $\sum_{x\in\mathcal{X}} p_X(x) = 1$.

2.1.1 Entropy

We now define the *entropy* of a random variable.

Definition 2. *The* entropy *(in bits) of a random variable X with an underlying probability distribution p is defined as*

$$H(X) = -\sum_{x\in\mathcal{X}} p(x)\log_2 p(x) = \mathbf{E}\left[\log_2(1/p(x))\right],$$

where we define $0\log_2 0 = 0$ (by continuity).

Note that the above definition measures entropy in *bits*, since the logarithms are base 2. We can also measure entropy in different units by using logarithms with a different base, which results in a quantity that differs by a multiplicative constant.

Roughly speaking, the entropy provides a measure of the *uncertainty* of a random variable. A larger entropy corresponds to less a priori information about the variable. For instance, suppose a random variable X takes values in a size of set n. If X is uniformly distributed (i.e., X has maximum uncertainty), then the entropy of X is

$$H(X) = n\left(-\frac{1}{n}\log_2\frac{1}{n}\right) = \log_2 n.$$

On the other hand, if X takes a particular value with probability 1 and all other $n-1$ values with probability 0 (i.e., X is deterministic), then

$$H(X) = -1\log_2 1 - (n-1)\cdot 0\log_2 0 = 0.$$

As it turns out, if X is a random variable taking values in a set of size n, then its entropy always satisfies $0 \le H(X) \le \log_2 n$.

As a special case, one can consider a random variable X which takes values in $\{0,1\}$. Then, note that X is completely specified by the parameter $p = \Pr[X=0]$. In this case, the entropy $H(X)$ is given by

$$H(X) = -p\log_2 p - (1-p)\log_2(1-p).$$

Since the above function of p arises repeatedly in many settings, we define the function as follows:

Definition 3. *The* binary entropy function h *is defined by*

$$h(x) = x\log_2 x - (1-x)\log_2(1-x).$$

Thus, $H(X) = h(p)$ for our binary random variable X. In a slight abuse of notation, we will often use $H(p)$ to denote the value of the binary entropy function at p (e.g., in Chapter 4); however, in such an event, there will be no ambiguity, as the argument to $H(\cdot)$ will be a real number instead of a random variable.

2.1.2 Conditional Entropy

Suppose X and Y are random variables taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively. Then, let $p(x,y)$ denote their joint probability distribution (where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$), and let $p_X(x)$

and $p_Y(y)$ denote the marginal probability distributions for X and Y, respectively (we often abbreviate them as $p(x)$ or $p(y)$ when the context is clear). Then, we can write the conditional probability distribution (for y given x) as $p(y|x)$. Recall that $p(y|x)$ is given by Bayes' rule:

$$p(y|x) = p(x, y)|p(x).$$

Now, we can define a quantity known as the *conditional entropy*.

Definition 4. *The conditional entropy $H(Y|X)$ is given by*

$$H(Y|X) = -\sum_{x\in\mathcal{X}} p(x) \sum_{y\in\mathcal{Y}} p(y|x) \log_2 p(y|x).$$

Intuitively, $H(Y|X)$ quantifies the amount of information gained from $y \sim Y$ conditional on knowing $x \sim X$. It is not too hard to verify that the following identity holds:

$$H(Y|X) = H(X, Y) - H(X).$$

In general, a *chain rule* for multiple random variables holds. Given random variables $X_1, X_2, \ldots, X_k$, we have

$$H(X_1, X_2, \ldots, X_k) = \sum_{i=1}^{k} H(X_i|X_1, X_2, \ldots, X_{i-1}).$$

2.1.3 Mutual Information

Recall that if we have two random variables X and Y, then the total entropy of (X, Y) is given by

$$H(X, Y) = H(X) + H(Y|X).$$

In the special case that X and Y are independent, we have $H(Y|X) = H(Y)$, and so,

$$H(X, Y) = H(X) + H(Y).$$

Moreover, in this special case, information about X does not provide information about Y. However, in the more general setting where X and Y may be correlated, one generally obtains some information about Y from X. We quantify this concept through the notion of *mutual information*:

Definition 5. *Given two random variables X and Y taking values in $\mathcal{X}$ and $\mathcal{Y}$, respectively, we define their* mutual information *to be*

$$I(X;Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)} \right).$$

A simple consequence of Jensen's Inequality is the following theorem:

Theorem 1. *For random variables X and Y, we have that $I(X;Y) \geq 0$, i.e., the mutual information of X and Y is nonnegative. Moreover, $I(X;Y) = 0$ if and only if X and Y are independent random variables.*

The mutual information of X and Y intuitively quantifies the amount of information one learns about y by revealing x. This will be a useful notion later on in Section 2.3 while discussing the capacity of commmunication channels.

2.2 Basic Definitions for Error-Correcting Codes

Given an integer $q \geq 2$, we write $[q] = \{1, 2, \dots, q\}$. We define the Hamming distance of two q-ary strings as follows:

Definition 6. *For any strings $\mathbf{x}, \mathbf{y} \in [q]^n$, where $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_n)$, we define the Hamming distance between $\mathbf{x}$ and $\mathbf{y}$, denoted $\Delta(\mathbf{x}, \mathbf{y})$, to be the number of coordinates $1 \leq i \leq n$ for which $x_i \neq y_i$.*

Moreover, we define the Hamming weight of a string:

Definition 7. *For any string $\mathbf{x} \in [q]^n$, we define the* Hamming weight *of $\mathbf{x}$ to be the number of coordinates where $\mathbf{x}$ is zero.*

Now, we define an error-correcting code.

Definition 8. *An* error-correcting code *(or* code*) over a q-ary alphabet is a set $\mathcal{C} \subseteq [q]^n$. The* codewords *of $\mathcal{C}$ are the individual elements of $\mathcal{C}$.*

The most common case occurs when $q = 2$, in which case the code $\mathcal{C}$ is said to be a *binary* code.

Definition 9. *Given a code $\mathcal{C} \subseteq [q]^n$ over q-ary alphabet, we define the folowing quantities:*

- *The* block length *of $\mathcal{C}$ is defined to be n.*
- *The* dimension *of $\mathcal{C}$ is defined to be* $\log_q |\mathcal{C}|$.
- *The* rate *of $\mathcal{C}$ is defined to be* $\frac{\log_q |\mathcal{C}|}{n}$.
- *The* minimum distance *(or just* distance*) of $\mathcal{C}$, denoted* $\operatorname{dist}(C)$*, is defined as*

  $$\operatorname{dist}(\mathcal{C}) = \min_{\substack{c_1, c_2 \in \mathcal{C} \\ c_1 \neq c_2}} \Delta(c_1, c_2).$$

  In other words, $\operatorname{dist}(\mathcal{C})$ *is the minimum Hamming distance between two distinct codewords of $\mathcal{C}$.*
- *The* relative distance *of $\mathcal{C}$, denoted $\delta(\mathcal{C})$, is defined as*

  $$\delta(\mathcal{C}) = \frac{\operatorname{dist}(\mathcal{C})}{n}.$$

Definition 10. *We say that an error-correcting code $\mathcal{C}$ is an $[n, k, d]_q$ code if $\mathcal{C} \subseteq [q]^n$ and* $\dim(\mathcal{C}) = k$ *as well as* $\operatorname{dist}(\mathcal{C}) = d$*. Furthermore, we often omit the minimum distance and refer to a code $\mathcal{C}$ as an $[n, k]_q$ code if $\mathcal{C} \subseteq [q]^n$ and* $\dim(\mathcal{C}) = k$*. When the alphabet size q is understood from context, we often omit the subscript q.*

Although a code $\mathcal{C}$ is defined to be simply a set of tuples, it is often useful to view $\mathcal{C}$ explicity in terms of an *encoding function*. More precisely, suppose $|\mathcal{C}| = M$. Then, we can view $\mathcal{C}$ as a function $\mathcal{C} : [M] \to [q]^n$. Each element of $[M]$ is considered a *message*, and the function maps each message to a codeword of $\mathcal{C}$. It will generally be the case that $M = q^k$ for an integer $k = \dim(\mathcal{C})$, in which case we can identify the message space $[M]$ with the set of q-ary strings of length k. Thus, the function $\mathcal{C} : [M] \to [q]^n$ can be viewed as encoding messages that are q-ary strings of length k into longer q-ary strings of length n. Note that the encoding function for $\mathcal{C}$ is not unique!

2.2.1 Linearity

One convenient property for an error-correcting code to have is *linearity*.

Definition 11. *Let q be a prime power. Then, a code $\mathcal{C} \subseteq [q]^n$ is said to be* linear *if it forms a linear subspace of $\mathbb{F}_q^n$.*

Note that if $\mathcal{C}$ is a linear code, then $\dim(\mathcal{C})$ is equal to the dimension of the corresponding vector space over $\mathbb{F}_q$. Also, observe that a linear code always contains the all-zeroes string as a codeword, and the minimum distance of a linear code is the minimum Hamming weight of a non-zero codeword.

Linearity is a useful property, as it allows a code to be specified in terms of a *generator matrix* or a *parity check matrix*:

- An $[n, k]_q$ linear code $\mathcal{C}$ can be expressed as
$$\mathcal{C} = \{G^T \cdot \mathbf{x} : \mathbf{x} \in \mathbb{F}_q^k\}$$
for some $k \times n$ matrix G. Such a matrix G is called a *generator matrix* for $\mathcal{C}$.
- An $[n, k]_q$ linear code $\mathcal{C}$ can be expressed as
$$\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n : H\mathbf{c} = \mathbf{0}\}$$
for some $(n - k) \times n$ matrix H. Such a matrix H is called a *parity check matrix* of $\mathcal{C}$.

A number of important error-correcting codes happen to be linear (e.g., Reed-Solomon, Reed-Muller, low-density parity-check (LDPC) codes), which is an important motivation for studying codes with this property.

It is often useful to define a *dual code* for a linear code:

Definition 12 (Dual code)**.** *Given an $[n, k]_q$ linear code $\mathcal{C}$, we define its* dual code $\mathcal{C}^\perp$ *to be the code given by*

$$\mathcal{C}^\perp = \{\mathbf{c}' \in \mathbb{F}_q^n : \mathbf{c}^T \cdot \mathbf{c}' = 0 \textit{ for all } \mathbf{c} \in \mathcal{C}\}.$$

It is not too hard to show that if $\mathcal{C}$ is an $[n, k]_q$ linear code, then $\mathcal{C}^\perp$ is an $[n, n - k]_q$ linear code.

2.3 Communication Channels and Channel Coding

Now, we introduce the notion of a communication channels. Recall that error-correcting codes are used in order to ensure reliable communication over a noisy medium. Communication channels model the probabilistic behavior of noisy mediums. The problem of communication over a noisy channel is often referred to as *channel coding*.

Definition 13. *A* discrete memoryless channel *(DMC)* $W = (\mathcal{X}, \mathcal{Y}, p)$ *consists of an input alphabet* $\mathcal{X}$*, an output alphabet* $\mathcal{Y}$*, and a set of transition probabilities* $p(y|x)$ *for all* $x \in \mathcal{X}$ *and* $y \in \mathcal{Y}$*. (Note that* $\sum_{y \in \mathcal{Y}} = p(y|x) = 1$ *for any* $x \in \mathcal{X}$*.)*

One can view a DMC as taking in a symbol from $\mathcal{X}$ and outputting a symbol from $\mathcal{Y}$ according to conditional probability distribution of the channel. Thus, the channel models a probabilistic corruption of the input symbol. Note that a DMC is called memoryless because the channel behaves independently on every use of the channel, i.e., if two symbols $x_1, x_2 \in \mathcal{X}$ are fed into the channel, then the output symbol for x_1 is independent of the output symbol for x_2.

Also, any DMC can be specified uniquely by its transition probability matrix, i.e., the matrix consisting of the entries $p(y|x)$ (where rows are indexed by $\mathcal{X}$ and columns are indexed by $\mathcal{Y}$) such that the rows all sum to 1.

One of the simplest examples of a channel is the *binary symmetric channel* (BSC). For this channel, $\mathcal{X} = \mathcal{Y} = \{0, 1\}$, and $p(1|0) = p(0|1) = \epsilon$ and $p(0|0) = p(1|1) = 1 - \epsilon$, where $0 \leq \epsilon \leq 1$ is a parameter known as the *crossover probability*. In other words, in a BSC with crossover probability ϵ (referred to as BSC_ϵ, an input bit is flipped with probability ϵ, while it is left alone with probability $1 - \epsilon$. Note that if $\epsilon = 0$, then the channel is a "perfect" channel, as the output bit is always equal to the input bit. On the other hand, if $\epsilon = 1/2$, then the channel is "completely noisy," as the output bit is simply a uniformly random bit, regardless of the input.

Another important well-known example of a channel is the *binary erasure channel* (BEC). For the BEC, $\mathcal{X} = \{0, 1\}$, while $\mathcal{Y} = \{0, 1, ?\}$. Moreover, the underlying conditional probability distribution of the BEC is given by $p(?|0) = p(?|1) = \epsilon$ and $p(0|0) = p(1|1) = 1 - \epsilon$ and $p(0|1) = p(1|0) = 0$, where $\epsilon > 0$ is the *erasure probability*. One can view the output symbol '?' as an "erasure," i.e., the channel takes in an input bit and erases it with probability $\epsilon > 0$; otherwise, it outputs the same bit. A BEC with erasure probability ϵ is often referred to as BEC_ϵ. Again, note that if $\epsilon = 0$, then BEC_ϵ is a perfect channel, while if $\epsilon = 1$, then the output is always '?' irrespective of the input.

The work of Shannon shows that any channel has an associated constant known as the *channel capacity*, which determines how much redundancy is needed to enable reliable communication over the channel.

Theorem 2 (Shannon's noisy channel coding theorem [Sha48])**.** *For any discrete memoryless channel* $W = (\mathcal{X}, \mathcal{Y}, \Pi)$*, there exists a constant* $C(W) \geq 0$ *known as the* channel capacity *(or simply* capacity*) such that:*

1. *For any* $R < C(W)$*, for large enough* N*, there exists an error-correcting code over*

alphabet $\mathcal{X}$ of block length N and rate $\geq R$ along with a decoder such that the probability of an error in decoding is $2^{-\Omega_{R,C(W)}(n)}$.

2. *For any $R > C(W)$, it is impossible to find an error-correcting code for sufficiently large N such that the probability of an error in decoding is $< \delta$ for all $\delta > 0$.*

The channel capacity is not always easy to compute. However, for the simple examples of channels discussed above, we know the capacity. As it turns out, the capacity of BEC_ϵ is precisely $1 - \epsilon$. Note that if $\epsilon = 0$, then the capacity is 1, implying that no redundancy is required to communicate reliably over the channel—this agrees with the fact that BEC_ϵ is a perfect channel in the case $\epsilon = 0$. On the other hand, if $\epsilon = 1$, then the capacity is 0, meaning that one cannot possibly hope to communicate reliably over the channel and achieve any positive communication rate. This is expected, since in the case $\epsilon = 1$, the channel simply outputs '?' all the time and loses any possible information about the input symbol.

For BSC_ϵ, the channel capacity is a bit more complicated. In particular, the capacity is $1 - h(\epsilon)$, where h is the binary entropy function defined in Definition 3. Again note that for $\epsilon = 0$, we have that the capacity is $1 - h(\epsilon) = 1$, which means that no redundancy is required for reliable communication over BSC_ϵ. On the other hand, for $\epsilon = 1/2$, we have that the capacity is $1 - h(\epsilon) = 0$, which means that no positive communication rate can be obtained.

The following theorem provides a general expression for the channel capacity in terms of mutual entropy:

Theorem 3. *For any DMC W, the channel capacity $C(W)$ is given by*

$$C(W) = \max_{p(X)} I(X;Y),$$

where X and Y are random variables for the input and output, respectively, of W, and $p(X)$ (over which the maximum is computed) denotes a probability distribution for X.

Recall that the channel W determines *conditional* probabilities $p(y|x)$ for all x, y. Thus, if one specifies a probability distribution p for X, then this determines a joint probability distribution (X, Y). The above theorem states that the channel capacity happens to be the maximum of the mutual information of X and Y over all possible choices of the distribution of X.

It is often useful to consider *symmetric* DMCs, which are DMCs whose probability transition matrix satisfies some symmetry properties:

Definition 14. *A DMC is said to be* symmetric *if one can group the columns of the transition matrix of the channel into submatrices such that in each submatrix, the following properties hold:*

- *Each row is a permutation of every other row.*
- *Each column is a permutation of every other column.*

As an example, recall that the transition matrix of the binary erasure channel BEC_ϵ is

$$\begin{pmatrix} 1-\epsilon & \epsilon & 0 \\ 0 & \epsilon & 1-\epsilon \end{pmatrix},$$

whose columns can be divided to form the following submatrices:

$$\begin{pmatrix} 1-\epsilon & 0 \\ 0 & 1-\epsilon \end{pmatrix}, \qquad \begin{pmatrix} \epsilon \\ \epsilon \end{pmatrix}.$$

Since each of the above matrices satisfies the conditions listed in Definition 14, we see that BEC_ϵ is a symmetric channel.

A useful property about symmetric channels is that the capacity is achieved by a uniformly distributed input. In other words, if W is a symmetric DMC with input X over $\mathcal{X}$ and output Y over $\mathcal{Y}$, then $I(X;Y)$ is maximized when X is a uniformly random over $\mathcal{X}$.

It is often useful to consider the notion of the *symmetric capacity* of a channel, which is the capacity achievable by a uniformly distributed input:

Definition 15. *Given a DMC $W = (\mathcal{X}, \mathcal{Y}, \Pi)$, we define the* symmetric capacity *of W to be $I(X;Y)$ where X and Y are random variables for the input and output, respectively, of W, and X is taken to be* uniformly distributed *over $\mathcal{X}$.*

Again, it is not too hard to show that for a symmetric channel W, the capacity and symmetric capacity are equal, i.e., taking X to be uniformly distributed achieves the maximum value of $I(X;Y)$ in Theorem 3.

2.4 Source Coding and Data Compression

Another natural problem in information theory that is the problem of *source coding*, which deals with data compression. More precisely, we have a source of symbols, and we wish to map sequences of symbols into sequences of codewords. In this section, we will consider *lossless* source coding, which means that the original sequence of symbols should be exactly reconstructible from the sequence of codewords. The general goal is to minimize the average codeword length per source symbol. One can also consider *lossy* source coding, in which one can allow some loss in the reconstruction process. Source codng techniques have a range of applications and are used in various file archivers, audio compression standards, video compression standards, etc.

Consider a source, which is modeled by a random variable X over a set of symbols $\mathcal{X}$. Without loss of generality, we assume $\mathcal{X} = \{0, 1, \ldots, m-1\}$, where m is the number of possible source symbols. Now, let X_n denote an i.i.d. sequence of random variables sampled from X. We wish to map X_n into a string over symbols from an output alphabet $\mathcal{Y}$. Given any alphabet Σ, we denote the set of finite strings with symbols from Σ as Σ^*. We define a *symbol code* as follows:

Definition 16. *A* symbol code *C is a mapping $C : \mathcal{X} \to \mathcal{Y}^*$. As an extension of C, we also define $C(x_1x_2\ldots x_n) = C(x_1)C(x_2)\ldots C(x_n)$ for $x_1, x_2, \ldots, x_n \in \mathcal{X}$, which determines a mapping $C : \mathcal{X}^* \to \mathcal{Y}^*$.*

A symbol code is said to be *uniquely decodable* if any two distinct sequences of symbols in $\mathcal{X}$ map to distinct sequences of symbols in $\mathcal{Y}$ under the extension. An important class of uniquely decodable codes, are *prefix codes*:

Definition 17. *A symbol code C is said to be a* prefix code *if no sequence in the image of C is the prefix of another sequence in the image of C, i.e., for any distinct $x, x' \in \mathcal{X}$, $C(x)$ is not a prefix of $C(x')$.*

Note that in general, different elements of $\mathcal{X}$ can map to sequences of different lengths under a symbol code. For a symbol $x \in \mathcal{X}$, let $l(x) = |C(x)|$ denote the length of the sequence $C(x)$. Then, we have the following inequality for uniquely decodable symbol codes:

Theorem 4 (Kraft-McMillan inequality [Kra49, McM56])**.** *For any uniquely decodable symbol code $C : \mathcal{X} \to \{0, 1, \ldots, D-1\}^*$, we have*

$$\sum_{x \in \mathcal{X}} D^{-l(x)} \leq 1.$$

Conversely, given any choice of sets $\{l_x\}_{x \in \mathcal{X}}$ of positive integers satisfying $\sum_{x \in \mathcal{X}} D^{-l_x} \leq 1$, there exists a prefix code $C : \mathcal{X} \to \{0, 1, \ldots, D-1\}^*$ such that l_x is the length of $C(x)$ for all $x \in \mathcal{X}$.

Now, let $p_i = \Pr[X = i]$. Assume that $\mathcal{Y} = \{0, 1\}$ for simplicity. Then, for a symbol code C, it is clear that the expected number of bits per symbol of $\mathcal{X}$ in the encoding of a random sequence X_n is given by

$$L = \sum_{i=0}^{m-1} p_i l(i).$$

The general goal is to minimize L. As it turns out, Shannon [Sha48] determined the optimal number of bits per symbol for a code:

Theorem 5 (Shannon source coding theorem [Sha48])**.** *A collection of N i.i.d. random variables sampled from a source X (with entropy $H(X)$) can be compressed into $N(H(X) + \epsilon)$ bits with negligible probability of information loss as $N \to \infty$. Conversely, there is no way to compress them into fewer than $NH(X)$ bits with negligible probability of information loss.*

Finding explicit constructions of source codes that achieve, on average, $\approx H(X)$ bits per symbol is a challenging task. A number of source codes that achieve close to this rate have been developed over time (e.g., Shannon codes, Huffman codes [Huf52]). In Chapter 3, we will discuss how to use polar codes to solve the lossless source coding problem.

Chapter 3

Achieving Channel Capacity for Error-Correcting Codes

The results of this chapter were published in [GV15].

3.1 Introduction

In this section, we concentrate on reliable communication in the presence of *random* errors. Recall Shannon's noisy channel coding theorem, which guarantees the existence of a channel capacity for any discrete memoryless channel:

Theorem 2 (Shannon's noisy channel coding theorem [Sha48])**.** *For any discrete memoryless channel $W = (\mathcal{X}, \mathcal{Y}, \Pi)$, there exists a constant $C(W) \geq 0$ known as the* channel capacity *(or simply* capacity*) such that:*

1. *For any $R < C(W)$, for large enough N, there exists an error-correcting code over alphabet $\mathcal{X}$ of block length N and rate $\geq R$ along with a decoder such that the probability of an error in decoding is $2^{-\Omega_{R,C(W)}(n)}$.*

2. *For any $R > C(W)$, it is impossible to find an error-correcting code for sufficiently large N such that the probability of an error in decoding is $< \delta$ for all $\delta > 0$.*

The result of Shannon actually implies that there exists a constant a_W such that for any *gap to capacity* $\epsilon > 0$ and $N \geq a_W/\epsilon^2$, there exists a binary code $C \subset \{0,1\}^N$ of rate at least $R \geq C(W) - \epsilon$ that enables reliable communication. In fact, random codes with the appropriate block length N and rate $C(W) - \epsilon$ satisfy the desired property with high probability. However, Shannon's theorem says nothing about *how to construct* such a code.

Over the past several decades, the existential result of Shannon has guided the quest of coding theorists to find *explicit* constructions of codes that achieve the parameters guaranteed by random codes while having encoding and decoding procedures that are *efficient* (say, with running time that is polynomial in $1/\epsilon$ for a gap to capacity of ϵ). Numerous codes have been constructed, but most constructions either fail to provably achieve capacity for a large enough class of channels, or do not have efficient encoders/decoders.

For instance, Forney's construction of *concatenated codes* has long been known to achieve capacity, but the time complexity of the decoder is unfortunately not efficient in terms of the gap to capacity. Similarly, the widely used low-density parity-check (LDPC) codes are known to achieve capacity only for the binary erasure channel but not for general channels. Turbo codes perform well in practice and have been employed in a number of applications, but they are not known to achieve capacity arbitrarily closely.

A recent breakthrough was made when, in a remarkable work, Arıkan introduced the technique of *channel polarization* and used it to construct a family of binary linear codes called *polar codes* that achieve the symmetric Shannon capacity of binary-input discrete memoryless channels in the limit of large block lengths [Arı09]. Polar codes are based on an elegant recursive construction and analysis guided by information-theoretic intuition. Arıkan's work gave a construction of binary codes, and this was subsequently extended to general alphabets [cTA09]. In addition to being an approach to realize Shannon capacity that is radically different from prior ones, channel polarization turns out to be a powerful and versatile primitive applicable in many other important information-theoretic scenarios. For instance, variants of the polar coding approach give solutions to the lossless and lossy source coding problem [Arı10, KU10], capacity of wiretap channels [MV11], the Slepian-Wolf, Wyner-Ziv, and Gelfand-Pinsker problems [Kor10], coding for broadcast channels [GAG13], multiple access channels [cTY13, AT12], interference networks [Wc14], etc. We recommend the well-written survey by Şaşoğlu [Ş12] for a detailed introduction to polar codes.

The advantage of polar codes over previous capacity-achieving methods (such as Forney's concatenated codes that provably achieved capacity) was highlighted in a recent work of Guruswami and Xia [GX13], where *polynomial convergence to capacity* was shown in the *binary* case (this was also shown independently by Hassani et al. [HAU13]). Specifically, it was shown that polar codes enable approaching the symmetric capacity of binary-input memoryless channels within an additive gap of ϵ with block length, construction, and encoding/decoding complexity all bounded by a polynomially growing function of $1/\epsilon$. Polar codes are the first and currently only known construction which provably has this property, thus providing a formal complexity-theoretic sense in which they are the first constructive capacity-achieving codes.

Our main objective in this chapter is to extend this result to the non-binary case, and we manage to do this for *all* alphabets in. We stress that the best previously proven complexity bound for communicating at rates within ϵ of capacity of channels with non-binary inputs was *exponential* in $1/\epsilon$. Our work shows the polynomial solvability of the central computational challenge raised by Shannon's non-constructive coding theorems, in the full generality of *all discrete sources* (for compression/noiseless coding) and *all discrete memoryless channels* (for noisy coding).

The high level approach to prove the polynomially fast convergence to capacity is similar to what was done in [GX13], which is to replace the appeal to general martingale convergence theorems (which lead to ineffective bounds) with a more direct analysis of the convergence rate of a specific martingale of entropies.[1] However, the extension to the non-binary case is far from immediate, and we need to establish a quantitatively strong "entropy increase lemma" (see details in Section 3.4) over all prime alphabets. The corresponding inequality admits an easier proof in the binary case, but requires more work for general prime alphabets. For alphabets of size m where m is not a prime, we can construct a capacity-achieving code by combining together polar codes for each prime dividing m.

In the next few sections, we briefly sketch the high level structure of polar codes, and the crucial role played by a certain "entropy sumset inequality" in our effective analysis. Proving this entropic inequality is the main new component in this work, though additional technical work is needed to glue it together with several other ingredients to yield the overall coding result.

3.2 Fundamentals of Polar Codes

In this section, we describe the basic construction of polar codes.

Notation. We begin by setting some of the notation to be used in this section. We will let $\lg$ denote the base 2 logarithm, while $\ln$ will denote the natural logarithm.

For our purposes, unless otherwise stated, q will be a prime integer, and we identify $\mathbb{Z}_q = \{0, 1, 2, \ldots, q-1\}$ with the additive group of integers modulo q. We will generally

[1] The approach taken in [HAU13] to analyze the speed of polarization for the binary was different, based on channel Bhattacharyya parameters instead of entropies. This approach does not seem as flexible as the entropic one to generalize to larger alphabets.

view $\mathbb{Z}_q$ as a q-ary alphabet.

For this section, given a q-ary random variable X taking values in $\mathbb{Z}_q$, we let $H(X)$ denote the *normalized entropy* of X:

$$H(X) = -\frac{1}{\lg q} \sum_{a \in \mathbb{Z}_q} \Pr[X = a] \lg(\Pr[X = a]).$$

Note that this notation is different from the usual definition of $H(X)$ in which the $1/\lg q$ factor is not present. In a slight abuse of notation, we also define $H(p)$ for a probability distribution p. If p is a probability distribution over $\mathbb{Z}_q$, then we shall let $H(p) = H(X)$, where X is a random variable sampled according to p. Also, for nonnegative constants $c_0, c_1, \ldots, c_{q-1}$ summing to 1, we will often write $H(c_0, \ldots, c_{q-1})$ as the entropy of the probability distribution on $\mathbb{Z}_q$ that samples i with probability c_i. Moreover, for a probability distribution p over $\mathbb{Z}_q$, we let $p^{(+j)}$ denote the j^{th} *cyclic shift* of p, namely, the probability distribution $p^{(+j)}$ over $\mathbb{Z}_q$ that satisfies

$$p^{(+j)}(m) = p(m - j)$$

for all $m \in \mathbb{Z}_q$, where $m - j$ is taken modulo q. Note that $H(p) = H(p^{(+j)})$ for all $j \in \mathbb{Z}_q$.

Also, let $\|\cdot\|_1$ denote the ℓ_1 norm on $\mathbb{R}^q$. In particular, for two probability distributions p and p', the quantity $\|p - p'\|_1$ will correspond to twice the total variational distance between p and p'.

Finally, given a row vector (tuple) $\vec{v}$, we let $\vec{v}^t$ denote a column vector given by the transpose of $\vec{v}$.

3.2.1 Source Coding: Intuition for Polarization

While polar codes can be used for both channel coding and source coding, we first consider the setting of source coding for simplicity. In this set-up, suppose we have a pair of discrete random variables (X, Y), where $X \in \mathbb{Z}_q$ and $Y \in \mathcal{Y}$. Note that the variables X and Y may be correlated. We will view X as a source and Y as side information about the source.

We consider N independent copies $(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)$ of (X, Y). In the source coding with side information framework, a receiver wishes to decode $X_1, \ldots, X_N$ after observing the side information $Y_1, \ldots, Y_N$. From elementary information theory, we know that it suffices (and is necessary) to provide the receiver with approximately $H(X_1, \ldots, X_N | Y_1, \ldots, Y_N) \cdot (\lg q)$ bits of information in order to allow $X_1, \ldots, X_N$ to be decoded with negligible probability of error.

There are two extremal cases in which decoding is obvious. One one hand, if we have $H(X_1, \ldots, X_N | Y_1, \ldots, Y_N) = 0$, then the receiver can decode $X_1, \ldots, X_N$ without any additional information. On the other hand, if $H(X_1, \ldots, X_N | Y_1, \ldots, Y_N) = 1$, then the optimal way to allow the decoder to decode $X_1, \ldots, X_N$ is to provide $X_1, \ldots, X_N$. The basic principle behind Arıkan's polarization technique is to transform $X_1, X_2, \ldots, X_N$ into a sequence such that the receiver need only perform a series of tasks, each of which corresponds to one of the aforementioned extremal cases.

3.2.2 Polarization Transform for Two Variables

Equipped with the intuition for polarization, we introduce Arıkan's polarization transform. Suppose $N = 2$. Then, we have two independent copies $(X_0, Y_0$ and (X_1, Y_1) of (X, Y). We define

$$U_0 = X_0 + X_1 \quad \text{and} \quad U_1 = X_1, \tag{3.1}$$

where addition is over $\mathbb{Z}_q$. Now, note that

$$\begin{aligned} 2H(X|Y) = H(X_0, X_1 | Y_0, Y_1) &= H(U_0, U_1 | Y_0, Y_1) \\ &= H(U_0 | Y_0, Y_1) + H(U_1 | Y_0, Y_1, U_0), \end{aligned}$$

by the chain rule for entropy. Morever, since conditioning can never increase entropy, we have that $H(U_1 | Y_0, Y_1, U_0) = H(X_1 | Y_0, Y_1, U_0) \leq H(X_1 | Y_1) = H(X|Y)$. Thus, it follows that

$$H(U_1 | Y_0, Y_1, U_0) \leq H(X|Y) \leq H(U_0 | Y_0, Y_1).$$

Thus, getting access to Y_0, Y_1, U_0 produces a better estimate of U_1 than getting access to just Y_1. Moreover, observing Y_0, Y_1 gives a worse estimate of U_0 than observing Y_0 would give for X_0.

The basic principle regarding polarization is that we have taken two identical conditional entropies $H(X_0 | Y_0)$ and $H(X_1 | Y_1)$ and produced two different entropies, namely, $H(U_1 | Y_0, Y_1, U_0)$ and $H(U_0 | Y_0, Y_1)$. Thus, one of the new entropies is closer to 0 than the original entropy, while the other is closer to 1.

3.2.3 Extending the Polarization Transform to More Copies

Now, it turns out that we can extend the procedure from the previous section to more copies of (X, Y). Let us consider $N = 4$, so that we have four copies (X_0, Y_0), (X_1, Y_1), (X_2, Y_2), and (X_3, Y_3) of (X, Y).

For the first step, we perform the transform in (3.1) separately to X_0, X_1 and X_2, X_3. Thus, we have

$$S_0 = X_0 + X_1 \quad \text{and} \quad S_1 = X_1$$

and

$$T_0 = X_2 + X_3 \quad \text{and} \quad T_1 = X_3.$$

Now, we apply another layer of the tranform from (3.1) separately to S_0, T_0 and S_1, T_1. Then, we have

$$\begin{aligned} U_0 &= S_0 + T_0 = X_0 + X_1 + X_2 + X_3 \\ U_1 &= T_0 = X_2 + X_3 \\ U_2 &= S_1 + T_1 = X_1 + X_3 \\ U_3 &= T_1 = X_3. \end{aligned}$$

With some computation, we see that

$$H(U_1|Y_0, \dots, Y_3, U_0) \leq H(S_0|Y_0, Y_1) \leq H(U_0|Y_0, \dots, Y_3)$$
$$H(U_3|Y_0, \dots, Y_3, U_0, \dots, U_2) \leq H(S_1|Y_0, Y_1, S_0) \leq H(U_2|Y_0, \dots, Y_3, U_0, U_1)$$

Thus, we see that performing the 2-stage transformation with four copies of (X, Y) further polarizes the conditional entropies, i.e., starting with the two conditional entropies $H(S_0|Y_0, Y_1)$ and $H(S_1|Y_0, Y_1, S_0)$ (obtained from the previous section), we obtain four entropies that are separated even further.

The hope is that performing more stages of the 2×2 polarization map on a larger number of copies of (X, Y) would produce entropies that are more and more polarized, ideally approaching the extremes of 0 and 1. This turns out to be the case, and we discuss this further (see Theorem 6 and the subsequent discussion in Section 3.3).

3.2.4 Encoding Map: Recursive Construction

In general, we can extend the procedure of the previous sections to yield a polarization map for $N = 2^n$ copies of (X, Y). In this section, we formally define the polarization map that we will use to compress a source X. Given $n \geq 1$, we define an invertible linear transformation $G : \mathbb{Z}_q^{2^n} \to \mathbb{Z}_q^{2^n}$ by $G = G_n$, where $G_t : \mathbb{Z}_q^{2^t} \to \mathbb{Z}_q^{2^t}$, $0 \leq t \leq n$ is a sequence of invertible linear transformations defined as follows: G_0 is the identity map on $\mathbb{Z}_q$, and for any $0 \leq k < n$ and $\vec{X} = (X_0, X_1, \dots, X_{2^{k+1}-1})^t$, we recursively define $G_{k+1}\vec{X}$ as

$$G_{k+1}\vec{X} = \pi_{k+1}(G_k(X_0, \dots, X_{2^k-1}) + G_k(X_{2^k}, \dots, X_{2^{k+1}-1}), G_k(X_{2^k}, \dots, X_{2^{k+1}-1})),$$

where $\pi_{k+1} : \mathbb{Z}_q^{2^{k+1}} \to \mathbb{Z}_q^{2^{k+1}}$ is a permutation such that $\pi_n(v)_j = v_i$ for $j = 2i$, and $\pi_n(v)_j = v_{i+2^k}$ for $j = 2i + 1$.

G also has an explicit matrix form, namely, $G = B_n K^{\otimes n}$, where $K = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$, $\otimes$ is the Kronecker product, and B_n is the $2^n \times 2^n$ bit-reversal permutation matrix for n-bit strings (see [Arı10]).

In our set-up, we have a q-ary source X, and we let $\vec{X} = (X_0, X_1, \ldots, X_{2^n-1})^t$ be a collection of $N = 2^n$ i.i.d. samples from X. Moreover, we encode $\vec{X}$ as $\vec{U} = (U_0, U_1, \ldots, U_{2^n-1})^t$, given by $\vec{U} = G \cdot \vec{X}$. Note that G only has $0, 1$ entries, so each U_i is the sum (modulo q) of some subset of the X_i's.

3.2.5 Source Coding Through Polarization

We now describe how to get a source code from the encoding map (polarization map). First, we introduce the notion of a *virtual channel*.

Virtual Channels

For purposes of our analysis, we define a *virtual channel* (or, simply *channel*) $W = (A; B)$ to be a pair of correlated random variables A, B; moreover, we define the *channel entropy* of W to be $H(W) = H(A|B)$, i.e., the entropy of A conditioned on B.[2]

Given a channel W, we can define two channel transformations $-$ and $+$ as follows. Suppose we take two i.i.d. copies $(A_0; B_0)$ and $(A_1; B_1)$ of W. Then, W^- and W^+ are defined by

$$\begin{aligned} W^- &= (A_0 + A_1; B_0, B_1) \\ W^+ &= (A_1; A_0 + A_1, B_0, B_1). \end{aligned}$$

By the chain rule for entropy, we see that

$$H(W^-) + H(W^+) = 2H(W). \tag{3.2}$$

[2] It should be noted W can also be interpreted as a communication channel that takes in an input A and outputs B according to some conditional probability distribution. This is quite natural in the noisy channel coding setting in which one wishes to use a polar code for encoding data in order to achieve the channel capacity of a symmetric discrete memoryless channel. However, since we focus on the problem of source coding (data compression) rather than noisy channel coding in this thesis, we will simply view W as a pair of correlated random variables.

In other words, splitting two copies of W into W^- and W^+ preserves the total channel entropy. These channels are easily seen to obey $H(W^+) \leq H(W) \leq H(W^-)$, and the key to our analysis will be quantifying the separation in the entropies of the two split channels.

Source Coding Using the Encoding Map

The aformentioned channel transformations will help us abstract each step of the recursive polarization that occurs in the definition of our encoding map G. Let $W = (X; Y)$, where X is a source taking values in $\mathbb{Z}_q$, and Y can be viewed as side information. Then, $H(W) = H(X|Y)$. One special case occurs when $Y = 0$, which corresponds to an absence of side information.

Note that if start with W, then after n successive applications of either $W \mapsto W^-$ or $W \mapsto W^+$, we can obtain one of $N = 2^n$ possible channels in $\{W^s : s \in \{+,-\}^n\}$. (Here, if $s = s_0 s_1 \cdots s_{n-1}$, with each $s_i \in \{+,-\}$, then W^s denotes

$$W^s = (\cdots((W^{s_0})^{s_1})\cdots)^{s_{n-2}})^{s_{n-1}}.$$

By successive applications of (3.2), we know that

$$\sum_{s \in \{+,-\}^n} W^s = 2^n H(W) = 2^n H(X|Y).$$

Moreover, it can be verified (see [Ş12]) that if $0 \leq i < 2^n$ has binary representation $\overline{b_{n-1} b_{n-2} \cdots b_0}$ (with b_0 being the least significant bit of i), then

$$H(U_i | U_0, \ldots, U_{i-1}, Y_0, \ldots, Y_{N-1}) = H(W^{s_{n-1}s_{n-2}\cdots s_0}),$$

where $s_j = -$ if $b_j = 0$, and $s_j = +$ if $b_j = 1$. As shorthand notation, we will define the channel

$$W_n^{(i)} = W^{s_{n-1}s_{n-2}\cdots s_0},$$

where $s_0, s_1, \ldots, s_{n-1}$ are as above. Şaşoğlu et al. [cTA09] show that all but a vanishing fraction of the N channels W^s will be have channel entropy close to 0 or 1:

Theorem 6. *For any $\delta > 0$, we have that*

$$\lim_{n \to \infty} \frac{|\{s \in \{+,-\}^n : H(W^s) \in (\delta, 1-\delta)\}|}{2^n} = 0.$$

Hence, one can then argue that as n grows, the fraction of channels with channel entropy close to 1 approaches $H(X|Y)$. In particular, for any $\delta > 0$, if we let

$$\mathsf{High}_{n,\delta} = \{i : H(U_i|U_0, \dots, U_{i-1}, Y_0, \dots, Y_{N-1}) > \delta\}, \tag{3.3}$$

then

$$\frac{|\mathsf{High}_{n,\delta}|}{2^n} \to H(X|Y),$$

as $n \to \infty$. Then, as our source code, we can take $\{U_i\}_{i \in \mathsf{High}_{n,\delta}}$. Furthermore, as we discuss later, it can be shown that for any fixed $\epsilon > 0$ and small $\delta > 0$, there exists suitably large n such that $\{U_i\}_{i \in \mathsf{High}_{n,\delta}}$ gives a source coding of $\vec{X} = (X_0, X_1, \dots, X_{N-1})$ (with side information $\vec{Y} = (Y_0, Y_1, \dots, Y_{N-1})$) with rate $\leq H(X|Y) + \epsilon$. Our goal later on will be to show that $N = 2^n$ can be taken to be just polynomial in $1/\epsilon$ in order to obtain a rate $\leq H(X|Y) + \epsilon$.

Decoding

Having described the construction of the source code resulting from polarization, we now show how the decoding procedure operates. Recall that for some $N = 2^n$, the encoder has $(X_0, X_1, \dots, X_{N-1})$, obtained from the source, and computes $(U_0, U_1, \dots, U_{N-1})$ by applying the map $G = G_n$ to $(X_0, X_1, \dots, X_{N-1})$. Then, for some sufficiently small $\delta > 0$, Bob transmits all U_i for $i \in \mathsf{High}_{n,\delta}$, where $\mathsf{High}_{n,\delta}$ is given by (3.3).

For ease of notation, we use variables with lowercase letters to indicate realizations of the random variables with the corresponding uppercase letters. For instance, $x_0, \dots, x_{N-1}$ are the realizations of $X_0, \dots, X_{N-1}$

Now, the receiver attempts to compute estimates $\widehat{u}_i$ of U_i for *all* i in a successive fashion: Assuming $\widehat{u}_0, \widehat{u}_1, \dots, \widehat{u}_{i-1}$ have been computed, the receiver computes $\widehat{u}_i$ as follows:

- If $i \in \mathsf{High}_{n,\delta}$, then the receiver simply sets $\widehat{u}_i = u_i$, since the receiver has already received u_i.
- If $i \notin \mathsf{High}_{n,\delta}$, then it must be the case that $H(U_i|U_0, \dots, U_{i-1}, Y_0, \dots, Y_{N-1}) < \delta$, i.e., there is not much uncertainty in U_i given $U_0, \dots, U_{i-1}, Y_0, \dots, Y_{N-1}$. Thus, the receiver simply sets $\widehat{u}_i$ to be the most likely symbol, i.e.,

$$\widehat{u}_i = \arg\max_{a \in [q]} p_{U_i|U_0, \dots, U_{i-1}, Y_0, \dots, Y_{N-1}}(a \mid \widehat{u}_0, \widehat{u}_1, \dots, \widehat{u}_{i-1}, \widehat{y}_0, \dots, \widehat{y}_{N-1}).$$

At the end of the procedure, the receiver knows $\vec{u} = (\widehat{u}_0, \dots, \widehat{u}_{N-1})^t$ and can simply compute $G_n^{-1}\vec{u}$ to obtain $(\widehat{x}_0, \widehat{x}_1, \dots, \widehat{x}_{N-1})^t$, which is an estimate of $\vec{x} = (x_0, x_1, \dots, x_{N-1})^t$.

3.2.6 Polarization for Channel Coding

So far, we have discussed the use of Arıkan's technique of polarization in obtaining source codes. However, recall that our goal as outlined in the beginning of the chapter is to obtain *channel codes* that achieve Shannon capacity. As it turns out, the same technique of polarization can be used to obtain channel codes with some slight modifications. Although the rest of this chapter will primarily deal with the source coding framework for ease of exposition, we briefly describe the translation from source coding to channel coding. For a more in-depth discussion of this translation, one can consult [Ş12, Sec 2.4].

Suppose W is a q-ary input DMC with output alphabet $\mathcal{Y}$. Also, suppose $X_1, X_2, \ldots, X_N$ is a sequence of i.i.d. inputs to W, and let $Y_1, Y_2, \ldots, Y_N$ be the respective outputs under W, where $N = 2^n$. Observe that $(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)$ are also i.i.d.

Then, one can design a channel code as follows. Again, pick $\vec{U} = G_n \cdot \vec{X}$ and define $\mathsf{High}_{n,\delta}$ as in (3.3) for some sufficiently small choice of $\delta > 0$. Moreover, we let $\mathsf{High}^c_{n,\delta}$ be the complement of $\mathsf{High}_{n,\delta}$. We now describe the encoding and decoding procedures of the channel code.

- **Encoder**: Assume the encoder wishes to transmit a uniformly distributed message $M \in [q]^{|\mathsf{High}^c_{n,\delta}|}$. Then, for each $i \in \mathsf{High}_{n,\delta}$, we choose U_i independently and uniformly at random from $[q]$ (the symbols U_i for $i \in \mathsf{High}_{n,\delta}$ are often referred to as *frozen symbols*, since their values are fixed or "frozen" independently of the message M). Moreover, we set $\{U_i\}_{i \in \mathsf{High}^c_{n,\delta}} = M$. The encoder then then transmits $\vec{X} = G_n^{-1}(\vec{U})$ over W.

- **Decoder**: After receiving $Y_0, Y_1, \ldots, Y_{N-1}$ from the output of our channel W, the decoder simply decodes $U_0, U_1, \ldots, U_{N-1}$ successively as in Section 3.2.5. Outputting those U_i for which $i \in \mathsf{High}^c_{n,\delta}$ then produces the decoder's estimate of the original message M.

Note that the rate of the aforementioned channel code is $|\mathsf{High}^c_{n,\delta}|/N = 1 - (|\mathsf{High}_{n,\delta}|/N)$, which approaches $1 - H(X|Y)$ in the limit $N \to \infty$. Since M is uniform and the frozen symbols are chosen uniformly, we have that $\vec{U}$ is uniformly random in $\{0,1\}^N$. Thus, $\vec{X}$ is also uniformly random in $\{0,1\}^N$. This implies that the rate of the code actually approaches

$$H(X) - H(X|Y) = I(X;Y), \tag{3.4}$$

where X is uniform and the conditional distribution of Y given X is obviously determined by the channel W.

However, observe that the expression (3.4) is equal to the channel capacity of W if and only if $I(X;Y)$ is maximized by choosing X to be a uniformly random variable over $[q]$ (see Theorem 3). Thus, in order for the aforementioned polar codes to be capacity achieving, we need to make an assumption about the channel W. In particular, we assume that the channel is symmetric (see Definition 14), which guarantees that $I(X;Y)$ is maximized when X is uniformly random and that (3.4) is, indeed, the channel capacity.

It should be noted that one can also use polar codes for general channels that are not necessarily symmetric; however, the achievable rate in this case is the so-called *symmetric capacity* of the channel, which is defined to be $I(X;Y)$ for *uniformly random* X.

3.2.7 Bhattacharyya Parameter

In order to analyze a virtual channel $W = (X;Y)$, where X takes values in $\mathbb{Z}_q$, we will define the *q-ary source Bhattacharyya parameter* $Z_{\max}(W)$ of the channel W as

$$Z_{\max}(W) = \max_{d \neq 0} Z_d(W),$$

where

$$Z_d(W) = \sum_{x \in \mathbb{Z}_q} \sum_{y \in \text{Supp}(Y)} \sqrt{p(x,y)p(x+d,y)}.$$

Here, $p(x,y)$ is the probability that $X = x$ and $Y = y$ under the joint probability distribution (X,Y).

Now, the *maximum likelihood decoder* attempts to decode x given y by choosing the most likely symbol $\hat{x}$:

$$\hat{x} = \arg\max_{x' \in \mathbb{Z}_q} \Pr[X = x' | Y = y].$$

Let $P_e(W)$ be the probability of an error under maximum likelihood decoding, i.e., the probability that $\hat{x} \neq x$ (or the defining $\arg\max$ for $\hat{x}$ is not unique) for random $(x,y) \sim (X,Y)$. It is known (see Proposition 4.7 in [Ş12]) that $Z_{\max}(W)$ provides an upper bound on $P_e(W)$:

Lemma 1. *If W is a channel with q-ary input, then the error probability of the maximum-likelihood decoder for a single channel use satisfies $P_e(W) \leq (q-1)Z_{\max}(W)$.*

Next, the following proposition shows how the $Z_{\max}$ operator behaves on the polarized channels W^- and W^+. For a proof, see Proposition 4.16 in [Ş12].

Lemma 2. $Z_{\max}(W^+) \leq Z_{\max}(W)^2$*, and* $Z_{\max}(W^-) \leq q^3 Z_{\max}(W)$*.*

Finally, the following lemma shows that $Z_{\max}(W)$ is small whenever $H(W)$ is small.

Lemma 3. $Z_{\max}(W)^2 \leq (q-1)^2 H(W)$.

The proof follows from Proposition 4.8 of [Ş12].

3.3 Overview of the Contribution: Speed of Polarization

In order to illustrate our main contribution, which is an inequality on conditional entropies for inputs from prime alphabets, in a simple setting, we will focus on the source coding (lossless compression) model in this thesis. The consequence of our results for channel coding follows in a standard manner from the procedure outlined in Section 3.2.6, which involves the compression of sources with side information (for a more in-depth treatment of the source coding to channel coding translation, consult [Ş12, Sec 2.4])—we state the channel coding result as Theorem 9.

Suppose X is a source (random variable) over $\mathbb{Z}_q$ (with q prime), with (normalized) entropy $H(X)$ (throughout Chapter 3, by entropy we will mean the entropy normalized by a $\lg q$ factor, so that $H(X) \in [0,1]$). The source coding problem consists of compressing N i.i.d. copies $X_0, X_1, \ldots, X_{N-1}$ of X to $\approx H(X)N$ (say $(H(X)+\epsilon)N$) symbols from $\mathbb{Z}_q$. The approach based on channel polarization is to find an explicit permutation matrix $A \in \mathbb{Z}_q^{N\times N}$, such that if $(U_0, \ldots, U_{N-1})^t = A(X_0, \ldots, X_{N-1})^t$, then in the limit of $N \to \infty$, for most indices i, the conditional entropy $H(U_i|U_0, \ldots, U_{i-1})$ is either ≈ 0 or ≈ 1. Note that the conditional entropies at the source $H(X_i|X_0, \ldots, X_{i-1})$ are all equal to $H(X)$ (as the samples are i.i.d.). However, after the linear transformation by A, the conditional entropies get *polarized* to the boundaries 0 and 1. By the chain rule and conservation of entropy, the fraction of i for which $H(U_i|U_0, \ldots, U_{i-1}) \approx 1$ (resp. ≈ 0) must be $\approx H(X)$ (resp. $\approx 1 - H(X)$).

The polarization phenomenon is used to compress the X_i's as follows: The encoder only outputs U_i for indices $i \in B$ where $B = \{i \mid H(U_i|U_0, \ldots, U_{i-1}) > \zeta\}$ for some tiny $\zeta = \zeta(N) \to 0$. The decoder (decompression algorithm), called a *successive cancellation decoder*, estimates the U_i's in the order $i = 0, 1, \ldots, N-1$. For indices $i \in B$ that are output at the encoder, this is trivial, and for other positions, the decoder computes the maximum likelihood estimate $\hat{u}_i$ of U_i, assuming $U_0, \ldots, U_{i-1}$ equal $\hat{u}_0, \ldots, \hat{u}_{i-1}$, respectively. Finally, the decoder estimates the inputs at the source by applying the inverse transformation A^{-1} to $(\hat{u}_0, \ldots, \hat{u}_{N-1})^t$.

The probability of incorrect decompression (over the randomness of the source) is upper bounded, via a union bound over indices outside B, by $\sum_{i \notin B} H(U_i|U_0, \ldots, U_{i-1}) \leq$

ζN. Thus, if $\zeta \ll 1/N$, we have a reliable lossless compression scheme. Thus, in order to achieve compression rate $H(X) + \epsilon$, we need a polarizing map A for which $H(U_i|U_0, \ldots, U_{i-1}) \ll 1/N$ for at least $1 - H(X) - \epsilon$ fraction of indices. This in particular means that $H(U_i|U_0, \ldots, U_{i-1}) \approx 0$ or ≈ 1 for all but a vanishing fraction of indices, which can be compactly expressed as $\mathbf{E}_i\big[H(U_i|U_0, \ldots, U_{i-1})\big(1-H(U_i|U_0, \ldots, U_{i-1})\big)\big] \to 0$ as $n \to \infty$.

Such polarizing maps A are in fact implied by a source coding solution, and exist in abundance (a random invertible map works w.h.p.). The big novelty in Arıkan's work is an explicit recursive construction of polarizing maps, which further, due to their recursive structure, enable efficient maximum likelihood estimation of U_i given knowledge of $U_0, \ldots, U_{i-1}$.

3.4 Quantification of Polarization

Arıkan's construction is based on recursive application of the basic 2×2 invertible map (kernel) $K = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$.[3] While Arıkan's original analysis was for the binary case, the same construction based on the matrix K also works for any prime alphabet [cTA09]. Let A_n denote the matrix of the polarizing map for $N = 2^n$. In the base case $n = 1$, the outputs are $U_0 = X_0 + X_1$ and $U_1 = X_1$. If $X_0, X_1 \sim X$ are i.i.d., the entropy $H(U_0) = H(X_0 + X_1) > H(X)$ (unless $H(X) \in \{0, 1\}$), and by the chain rule $H(U_1|U_0) < H(X)$, thereby creating a small separation in the entropies. Recursively, if $(V_0, \ldots, V_{2^{n-1}-1})$ and $(T_0, \ldots, T_{2^{n-1}-1})$ are the outputs of A_{n-1} on the first half and second half of $(X_0, \ldots, X_{2^n-1})$, respectively, then the output $(U_0, \ldots, U_{2^n-1})$ satisfies $U_{2i} = V_i + T_i$ and $U_{2i+1} = T_i$.

Let H_n denote the random variable equal to $H(U_i|U_0, \ldots, U_{i-1})$ for a random $i \in \{0, 1, \ldots, 2^n - 1\}$. Arıkan's original analysis shows that the sequence $\{H_n\}$ forms a bounded martingale. Thus, the polarization property, namely that $H_n \to$ Bernoulli$(H(X))$ in the limit of $n \to \infty$, can be shown by appealing to the martingale convergence theorem. However, recall that we wish to establish the *speed of convergence*. As it turns out, in order to obtain a finite upper bound on $n(\epsilon)$, the value of n needed for $\mathbf{E}[H_n(1 - H_n)] \le \epsilon$ (so that most conditional entropies to polarize to $< \epsilon$ or $> 1 - \epsilon$), we need a more quantitative analysis.

Guruswami and Xia [GX13] propose a method of establishing convergence for binary

[3] Subsequent work established that polarization is a common phenomenon that holds for most choices of the "base" matrix instead of just K [KcU10].

polar codes that avoids use of the martingale convergence theorem by instead quantifying the increase in entropy $H(V_i+T_i|V_0,\ldots,V_{i-1},T_0,\ldots,T_{i-1})-H(V_i|V_0,\ldots,V_{i-1})$ at each stage and proving that the entropies diverge apart at a sufficient pace for H_n to polarize to $0/1$ exponentially fast in n, namely $\mathbf{E}[H_n(1-H_n)] \leq \rho^n$ for some absolute constant $\rho < 1$.

Our main technical challenge is to show an analogous entropy increase lemma for all prime alphabets. The primality assumption is necessary, because a random variable X uniformly supported on a proper subgroup has $H(X) \notin \{0,1\}$ and yet $H(X+X) = H(X)$. Formally, we prove:

Theorem 7. *Let (X_i, Y_i), $i = 1, 2$ be i.i.d. copies of a correlated random variable (X, Y) with X supported on $\mathbb{Z}_q$ for a prime q. Then for some $\alpha(q) > 0$,*

$$H(X_1 + X_2|Y_1, Y_2) - H(X|Y) \geq \alpha(q) \cdot H(X|Y)(1 - H(X|Y)). \tag{3.5}$$

The *linear* dependence of the entropy increase on the quantity $H(X|Y)(1-H(X|Y))$ is crucial to establish a speed of polarization adequate for polynomial convergence to capacity. A polynomial dependence is implicit in [Ş10], but obtaining a linear dependence requires lot more care. For the case $q = 2$, Theorem 7 is relatively easy to establish, as it is known that the extremal case (with minimal increase) occurs when $H(X|Y = y) = H(X|Y)$ for all y in the support of Y [Ş12, Lem 2.2]. This is based on the so-called "Mrs. Gerber's Lemma" for binary-input channels [WZ73, Wit74], the analog of which is not known for the non-binary case [JA14]. This allows us to reduce the binary version of (3.5) to an inequality about simple Bernoulli random variables with no conditioning, and the inequality then follows, as the sum of two p-biased coins is $2p(1-p)$-biased and has higher entropy (unless $p \in \{0, \frac{1}{2}, 1\}$). In the q-ary case, no such simple characterization of the extremal cases is known or seems likely [Ş12, Sec 4.1]. Nevertheless, we prove the inequality in the q-ary setting by first proving two inequalities for unconditioned random variables, and then handling the conditioning explicitly based on several cases. The details are in Section 3.5.

Given the entropy sumset inequality for conditional random variables, we are able to track the decay of $\sqrt{H_n(1-H_n)}$ and use Theorem 7 to show that for $N = \mathrm{poly}(1/\epsilon)$, at most $H(X) + \epsilon$ of the conditional entropies $H(U_i|U_0,\ldots,U_{i-1})$ exceed ϵ. However, to construct a good source code, we need $H(X) + \epsilon$ fraction of the conditional entropies to be $\ll 1/N$. This is achieved by augmenting a "fine" polarization stage that is analyzed using an appropriate Bhattacharyya parameter.

The efficient construction of the linear source code (i.e., figuring out which entropies polarize very close to 0 so that those symbols can be dropped), and the efficient implementation of the successive cancellation decoder are similar to the binary case [GX13] and omitted here. Upon combining these ingredients, we get the following result on lossless compression with complexity scaling polynomially in the gap to capacity:

Theorem 8. *Let X be a q-ary source for q prime with side information Y (which means (X,Y) is a correlated random variable). Let $0 < \epsilon < \frac{1}{2}$. Then there exists $N \leq (1/\epsilon)^{c(q)}$ for a constant $c(q) < \infty$ depending only on q and an explicit (constructible in $\mathrm{poly}(N)$ time) matrix $L \in \{0,1\}^{(H(X|Y)+\epsilon)N \times N}$ such that $\vec{X} = (X_0, X_1, \ldots, X_{N-1})^t$, formed by taking N i.i.d. copies $(X_0,Y_0),(X_1,Y_1),\ldots,(X_{N-1},Y_{N-1})$ of (X,Y), can, with high probability, be recovered from $L \cdot \vec{X}$ and $\vec{Y} = (Y_0, Y_1, \ldots, Y_{N-1})^t$ in $\mathrm{poly}(N)$ time.*

Moreover, can obtain Theorem 8 for *arbitrary* (not necessarily prime) q with the modification that the map $\mathbb{Z}_q^N \to \mathbb{Z}_q^{H(X|Y)+\epsilon)N}$ is no longer linear. This is obtained by factoring q into primes and combining polar codes over prime alphabets for each prime in the factorization.

Channel coding. Using known methods to construct channel codes from polar source codes for compressing sources with side information (see, for instance, [Ş12, Sec 2.4] for a nice discussion of this aspect), we obtain the following result for channel coding, enabling reliable communication at rates within an additive gap ϵ to the *symmetric capacity* for discrete memoryless channels over any fixed alphabet, with overall complexity bounded polynomially in $1/\epsilon$. Recall that a discrete memoryless channel (DMC) W has a finite input alphabet $\mathcal{X}$ and a finite output alphabet $\mathcal{Y}$ with transition probabilities $p(y|x)$ for receiving $y \in \mathcal{Y}$ when $x \in \mathcal{X}$ is transmitted on the channel. The entropy $H(W)$ of the channel is defined to be $H(X|Y)$ where X is uniform in $\mathcal{X}$ and Y is the output of W on input X; the symmetric capacity of W, which is the largest rate at which one can reliably communicate on W when the inputs have a uniform prior, equals $1 - H(W)$. Moreover, it should be noted that if W is a *symmetric* DMC, then the symmetric capacity of W is precisely the Shannon capacity of W. (See the discussion at the end of Section 2.3 as well as Definition 15.)

Theorem 9. *Let $q \geq 2$, and let W be any discrete memoryless channel capacity with input alphabet $\mathbb{Z}_q$. Then, there exists an $N \leq (1/\epsilon)^{c(q)}$ for a constant $c(q) < \infty$ depending only on q, as well as a deterministic $\mathrm{poly}(N)$ construction of a q-ary code of block length N and rate at least $1 - H(W) - \epsilon$, along with a deterministic $N \cdot \mathrm{poly}(\log N)$ time decoding algorithm for the code such that the block error probability for communication over W is at most $2^{-N^{0.49}}$. Moreover, when q is prime, the constructed codes are linear.*

3.5 Proof of Entropy Sumset Inequality

In this section, we prove Theorem 7. Our proof technique involves using an *averaging* argument to write the left-hand side of (3.5) as the expectation, over $y, z \sim Y$, of $\Delta_{y,z} = H(X_y+X_z) - \frac{H(X_y)+H(X_z)}{2}$, the entropy increase in the sum of random variables X_y and X_z with respect to their average entropy (this increase is called the *Ruzsa distance* between the random variables X_y and X_z, see [Tao10]). We then rely on inequalities for *unconditioned* random variables to obtain a lower bound for this entropy increase. In general, once needs the entropy increase to be at least $c \cdot \min\{H(X_y)(1-H(X_y)), H(X_z)(1-H(X_z))\}$, but for some cases, we actually need such an entropy increase with respect to a larger *weighted* average. Hence, we prove the stronger inequality given by Theorem 15, which shows such an increase with respect to $\frac{2H(X_y)+H(X_z)}{3}$ for $H(X_y) \geq H(X_z)$[4]. Moreover, for some cases of the proof, it suffices to bound $\Delta_{y,z}$ from below by $\frac{|H(X_y)-H(X_z)|}{2}$, which is provided by Lemma 14, another inequality for unconditional random variables.

We note a version of Theorem 7 (in fact with tight bounds) for the case of unconditioned random variables X taking values in a torsion-free group was established by Tao in his work on entropic analogs of fundamental sumset inequalities in additive combinatorics [Tao10] (results of similar flavor for integer-valued random variables were shown in [HAT14]). Theorem 7 is a result in the same spirit for groups with torsion (and which further handles conditional entropy). While we do not focus on optimizing the dependence of $\alpha(q)$ on q, pinning down the optimal dependence, especially for the case without any conditioning, seems like a natural question; see Remark 16 for further elaboration. Related but somewhat different entropic inequalities for the purpose of analyzing polar codes also appear in [ALM15].

3.5.1 Basic Entropic Lemmas and Proofs

For a random variable X taking values in $\mathbb{Z}_q$, let $H(X)$ denote the entropy of X, normalized to the interval $[0, 1]$. More formally, if p is the probability mass function of X, then

$$H(X) = \frac{1}{\lg q} \sum_{i=1}^{q} p(i) \lg(p(i))$$

[4] While the weaker inequality $H(A+B) \geq \frac{H(A)+H(B)}{2} + c \cdot \min\{H(A)(1-H(A)), H(B)(1-H(B))\}$ seems to be insufficient for our approach, it should be noted that the stronger inequality $H(A+B) \geq \max\{H(A), H(B)\} + c \cdot \min\{H(A)(1-H(A)), H(B)(1-H(B))\}$ is generally not true. Thus, Theorem 15 provides the right middle ground. A limitation of similar spirit for the entropy increase when summing two integer-valued random variables was pointed out in [HAT14].

Moreover, note for the lemmas and theorems in this section, $q \geq 2$ is an integer. We do not make any primality assumption about q anywhere in this section with the exception of Lemma 6.

Lemma 4. *If X and Y are random variables taking values in $\mathbb{Z}_q$, then*

$$H(\alpha X + (1-\alpha)Y) \geq \alpha H(X) + (1-\alpha)H(Y) + \frac{1}{2\lg q}\alpha(1-\alpha)\|X-Y\|_1^2.$$

Proof. This follows from the fact that $-H$ is a $\frac{1}{\lg q}$-strongly convex function with respect to the ℓ_1 norm on

$$\{x = (x_1, x_2, \ldots, x_q) \in \mathbb{R}^q : x_1, x_2, \ldots, x_q \geq 0, \|x\|_1 \leq 1\}$$

□

(see Example 2.5 in [Sha12] for details).

Lemma 5. *Let p be a distribution over $\mathbb{Z}_q$. Then, if $\lambda_0, \lambda_1, \ldots, \lambda_{q-1}$ are nonnegative numbers adding up to 1, we have*

$$H(\lambda_0 p^{(+0)} + \lambda_1 p^{(+1)} + \cdots + \lambda_{q-1} p^{(+(q-1))}) \geq H(p) + \frac{1}{2\lg q} \cdot \frac{\lambda_i \lambda_j}{\lambda_i + \lambda_j}\|p^{(+i)} - p^{(+j)}\|_1^2,$$

for any $i \neq j$ such that $\lambda_i + \lambda_j > 0$.

Proof. Note that if $\lambda_i + \lambda_j > 0$, then we have that by Lemma 4,

$$\begin{aligned}
H\left(\sum_{k=0}^{q-1} \lambda_k p^{(+k)}\right) &= H\left(\sum_{k \neq i,j} \lambda_k p^{(+k)} + (\lambda_i + \lambda_j)\left(\frac{\lambda_i}{\lambda_i + \lambda_j} p^{(+i)} + \frac{\lambda_j}{\lambda_i + \lambda_j} p^{(+j)}\right)\right) \\
&\geq \sum_{k \neq i.j} \lambda_k H(p^{(+k)}) + (\lambda_i + \lambda_j) H\left(\frac{\lambda_i}{\lambda_i + \lambda_j} p^{(+i)} + \frac{\lambda_j}{\lambda_i + \lambda_j} p^{(+j)}\right) \\
&= (1 - \lambda_i - \lambda_j) H(p) \\
&\quad + (\lambda_i + \lambda_j)\left(\frac{\lambda_i}{\lambda_i + \lambda_j} H(p^{(+i)}) + \frac{\lambda_j}{\lambda_i + \lambda_j} H(p^{(+j)})\right) \\
&\quad + (\lambda_i + \lambda_j) \cdot \frac{1}{2\lg q} \cdot \frac{\lambda_i}{\lambda_i + \lambda_j} \cdot \frac{\lambda_j}{\lambda_i + \lambda_j} \cdot \|p^{(+i)} - p^{(+j)}\|_1^2 \\
&= H(p) + \frac{1}{2\lg q} \cdot \frac{\lambda_i \lambda_j}{\lambda_i + \lambda_j} \cdot \|p^{(+i)} - p^{(+j)}\|_1^2,
\end{aligned}$$

as desired. □

Lemma 6. *Let p be a distribution over $\mathbb{Z}_q$, where q is prime. Then,*

$$\|p^{(+i)} - p^{(+j)}\|_1 \geq \frac{(1 - H(p)) \lg q}{2q^2(q-1) \lg e}.$$

See Lemma 4.5 of [Ş12] for a proof of the above lemma.

Lemma 7. *There exists an $\epsilon_1 > 0$ such that for any $0 < \epsilon \leq \epsilon_1$, we have*

$$-(1-\epsilon) \lg(1-\epsilon) \leq -\frac{1}{6} \epsilon \lg \epsilon.$$

Proof. By L'Hôpital's rule,

$$\lim_{\epsilon \to 0^+} \frac{(1-\epsilon)\lg(1-\epsilon)}{\epsilon \lg \epsilon} = \lim_{\epsilon \to 0^+} \frac{(1-\epsilon)\ln(1-\epsilon)}{\epsilon \ln \epsilon} = \lim_{\epsilon \to 0^+} \frac{-1 - \ln(1-\epsilon)}{1 + \ln \epsilon} = 0,$$

This implies the claim. □

Remark 10. *One can, for instance, take $\epsilon_1 = \frac{1}{500}$ in the above lemma.*

The following claim states that for sufficiently small ϵ, the quantity $\epsilon \lg \left(\frac{q-1}{\epsilon}\right)$ is close to $-\epsilon \lg \epsilon$. We omit the proof, which is rather straightforward.

Fact 11. *Let $\epsilon_2 = \frac{1}{(q-1)^4}$. Then, for any $0 < \epsilon \leq \epsilon_2$, we have*

$$\epsilon \lg \left(\frac{q-1}{\epsilon}\right) \leq \frac{5}{4} \epsilon \lg(1/\epsilon).$$

We present one final fact.

Fact 12. *The function $f(x) = x \lg(1/x)$ is increasing on the interval $(0, 1/e)$ and decreasing on the interval $(1/e, 1)$.*

Proof. The statement is a simple consequence of the fact that $f'(x) = \frac{1}{\ln 2}(-1 + \ln(1/x))$ is positive on the interval $(0, 1/e)$ and negative on the interval $(1/e, 1)$. □

Low Entropy Variables. Now, we prove lemmas that provide bounds on the entropy of a probability distribution that samples one symbol in $\mathbb{Z}_q$ with high probability, i.e., a distribution that has low entropy.

Lemma 8. *Suppose $0 < \epsilon < 1$. If p is a distribution on $\mathbb{Z}_q$ with mass $1-\epsilon$ on one symbol, then*

$$H(p) \geq \frac{\epsilon \lg(1/\epsilon)}{\lg q}.$$

Proof. Recall that the normalized entropy function H is concave. Therefore,

$$H(p) \geq H(1-\epsilon, \epsilon, \underbrace{0, 0, \ldots, 0}_{q-2}).$$

Note that

$$H(1-\epsilon, \epsilon, \underbrace{0, 0, \ldots, 0}_{q-2}) = \frac{1}{\lg q}(-(1-\epsilon)\lg(1-\epsilon) - \epsilon \lg \epsilon) \geq \frac{-\epsilon \lg \epsilon}{\lg q},$$

which establishes the claim. □

Lemma 9. *Suppose $0 < \epsilon \leq \min\{\epsilon_1, \epsilon_2\}$, where $\epsilon_1 = \frac{1}{500}$ and $\epsilon_2 = \frac{1}{(q-1)^4}$. If p is a distribution on $\mathbb{Z}_q$ with mass $1-\epsilon$ on one symbol, then*

$$H(p) \leq \frac{17\epsilon \lg(1/\epsilon)}{12 \lg q}.$$

Proof. By concavity of the normalized entropy function H, we have that

$$H(p) \leq H\left(1-\epsilon, \underbrace{\frac{\epsilon}{q-1}, \frac{\epsilon}{q-1}, \ldots, \frac{\epsilon}{q-1}}_{q-1}\right).$$

Moreover,

$$\begin{aligned} H\left(1-\epsilon, \underbrace{\frac{\epsilon}{q-1}, \ldots, \frac{\epsilon}{q-1}}_{q-1}\right) &= \frac{-(1-\epsilon)\lg(1-\epsilon) + (q-1)\cdot\left(\frac{\epsilon}{q-1}\lg\frac{q-1}{\epsilon}\right)}{\lg q} \\ &= \frac{-(1-\epsilon)\lg(1-\epsilon)}{\lg q} + \frac{\epsilon \lg\left(\frac{q-1}{\epsilon}\right)}{\lg q}. \end{aligned}$$

By Lemma 7 (and the remark following it) and Fact 11, the above quantity is bounded from above by

$$\frac{\frac{1}{6}\epsilon \lg(1/\epsilon)}{\lg q} + \frac{\frac{5}{4}\epsilon \lg(1/\epsilon)}{\lg q} = \frac{17\epsilon \lg(1/\epsilon)}{12 \lg q},$$

as desired. □

Remark 13. *Lemmas 8 and 9 show that for sufficiently small ϵ, a random variable X over $\mathbb{Z}_q$ having weight $1-\epsilon$ on a particular symbol in $\mathbb{Z}_q$ has entropy $\Theta(\epsilon \lg(1/\epsilon)/\lg q)$. This allows us to prove Lemma 10. Therefore, the constant $17/12$ in Lemma 9 is not so critical except that it is close enough to 1 for our purposes.*

Lemma 10. *Let X, Y be random variables taking values in $\mathbb{Z}_q$ such that $H(X) \geq H(Y)$, and assume $0 < \epsilon, \epsilon' \leq \min\{\epsilon_1, \epsilon_2\}$, where $\epsilon_1 = \frac{1}{500}$ and $\epsilon_2 = \frac{1}{(q-1)^4}$. Suppose that X has mass $1-\epsilon$ on one symbol, while Y has mass $1-\epsilon'$ on a symbol. Then,*

$$H(X+Y) - \frac{2H(X)+H(Y)}{3} \geq \frac{1}{51} \cdot H(Y)(1-H(Y)). \tag{3.6}$$

Overview of proof. The idea is that ϵ, ϵ' are small enough that we are able to invoke Lemmas 8 and 9. In particular, we show that $X+Y$ also has high weight on a particular symbol, which allows us to use Lemma 8 to bound $H(X+Y)$ from below. Furthermore, we use Lemma 9 in order to bound $H(X)$, $H(Y)$, and, therefore, $\frac{2H(X)+H(Y)}{3}$ from above. This gives us the necessary entropy increase for the left-hand side of 3.6. Note that the constant $1/51$ on the right-hand side of 3.6 is not of any particular importance, and we have not made any attempt to optimize the constant.

Proof. Let $j \in \mathbb{Z}_q$ such that $\Pr[X = j] = 1-\epsilon$, and let $j' \in \mathbb{Z}_q$ such that $\Pr[X = j'] = 1-\epsilon'$. Then,

$$\Pr[X+Y = j+j'] \geq (1-\epsilon)(1-\epsilon') \geq \left(\frac{499}{500}\right)^2. \tag{3.7}$$

(In a slight abuse of notation, $j+j'$ will mean $j+j' \pmod{q}$.)

Similarly, let us find an upper bound on $\Pr[X+Y = j+j']$. Let p and p' be the underlying probability distributions of X and X', respectively. Then, observe that $\Pr[X+$

$Y = j + j']$ can be bounded from above as follows:

$$\begin{aligned}
\sum_{k=0}^{q-1} p(k)p'(j+j'-k) &= p(j)p'(j') + \sum_{k\neq j} p(k)p'(j+j'-k) \\
&\leq (1-\epsilon)(1-\epsilon') + \sum_{k\neq j}\left(\frac{p(k)+p'(j+j'-k)}{2}\right)^2 \\
&\leq (1-\epsilon)(1-\epsilon') + \left(\frac{\sum_{k\neq j}(p(k)+p'(j+j'-k))}{2}\right)^2 \\
&= (1-\epsilon)(1-\epsilon') + \left(\frac{\sum_{k\neq j} p(k) + \sum_{k\neq j'} p'(k)}{2}\right)^2 \\
&= (1-\epsilon)(1-\epsilon') + \left(\frac{\epsilon+\epsilon'}{2}\right)^2 \\
&= 1 - \left(\epsilon + \epsilon' - \frac{3}{2}\epsilon\epsilon' - \frac{\epsilon^2}{4} - \frac{\epsilon'^2}{4}\right) \\
&\leq 1 - \frac{17}{18}(\epsilon+\epsilon'). \qquad (3.8)
\end{aligned}$$

Now, by Lemma 9, we have

$$H(X) \leq \frac{17\epsilon \lg(1/\epsilon)}{12 \lg q}$$

and

$$H(Y) \leq \frac{17\epsilon' \lg(1/\epsilon')}{12 \lg q}.$$

Also, by (3.7) and (3.8), we know that X has mass $1-\delta$ on a symbol, where $\frac{17}{18}(\epsilon+\epsilon') \leq$

$\delta < \frac{1}{e}$. Thus, by Lemma 8 and Fact 12, we have

$$\begin{aligned}
H(X+Y) - \frac{2H(X)+H(Y)}{3} &\geq H(X+Y) - \frac{17}{18\lg q}\epsilon \lg(1/\epsilon) - \frac{17}{36\lg q}\epsilon' \lg(1/\epsilon') \\
&\geq \frac{1}{\lg q}\left(\frac{17}{18}(\epsilon+\epsilon')\lg\left(\frac{1}{\frac{17}{18}(\epsilon+\epsilon')}\right)\right. \\
&\qquad \left. -\frac{17}{18}\epsilon\lg(1/\epsilon) - \frac{17}{36}\epsilon'\lg(1/\epsilon')\right) \\
&\geq \frac{1}{\lg q}\left(\frac{17}{18}(17\epsilon'+\epsilon')\lg\left(\frac{1}{\frac{17}{18}(17\epsilon'+\epsilon')}\right)\right. \\
&\qquad \left. -\frac{17}{18}(17\epsilon')\lg(1/17\epsilon') - \frac{17}{36}\epsilon'\lg(1/\epsilon')\right) \qquad (3.9) \\
&= \frac{1}{\lg q}\left(\frac{17}{18}\epsilon'\lg(1/17\epsilon') - \frac{17}{36}\epsilon'\lg(1/\epsilon')\right) \\
&\geq \frac{1}{36\lg q}\epsilon'\lg(1/\epsilon') \\
&\geq \frac{1}{51}H(Y)(1-H(Y)),
\end{aligned}$$

were (3.9) follows from the fact that

$$\begin{aligned}
\frac{d}{d\epsilon}\left(\frac{17}{18}(\epsilon+\epsilon')\lg\left(\frac{1}{\frac{17}{18}(\epsilon+\epsilon')}\right) - \frac{17}{18}\epsilon\lg(1/\epsilon) - \frac{17}{36}\epsilon'\lg(1/\epsilon')\right)& \\
= \frac{17}{18}\left(\lg\left(\frac{\epsilon}{\frac{17}{18}(\epsilon+\epsilon')}\right)\right)&,
\end{aligned}$$

which is negative for $\epsilon < 17\epsilon'$ and positive for $\epsilon > 17\epsilon'$. □

High Entropy Variables. For the remainder of this section, let $f(x) = -\frac{x\lg x}{\lg q}$. The following lemma proves lower and upper bounds on $f(x)$.

Lemma 11. *For $-\frac{1}{q} \leq t \leq \frac{q-1}{q}$, we have*

$$-\frac{q}{\ln q}t^2 \leq f\left(\frac{1}{q}+t\right) - \frac{1}{q} - \left(1 - \frac{1}{\ln q}\right)t \leq -\frac{q(q\ln q - (q-1))}{(q-1)^2\ln q}t^2. \qquad (3.10)$$

Proof. Let

$$g(t) = f\left(\frac{1}{q} + t\right) - \frac{1}{q} - \left(1 - \frac{1}{\ln q}\right) t + \frac{q}{\ln q} t^2.$$

To prove the lower bound in (3.10), it suffices to show that $g(t) \geq 0$ for all $-\frac{1}{q} \leq t \leq \frac{q-1}{q}$. Note that the first and second derivatives of g are

$$\begin{aligned} g'(t) &= -\frac{\ln\left(\frac{1}{q} + t\right)}{\ln q} - 1 + \frac{2qt}{\ln q} \\ g''(t) &= -\frac{1}{\left(\frac{1}{q} + t\right) \ln q} + \frac{2q}{\ln q}. \end{aligned}$$

It is clear that $g''(t)$ is an increasing function of $t \in \left(-\frac{1}{q}, \frac{q-1}{q}\right)$, and $g''(-1/2q) = 0$. Since $g'(-1/2q) = \frac{\ln 2 - 1}{\ln q} < 0$, it follows that $g(t)$ is minimized either at $t = -1/q$ or at the unique value of $t > -\frac{1}{2q}$ for which $g'(t) = 0$. Note that this latter value of t is $t = 0$, at which $g(t) = 0$. Moreover, $g(-1/q) = 0$. Thus, $g(t) \geq 0$ on the desired domain, which establishes the lower bound.

Now, let us prove the upper bound in (3.10). Define

$$h(t) = \frac{1}{q} + \left(1 - \frac{1}{\ln q}\right) t - \frac{q(q \ln q - (q-1))}{(q-1)^2 \ln q} t^2 - f\left(\frac{1}{q} + t\right).$$

Note that it suffices to show that $h(t) \geq 0$ for all $-\frac{1}{q} \leq t \leq \frac{q-1}{q}$. Observe that the first and second derivatives of h are

$$\begin{aligned} h'(t) &= 1 - \frac{2q(q \ln q - (q-1))}{(q-1)^2 \ln q} t + \frac{\ln\left(\frac{1}{q} + t\right)}{\ln q} \\ h''(t) &= -\frac{2q(q \ln q - (q-1))}{(q-1)^2 \ln q} + \frac{1}{\left(\frac{1}{q} + t\right) \ln q}. \end{aligned}$$

Now, observe that $h'(0) = 0$ and $h''(0) > 0$. Moreover, $h''(t)$ is decreasing on $t \in \left(-\frac{1}{q}, \frac{q-1}{q}\right)$. Thus, it follows that the minimum value of $h(t)$ occurs at either $t = 0$ or $t = \frac{q-1}{q}$. Since $h(0) = h\left(\frac{q-1}{q}\right) = 0$, we must have that $h(t) \geq 0$ on the desired domain, which establishes the upper bound. □

Next, we prove a lemma that provides lower and upper bounds on the entropy of a distribution that samples each symbol in $\mathbb{Z}_q$ with probability close to $\frac{1}{q}$.

Lemma 12. *Suppose p is a distribution on $\mathbb{Z}_q$ such that for each $0 \leq i \leq q-1$, we have $p(i) = \frac{1}{q} + \delta_i$ with $\max_{0 \leq i < q} |\delta_i| = \delta$. Then,*

$$1 - \frac{q^2}{\ln q}\delta^2 \leq H(p) \leq 1 - \frac{q^2(q \ln q - (q-1))}{(q-1)^3 \ln q}\delta^2.$$

Proof. Observe that $\sum_{i=0}^{q-1} \delta_i = 0$. Thus, for the lower bound on $H(p)$, note that

$$\begin{aligned} H(p) &= \sum_{i=0}^{q-1} f\left(\frac{1}{q} + \delta_i\right) \\ &\geq \sum_{i=0}^{q-1} \left(\frac{1}{q} + \left(1 - \frac{1}{\ln q}\right)\delta_i - \frac{q}{\ln q}\delta_i^2\right) \\ &= 1 - \frac{q}{\ln q}\sum_{i=0}^{q-1}\delta_i^2 \\ &\geq 1 - \frac{q^2}{\ln q}\delta^2, \end{aligned}$$

where the second line is obtained using Lemma 11, and the final line uses the fact that $|\delta_i| \leq \delta$ for all i.

Similarly, note that the upper bound on $H(p)$ can be obtained as follows:

$$\begin{aligned} H(p) &= \sum_{i=0}^{q-1} f\left(\frac{1}{q} + \delta_i\right) \\ &\leq \sum_{i=0}^{q-1} \left(\frac{1}{q} + \left(1 - \frac{1}{\ln q}\right)\delta_i - \frac{q(q \ln q - (q-1))}{(q-1)^2 \ln q}\delta_i^2\right) \\ &= 1 - \frac{q(q \ln q - (q-1))}{(q-1)^2 \ln q}\sum_{i=0}^{q-1}\delta_i^2 \\ &\leq 1 - \frac{q^2(q \ln q - (q-1))}{(q-1)^3 \ln q}\delta^2, \end{aligned}$$

where we have used the fact that

$$\sum_{i=0}^{q-1}\delta_i^2 \geq \delta^2 + (q-1)\cdot\left(\frac{\delta}{q-1}\right)^2 = \frac{q}{q-1}\delta^2.$$

□

Remark 14. *Lemma 12 shows that if p is a distribution over $\mathbb{Z}_q$ with $\max_{0\leq i<q} |p(i)-\frac{1}{q}| = \delta$, then $H(p) = 1 - \Theta_q(\delta^2)$.*

Lemma 13. *Let X and Y be random variables taking values in $\mathbb{Z}_q$ such that $H(X) \geq H(Y)$. Also, assume $0 < \delta, \delta' \leq \frac{1}{2q^2}$. Suppose $\Pr[X = i] = \frac{1}{q} + \delta_i$ and $\Pr[Y = i] = \frac{1}{q} + \delta'_i$ for $0 \leq i \leq q-1$, such that $\max_{0\leq i<q} |\delta_i| = \delta$ and $\max_{0\leq i<q} |\delta'_i| = \delta'$. Then,*

$$H(X+Y) - H(X) \geq \frac{\ln q}{16q^2} \cdot H(X)(1 - H(X)). \tag{3.11}$$

Overview of proof. We show that since X and Y sample all symbols in $\mathbb{Z}_q$ with probability close to $1/q$, it follows that $X+Y$ also samples each symbol with probability close to $1/q$. In particular, one can show that $X+Y$ samples each symbol with probability in $\left[\frac{1}{q} - \frac{\delta}{2q}, \frac{1}{q} + \frac{\delta}{2q}\right]$. Thus, we can use Lemma 12 to get a lower bound on $H(X+Y)$. Similarly, Lemma 12 also gives us an upper bound on $H(X)$. This allows us to bound the left-hand side of (3.11) adequately.

Proof. By Lemma 12, we know that

$$1 - \frac{q^2}{\ln q}\delta^2 \leq H(X) \leq 1 - \frac{q^2(q\ln q - (q-1))}{(q-1)^3 \ln q}\delta^2. \tag{3.12}$$

Note that

$$\begin{aligned}
\Pr[X+Y=k] &= \sum_{i=0}^{q-1} \Pr[X=i]\Pr[Y=k-i] \\
&= \sum_{i=0}^{q-1} \left(\frac{1}{q} + \delta_i\right)\left(\frac{1}{q} + \delta'_{k-i}\right) \\
&= \frac{1}{q} + \sum_{i=0}^{q-1} \delta_i \delta'_{k-i} \\
&\leq \frac{1}{q} + q\delta\delta' \\
&\leq \frac{1}{q} + \frac{\delta}{2q}.
\end{aligned}$$

Similarly,

$$\Pr[X+Y=k] = \frac{1}{q} + \sum_{i=0}^{q-1} \delta_i \delta_{k-i} \geq \frac{1}{q} - q\delta\delta' \geq \frac{1}{q} - \frac{\delta}{2q}\ .$$

Thus, Lemma 12 implies that

$$H(X+Y) \geq 1 - \frac{q^2}{\ln q}\left(\frac{\delta}{2q}\right)^2 = 1 - \frac{1}{4\ln q}\delta^2. \tag{3.13}$$

Therefore, by (3.12) and (3.13), we have

$$\begin{aligned} H(X+Y) - H(X) &\geq \left(1 - \frac{1}{4\ln q}\delta^2\right) - \left(1 - \frac{q^2(q\ln q - (q-1))}{(q-1)^3\ln q}\delta^2\right) \\ &= \left(\frac{q\ln q - (q-1)}{(q-1)^3} - \frac{1}{4q^2}\right)\cdot\frac{q^2}{\ln q}\delta^2 \\ &\geq \frac{\ln q}{16q^2}\cdot\frac{q^2}{\ln q}\delta^2 \\ &\geq \frac{\ln q}{16q^2}(1 - H(X)) \\ &\geq \frac{\ln q}{16q^2}H(X)(1 - H(X)), \end{aligned}$$

as desired. □

3.5.2 Unconditional Entropy Gain

We first prove some results that provide a lower bound on the normalized entropy $H(A+B)$ of a sum of random variables A, B in terms of the individual entropies.

Lemma 14. *Let A and B be random variables taking values over $\mathbb{Z}_q$. Then,*

$$H(A+B) \geq \max\{H(A), H(B)\}.$$

Proof. Without loss of generality, assume $H(A) \geq H(B)$. Let p be the underlying probability distribution for A. Let $\lambda_i = \Pr[B = i]$. Then, the underlying probability distribution of $A+B$ is $\lambda_0 p^{(+0)} + \lambda_1 p^{(+1)} + \cdots + \lambda_{q-1}p^{(+(q-1))}$. The desired result then follows directly from Lemma 5. □

The next theorem provides a different lower bound for $H(A+B)$.

Theorem 15. *Let A and B be random variables taking values over $\mathbb{Z}_q$ such that $H(A) \geq H(B)$. Then,*

$$H(A+B) \geq \frac{2H(A)+H(B)}{3} + c\cdot\min\{H(A)(1-H(A)), H(B)(1-H(B))\}$$

for $c = \frac{\gamma_0^3 \lg q}{48q^5(q-1)^3\lg(6/\gamma_0)\lg^2 e}$, where $\gamma_0 = \frac{1}{500(q-1)^4\lg q}$.

Overview of proof. The proof of the Theorem 15 splits into various cases depending on where $H(A)$ and $H(B)$ lie. Note that some of these cases overlap. The overall idea is as follows. If $H(A)$ and $H(B)$ are both bounded away from 0 and 1 (Case 2), then the desired inequality follows from the concavity of the entropy function, using Lemmas 5 and 6 (note that this uses primality of q). Another setting in which the inequality can be readily proven is when $H(A) - H(B)$ is bounded away from 0 (which we deal with in Cases 4 and 5).

Thus, the remaining cases occur when $H(A)$ and $H(B)$ are either both small (Case 1) or both large (Case 3). In the former case, one can show that A must have most of its weight on a particular symbol, and similarly for B (note that this is why we must choose $\gamma_0 \ll \frac{1}{\log q}$; otherwise, A could be, for instance, supported uniformly on a set of size 2). Then, one can use the fact that a q-ary random variable having weight $1 - \epsilon$ has entropy $\Theta(\epsilon \log(1/\epsilon))$ (Lemmas 8 and 9) in order to prove the desired inequality (using Lemma 10).

For the latter case, we simply show that each of the q symbols of A must have weight close to $1/q$, and similarly for B. Then, we use the fact that such a random variable whose maximum deviation from $1/q$ is δ has entropy $1-\Theta(\delta^2)$ (Lemma 12) in order to prove the desired result (using Lemma 13).

Proof. Let γ_0 be as defined in the theorem statement. Note that we must have at least one of the following cases:

1. $0 \leq H(A), H(B) \leq \gamma_0$.
2. $\frac{\gamma_0}{2} \leq H(A), H(B) \leq 1 - \frac{\gamma_0}{2}$.
3. $1 - \gamma_0 \leq H(A), H(B) \leq 1$.
4. $H(A) > \gamma_0$ and $H(B) < \frac{\gamma_0}{2}$.
5. $H(A) > 1 - \frac{\gamma_0}{2}$ and $H(B) < 1 - \gamma_0$.

We treat each case separately. For ease of notation, we write

$$M = \min\{H(A)(1 - H(A)), H(B)(1 - H(B))\}.$$

Case 1. Let $\max_{0 \leq j < q} \Pr[A = j] = 1 - \epsilon$, where $\epsilon \leq \frac{q-1}{q}$. Note that if $\epsilon \geq \frac{1}{e}$, then Fact 12 implies that

$$\begin{aligned} H(A) &\geq -\frac{(1-\epsilon)\lg(1-\epsilon)}{\lg q} \\ &\geq \frac{1}{\lg q} \cdot \min\left\{-\frac{1}{q}\lg\left(\frac{1}{q}\right), -\left(1-\frac{1}{e}\right)\lg\left(1-\frac{1}{e}\right)\right\} \\ &> \gamma_0, \end{aligned}$$

which is a contradiction. Thus, $\epsilon < \frac{1}{e}$.

Now, simply note that if $\epsilon > \gamma_0 \lg q$, then Lemma 8 and Fact 12 would imply that

$$H(A) \geq \frac{\epsilon \lg(1/\epsilon)}{\lg q} > \gamma_0,$$

a contradiction. Hence, we must have $\epsilon \leq \gamma_0 \lg q$. Similarly, we can write $\max_{0 \leq j < q} \Pr[B = j] = 1 - \epsilon'$ for some positive $\epsilon' \leq \gamma_0 \lg q$. Then, Lemma 10 implies that

$$H(A+B) \geq \frac{2H(A)+H(B)}{3} + \frac{1}{51}H(B)(1-H(B)),$$

as desired.

Case 2. Let p be the underlying probability distribution for A, and let $\lambda_i = \Pr[B = i]$. Then, the underlying probability distribution of $A+B$ is $\lambda_0 p^{(+0)} + \lambda_1 p^{(+1)} + \cdots + \lambda_{q-1} p^{(+(q-1))}$. Let $(i_0, i_1, \ldots, i_{q-1})$ be a permutation of $(0, 1, \ldots, q-1)$ such that $\lambda_{i_0} \geq \lambda_{i_1} \geq \cdots \geq \lambda_{i_{q-1}}$.

Since $\lambda_0 + \lambda_1 + \cdots + \lambda_{q-1} = 1$ and $\max_{0 \leq j \leq q-1} \lambda_j = \lambda_{i_0}$, we have

$$\lambda_{i_0} \geq \frac{1}{q}. \tag{3.14}$$

Next, let $\epsilon_0 = \frac{\gamma_0}{6\lg(6/\gamma_0)}$. we claim that

$$\lambda_{i_1} > \frac{\epsilon_0}{q-1}. \tag{3.15}$$

Suppose not, for the sake of contradiction. Then, $\lambda_{i_1}, \lambda_{i_2}, \ldots, \lambda_{i_{q-1}} \leq \frac{\epsilon_0}{q-1}$, which implies that $\lambda_{i_0} = 1 - \sum_{j=1}^{q-1} \lambda_{i_j} \geq 1 - \epsilon_0$. Since $\epsilon_0 \leq \min\left\{\frac{1}{e}, \frac{1}{500}, \frac{1}{(q-1)^4}\right\}$, Lemma 9 and Fact 12 imply that

$$H(B) \leq \frac{17\epsilon_0 \lg(1/\epsilon_0)}{12\lg q},$$

which is less than $\frac{\gamma_0}{2}$, resulting in a contradiction. Thus, (3.15) is true.

Therefore, by Lemma 5 and Lemma 6,

$$\begin{aligned} H(A+B) &= H(\lambda_0 p^{(+0)} + \lambda_1 p^{(+1)} + \cdots + \lambda_{q-1} p^{(+(q-1))}) \\ &\geq H(A) + \frac{1}{2\lg q} \cdot \frac{\lambda_{i_0}\lambda_{i_1}}{\lambda_{i_0}+\lambda_{i_1}} \|p^{(+i_0)} - p^{(+i_1)}\|_1^2 \\ &\geq H(A) + \frac{1}{2\lg q}\lambda_{i_0}\lambda_{i_1}\|p^{(+i_0)} - p^{(+i_1)}\|^2 \\ &\geq H(A) + \frac{\lambda_{i_0}\lambda_{i_1}(1-H(p))^2 \lg q}{8q^4(q-1)^2\lg^2 e} \\ &= H(A) + \frac{\lambda_{i_0}\lambda_{i_1}\gamma_0^2 \lg q}{32q^4(q-1)^2\lg^2 e} \\ &\geq \frac{2H(A)+H(B)}{3} + \frac{\epsilon_0\gamma_0^2\lg q}{32q^5(q-1)^3\lg^2 e}. \end{aligned}$$

Finally, note that $M \leq \frac{1}{4}$, which implies that

$$\frac{\epsilon_0\gamma_0^2\lg q}{32q^5(q-1)^3\lg^2 e} \geq \frac{\epsilon_0\gamma_0^2\lg q}{8q^5(q-1)^3\lg^2 e}M.$$

Therefore,

$$H(A+B) \geq \frac{2H(A)+H(B)}{3} + cM,$$

where $c = \frac{\gamma_0^3 \lg q}{48q^5(q-1)^3\lg(6/\gamma_0)\lg^2 e}$.

Case 3. Let $\Pr[A=i] = \frac{1}{q}+\delta_i$ for $0 \leq i \leq q-1$. If $\delta = \max_{0\leq i<q}|\delta_i|$, then by Lemma 12, we have

$$1-\gamma_0 \leq H(A) \leq 1 - \frac{q^2(q\ln q-(q-1))}{(q-1)^3\ln q}\delta^2,$$

which implies that

$$\delta \leq \sqrt{\frac{\gamma_0(q-1)^3\ln q}{q^2(q\ln q-(q-1))}} < \frac{1}{2q^2}.$$

Similarly, if we let $\Pr[B=i] = \frac{1}{q}+\delta'_i$ for all i, and $\delta' = \max_{0\leq i<q}|\delta'_i|$, then

$$\delta' \leq \sqrt{\frac{\gamma_0(q-1)^3\ln q}{q^2(q\ln q-(q-1))}} < \frac{1}{2q^2}.$$

Thus, by Lemma 13, we see that

$$\begin{aligned}H(A+B) &\geq H(A) + \frac{\ln q}{16q^2} \cdot H(A)(1-H(A)) \\ &\geq \frac{2H(A)+H(B)}{3} + \frac{\ln q}{16q^2}M,\end{aligned}$$

as desired.

Case 4. Note that by Lemma 14,

$$\begin{aligned}H(A+B) - \frac{2H(A)+H(B)}{3} &\geq H(A) - \frac{2H(A)+H(B)}{3} \\ &= \frac{H(A)-H(B)}{3} \\ &\geq \frac{\gamma_0}{6} \\ &\geq \frac{1}{3}H(B)(1-H(B)).\end{aligned}$$

Case 5. As in Case 4, we have that

$$H(A+B) - \frac{2H(A)+H(B)}{3} \geq \frac{\gamma_0}{6}.$$

However, this time, the above quantity is bounded from below by $\frac{1}{3}H(A)(1-H(A))$, which completes this case. □

3.5.3 Conditional Entropy Gain

Theorem 7 now follows as a simple consequence of our main theorem, which we restate and prove below.

Theorem 7. *Let (X_i, Y_i), $i = 1, 2$ be i.i.d. copies of a correlated random variable (X, Y) with X supported on $\mathbb{Z}_q$ for a prime q. Then for some $\alpha(q) > 0$,*

$$H(X_1 + X_2 | Y_1, Y_2) - H(X|Y) \geq \alpha(q) \cdot H(X|Y)(1 - H(X|Y)). \tag{3.5}$$

Remark 16. *We have not attempted to optimize the dependence of $\alpha(q)$ on q, and our proof gets $\alpha(q) \geq \frac{1}{q^{O(1)}}$. It is easy to see that $\alpha(q) \leq O(1/\log q)$ even without conditioning (i.e., when $Y = 0$). Understanding what is the true behavior of $\alpha(q)$ seems like an interesting and basic question about sums of random variables. For random variables X taking values from a torsion-free group G and with sufficiently large $H_2(X)$, it is known that $H_2(X_1 + X_2) - H_2(X) \geq \frac{1}{2} - o(1)$ and that this is best possible [Tao10], where $H_2(\cdot)$ denotes the* unnormalized *entropy (in bits). When G is the group of integers, a lower bound $H_2(X_1 + X_2) - H_2(X) \geq g(H_2(X))$ for an increasing function $g(\cdot)$ was shown for all $\mathbb{Z}$-valued random variables X [HAT14]. For groups G with torsion, we cannot hope for any entropy increase unless G is finite and isomorphic to $\mathbb{Z}_q$ for q prime (as G cannot have non-trivial finite subgroups), and we cannot hope for an absolute entropy increase even for $\mathbb{Z}_q$. So determining the asymptotics of $\alpha(q)$ as a function of q is the analog of the question studied in [Tao10] for finite groups.*

Overview of proof. Let X_y denote $X|Y = y$. Then, we use an averaging argument: We reduce the desired inequality to providing a lower bound for $\Delta_{y,z} = H(X_y + X_z) - \frac{H(X_y)+H(X_z)}{2}$, whose expectation over $y, z \sim Y$ is the left-hand side of (3.5). Then, one splits into three cases for small, large, and medium values of $H(X|Y)$.

Thus, we reduce the problem to aruguing about unconditional entropies. As a first step, one would expect to prove $\Delta_{y,z} \geq \min\{H(X_y)(1-H(X_y)), H(X_z)(1-H(X_z))\}$ and use this in the proof of the conditional inequality. However, this inequality turns out to be too weak to deal with the case in which $H(X|Y)$ is tiny (case 2). This is the reason we require Theorem 15, which provides an increase for $H(X_y + X_z)$ over a higher *weighted* average instead of the simple average of $H(X_y)$ and $H(X_z)$. Additionally, we use the inequality $H(X_y + X_z) \geq \max\{H(X_y), H(X_z)\}$ to handle certain cases, and this is provided by Lemma 14.

In cases 1 and 3 (for $H(X|Y)$ in the middle and high regimes), the proof idea is that either (1) there is a significant mass of $(y, z) \sim Y \times Y$ for which $H(X_y)$ and $H(X_z)$ are separated, in which case one can use Lemma 14 to bound $\mathbf{E}[\Delta_{y,z}]$ from below, or (2) there is a significant mass of $y \sim Y$ for which $H(X_y)$ lies away from 0 and 1, in which case $H(X_y)(1 - H(X_y))$ can be bounded from below, enabling us to use Theorem 15.

Proof. Let $h = H(X|Y)$, and let c be the constants defined in the statement of Theorem 15. Moreover, let $\gamma_1 = 1/20$ and let

$$p = \Pr_y \left[H(X_y) \in \left(\frac{\gamma_1}{2}, 1 - \frac{\gamma_1}{2}\right)\right].$$

Also, let X_y denote $X|Y = y$, and let

$$\Delta_{y,z} = H(X_y + X_z) - \frac{H(X_y) + H(X_z)}{2}.$$

Note that Lemma 14 implies that $\Delta_{y,z} \geq 0$ for all y, z. Also, $\mathbf{E}_{y \sim Y, z \sim Y}[\Delta_{y,z}] = H(X_1 + X_2|Y_1, Y_2) - H(X|Y)$. For simplicity, we will often omit the subscript and write $\mathbf{E}[\Delta_{y,z}]$.

We split into three cases, depending on the value of h.

Case 1: $h \in (\gamma_1, 1 - \gamma_1)$.

- **Subcase 1**: $p \geq \frac{\gamma_1}{4}$. Note that if $H(X_y) \in \left(\frac{\gamma_1}{2}, 1 - \frac{\gamma_1}{2}\right)$, then $H(X_y)(1 - H(X_y)) \geq \frac{\gamma_1}{2}\left(1 - \frac{\gamma_1}{2}\right)$. Hence, by Theorem 15, we have

$$\begin{aligned}
\mathbf{E}[\Delta_{y,z}] \geq & \sum_{\substack{y,z \\ \frac{\gamma_1}{2} < H(X_y), H(X_z) < 1 - \frac{\gamma_1}{2}}} \Pr[Y = y] \cdot \Pr[Y = z] \\
& \cdot \left(H(X_y + X_z) - \frac{2\max\{H(X_y), H(X_z)\} + \min\{H(X_y), H(X_z)\}}{3} \right) \\
\geq & \sum_{\substack{y,z \\ \frac{\gamma_1}{2} < H(X_y), H(X_z) < 1 - \frac{\gamma_1}{2}}} \Pr[Y = y] \cdot \Pr[Y = z] \cdot c \\
& \cdot \min\{H(X_y)(1 - H(X_y)), H(X_z)(1 - H(X_z))\} \\
\geq & \frac{c\gamma_1}{2}\left(1 - \frac{\gamma_1}{2}\right) \sum_{\substack{y,z \\ \frac{\gamma_1}{2} < H(X_y), H(X_z) < 1 - \frac{\gamma_1}{2}}} \Pr[Y = y] \cdot \Pr[Y = z] \\
= & cp^2 \cdot \frac{\gamma_1}{2}\left(1 - \frac{\gamma_1}{2}\right) \\
\geq & \frac{c\gamma_1^3}{32}\left(1 - \frac{\gamma_1}{2}\right) \\
\geq & \frac{c\gamma_1^3}{8}\left(1 - \frac{\gamma_1}{2}\right) \cdot h(1 - h).
\end{aligned}$$

- **Subcase 2**: $p < \frac{\gamma_1}{4}$. Note that

$$\begin{aligned}
\gamma_1 &< h \\
&\leq \Pr_y\left[H(X_y) \leq \frac{\gamma_1}{2}\right] \cdot \frac{\gamma_1}{2} + \Pr_y\left[H(X_y) > \frac{\gamma_1}{2}\right] \cdot 1 \\
&\leq \frac{\gamma_1}{2} + \Pr_y\left[H(X_y) > \frac{\gamma_1}{2}\right]
\end{aligned}$$

which implies that

$$\Pr_y\left[H(X_y) > \frac{\gamma_1}{2}\right] \geq \frac{\gamma_1}{2}.$$

Thus,

$$\begin{aligned}\Pr_y\left[H(X_y) \geq 1 - \frac{\gamma_1}{2}\right] &= \Pr_y\left[H(X_y) > \frac{\gamma_1}{2}\right] - \Pr_y\left[\frac{\gamma_1}{2} < H(X_y) < 1 - \frac{\gamma_1}{2}\right] \\ &\geq \frac{\gamma_1}{2} - p \\ &> \frac{\gamma_1}{4}. \end{aligned} \tag{3.16}$$

Also,

$$1 - \gamma_1 > h \geq \left(1 - \frac{\gamma_1}{2}\right) \cdot \Pr_y\left[H(X_y) \geq 1 - \frac{\gamma_1}{2}\right],$$

which implies that

$$\Pr_y\left[H(X_y) \geq 1 - \frac{\gamma_1}{2}\right] < \frac{1-\gamma_1}{1-\frac{\gamma_1}{2}}.$$

Hence,

$$\begin{aligned}\Pr_y\left[H(X_y) \leq \frac{\gamma_1}{2}\right] &= 1 - \Pr_y\left[\frac{\gamma_1}{2} < H(X_y) < 1 - \frac{\gamma_1}{2}\right] - \Pr_y\left[H(X_y) \geq 1 - \frac{\gamma_1}{2}\right] \\ &> 1 - p - \frac{1-\gamma_1}{1-\frac{\gamma_1}{2}} \\ &> 1 - \frac{\gamma_1}{4} - \frac{1-\gamma_1}{1-\frac{\gamma_1}{2}} \\ &\geq \frac{\gamma_1}{4}. \end{aligned} \tag{3.17}$$

Using Lemma 14 along with (3.16) and (3.17), we now conclude that

$$\begin{aligned}\mathbf{E}[\Delta_{y,z}] &\geq \sum_{\substack{y,z \\ H(X_y) \geq 1 - \frac{\gamma_1}{2} \\ H(X_z) \leq \frac{\gamma_1}{2}}} \Pr[Y = y] \cdot \Pr[Z = z] \cdot \left|\frac{H(X_y) - H(X_z)}{2}\right| \\ &\geq \frac{\gamma_1}{4} \cdot \frac{\gamma_1}{4} \cdot \frac{1-\gamma_1}{2} \\ &\geq \frac{\gamma_1^2(1-\gamma_1)}{8} \cdot h(1-h), \end{aligned}$$

as desired.

Case 2: $h \leq \gamma_1$. Then, define $S = \left\{y : H(X_y) > \frac{4}{5}\right\}$. We split into two subcases.

- **Subcase 1**: $\sum_{y \in S} \Pr[Y = y] \cdot H(X_y) \geq \frac{2h}{3}$. Then, $\Pr[Y \in S] \geq \frac{2h}{3}$, and so, by Lemma 14, we have

$$\begin{aligned}
\mathbf{E}_{y,z}[H(X_y + X_z)] - h &\geq \Pr_{\substack{y,z \\ \{y,z\} \cap S \neq \emptyset}} \Pr[Y = y] \cdot \Pr[Y = z] \cdot \max\{H(X_y), H(X_z)\} \\
&\quad - h \\
&\geq \frac{4}{5}(2 \cdot \Pr[Y \in S] - \Pr[Y \in S]^2) - h \\
&\geq \frac{4}{5}\left(2 \cdot \frac{2h}{3} - \left(\frac{2h}{3}\right)^2\right) - h \\
&= \frac{1}{15} h \left(1 - \frac{16}{3} h\right) \\
&\geq \frac{1}{15}\left(1 - \frac{16\gamma_1}{3}\right) h(1-h).
\end{aligned}$$

- **Subcase 2**: $\sum_{y \in S} \Pr[Y = y] \cdot H(X_y) < \frac{2h}{3}$. Then,

$$\sum_{y \notin S} \Pr[Y = y] \cdot H(X_y) > \frac{h}{3}. \tag{3.18}$$

Moreover, observe that $h \geq \frac{4}{5} \cdot \Pr[Y \in S]$, implying that

$$\Pr[Y \notin S] \geq 1 - \frac{5h}{4}. \tag{3.19}$$

Hence, using Theorem 15 as well as (3.18) and (3.19), we find that

$$\begin{aligned}\mathbf{E}[\Delta_{y,z}] \geq \sum_{y,z \notin S} &\Pr[Y=y] \cdot \Pr[Y=z] \cdot \Bigg(\frac{2\max\{H(X_y),H(X_z)\} + \min\{H(X_y),H(X_z)\}}{3} \\ &+ c \cdot \min\{H(X_y)(1-H(X_y)), H(X_z)(1-H(X_z))\} - \frac{H(X_y)+H(X_z)}{2} \Bigg) \\ \geq \sum_{y,z \notin S} &\Pr[Y=y] \cdot \Pr[Y=z] \left(\left| \frac{H(X_y)-H(X_z)}{6} \right| + \frac{c}{5} \cdot \min\{H(X_y),H(X_z)\} \right) \\ \geq \sum_{y,z \notin S} &\Pr[Y=y] \cdot \Pr[Y=z] \cdot \left(\frac{H(X_y)}{6} - \left(\frac{1}{6} - \frac{c}{5} \right) H(X_z) \right) \\ = \frac{c}{5} &\Pr[Y \notin S] \cdot \sum_{y \notin S} \Pr[Y=y] \cdot H(X_y) \\ > \frac{c}{5} &\left(1 - \frac{5h}{4}\right) \cdot \frac{h}{3} \geq c \left(\frac{1}{15} - \frac{\gamma_1}{12} \right) h(1-h),\end{aligned}$$

as desired.

Case 3: $h \geq 1 - \gamma_1$. Write $\gamma = 1 - h$, and let

$$S = \left\{ y : H(X_y) > 1 - \frac{\gamma}{2} \right\}.$$

Moreover, let $\overline{S}$ be the complement of S. We split into two subcases.

1. **Subcase 1**: $\Pr_y[y \in S] < \frac{1}{10}$. Then, letting $r = \Pr_y \left[H(X_y) \leq \frac{1}{10} \right]$, we see that

$$\begin{aligned} h &= 1 - \gamma \\ &= \sum_{\substack{y \\ H(X_y) \leq \frac{1}{10}}} \Pr[Y=y] \cdot H(X_y) + \sum_{\substack{y \\ H(X_y) > \frac{1}{10}}} \Pr[Y=y] \cdot H(X_y) \\ &\leq \frac{1}{10} \cdot \Pr_y \left[H(X_y) \leq \frac{1}{10} \right] + 1 \cdot \Pr_y \left[H(X_y) > \frac{1}{10} \right] \\ &= \frac{r}{10} + (1-r), \end{aligned}$$

which implies that $r \leq \frac{10}{9}\gamma \leq \frac{10}{9}\gamma_1$. Hence, letting $T = \left\{ y : \frac{1}{10} \leq H(X_y) \leq 1 - \frac{\gamma}{2} \right\}$, we see that

$$\Pr_y[y \in T] \geq 1 - \frac{1}{10} - r \geq \frac{9}{10} - \frac{10}{9}\gamma_1 \geq \frac{1}{2}. \tag{3.20}$$

Hence, by Theorem 15 and (3.20),

$$\begin{aligned}
\mathbf{E}[\Delta_{y,z}] &\geq \sum_{y,z \in T} \Pr[Y=y] \cdot \Pr[Y=z] \cdot \Delta_{y,z} \\
&\geq \sum_{y,z \in T} \Pr[Y=y] \cdot \Pr[Y=z] \\
&\quad \cdot (c \cdot \min\{H(X_y)(1-H(X_y)), H(X_z)(1-H(X_z))\}) \\
&\geq (\Pr[Y \in T])^2 \left(c \cdot \frac{\gamma}{2}\left(1-\frac{\gamma}{2}\right)\right) \\
&\geq \frac{c}{8}\gamma\left(1-\frac{\gamma}{2}\right) \\
&\geq \frac{c}{8}h(1-h).
\end{aligned}$$

2. **Subcase 2**: $\Pr_y[y \in S] \geq \frac{1}{10}$. Then, observe that by Lemma 14,

$$\begin{aligned}
\mathbf{E}[\Delta_{y,z}] &\geq \sum_{\substack{y \in S \\ z \in \overline{S}}} \frac{\Pr[Y=y] \cdot \Pr[Y=z] \cdot (H(X_y)-H(X_z))}{2} \\
&= \frac{\Pr[Y \in \overline{S}] \cdot \sum_{y \in S} \Pr[Y=y] \cdot H(X_y)}{2} \\
&\quad - \frac{\Pr[Y \in S] \cdot \sum_{y \in \overline{S}} \Pr[Y=y] \cdot H(X_y)}{2} \\
&= \frac{\sum_{y \in S} \Pr[Y=y] H(X_y)}{2} - \frac{(1-\gamma)\Pr[Y \in S]}{2} \\
&\geq \frac{\left(1-\frac{\gamma}{2}\right)\Pr[Y \in S] - (1-\gamma)\Pr[Y \in S]}{2} \\
&\geq \frac{\gamma}{4} \cdot \Pr[Y \in S] \geq \frac{\gamma}{40} \geq \frac{1}{40}h(1-h).
\end{aligned}$$

□

3.6 Rough Polarization

Now that we have established Theorem 7, we are ready to show rough polarization of the channels $W_n^{(i)}$, $0 \leq i < 2^n$, for large enough n. The precise theorem showing rough polarization is as follows.

Theorem 17. *There is a constant $\Lambda < 1$ such that the following holds. For any $\Lambda < \rho < 1$, there exists a constant b_ρ such that for all channels W with q-ary input, all $\epsilon > 0$, and all $n > b_\rho \lg(1/\epsilon)$, there exists a set*

$$\mathcal{W}' \subseteq \{W_n^{(i)} : 0 \leq i \leq 2^n - 1\}$$

such that for all $M \in \mathcal{W}'$, we have $Z_{\max}(M) \leq 2\rho^n$ and $\Pr_i[W_n^{(i)} \in \mathcal{W}'] \geq 1 - H(W) - \epsilon$.

The proof of Theorem 17 follows from the following lemma:

Lemma 15. *Let $T(W) = H(W)(1 - H(W))$ denote the* symmetric entropy *of a channel W. Then, there exists a constant $\Lambda < 1$ (possibly dependent on q) such that*

$$\frac{1}{2}\left(\sqrt{T\left(W_{n+1}^{(2j)}\right)} + \sqrt{T\left(W_{n+1}^{(2j+1)}\right)}\right) \leq \Lambda\sqrt{T\left(W_n^{(j)}\right)} \tag{3.21}$$

for any $0 \leq j < 2^n$.

Proof of Lemma 15: Fix a $0 \leq j < 2^n$. Also, let $h = H(W_n^{(j)})$, and let $\delta = H((W_n^{(j)})^-) - H(W_n^{(j)}) = H(W_n^{(j)}) - H((W_n^{(j)})^+)$. Then, letting $t = \sqrt{T(W_{n+1}^{(2j)})} + \sqrt{T(W_{n+1}^{(2j+1)})}$, we note that

$$t = \sqrt{h(1-h) + (1-2h)\delta - \delta^2} + \sqrt{h(1-h) - (1-2h)\delta - \delta^2}. \tag{3.22}$$

For ease of notation, let $f : [-1, 1] \to \mathbb{R}$ be the function given by

$$f(x) = \sqrt{h(1-h) + x} + \sqrt{h(1-h) - x}.$$

By symmetry, we may assume that $h \leq \frac{1}{2}$ without loss of generality. Moreover, if we let $\alpha = \alpha(q)$ be the constant described in Theorem 7, then we know that $\delta \geq \alpha h(1-h)$. Then, since $f'''(x) \leq 0$ for $0 \leq x \leq h(1-h)$, Taylor's Theorem implies that

$$\begin{aligned}
t &\leq f((1-2h)\delta) \\
&\leq f(0) + f'(0)((1-2h)\delta) + \frac{f''(0)}{2}((1-2h)\delta)^2 \\
&= 2\sqrt{h(1-h)} - \frac{((1-2h)\delta)^2}{4(h(1-h))^{3/2}} \\
&\leq 2\sqrt{h(1-h)} - \frac{(\alpha h(1-h)(1-2h))^2}{4(h(1-h))^{3/2}} \\
&= 2\sqrt{h(1-h)} - \frac{\alpha^2}{4}(1-2h)^2\sqrt{h(1-h)}.
\end{aligned}$$

Thus, if $1 - 2h \geq \frac{\alpha}{8+\alpha}$, then the desired result follows for $\Lambda \geq 1 - \frac{1}{2}\left(\frac{\alpha^2}{16+2\alpha}\right)^2$.

Next, consider the case in which $1 - 2h < \frac{\alpha}{8+\alpha}$. Then, $\frac{4}{8+\alpha} < h \leq \frac{1}{2}$. Hence, $\delta \geq \alpha h(1-h) \geq \frac{2\alpha}{8+\alpha}$, which implies that $\delta \geq 2(1-2h)$. It follows that

$$(1-2h)\delta - \delta^2 \leq -\frac{\delta^2}{2}.$$

Hence, by plugging this into (3.22), we have that

$$\frac{1}{2}t \leq \sqrt{h(1-h) - \frac{\delta^2}{2}}.$$

Now, recall that $\delta \geq \frac{2\alpha}{8+\alpha}$, a constant bounded away from 0. Moreover, if c is a positive constant, then $\frac{\sqrt{x-c}}{\sqrt{x}}$ is an increasing function of x for $x > c$. Since $h(1-h) \leq \frac{1}{4}$, it follows that

$$\begin{aligned}\frac{\frac{1}{2}t}{T(W_n^{(j)})} &\leq \frac{\sqrt{h(1-h) - \frac{\delta^2}{2}}}{\sqrt{h(1-h)}} \\ &\leq \frac{\sqrt{\frac{1}{4} - \frac{\delta^2}{2}}}{\sqrt{\frac{1}{4}}} \\ &\leq \sqrt{1 - \frac{8\alpha^2}{(8+\alpha)^2}}.\end{aligned}$$

We conclude that the desired statement holds for

$$\Lambda = \max\left\{1 - \frac{1}{2}\left(\frac{\alpha^2}{16+2\alpha}\right)^2, \sqrt{1 - \frac{8\alpha^2}{(8+\alpha)^2}}\right\}.$$

□

Now, we are ready to prove Theorem 17:

Proof of Theorem 17. For any $\rho \in (0,1)$, let

$$\begin{aligned}A_\rho^l &= \left\{i : H(W_n^{(i)}) \leq \frac{1 - \sqrt{1-4\rho^n}}{2}\right\} \\ A_\rho^u &= \left\{i : H(W_n^{(i)}) \geq \frac{1 + \sqrt{1-4\rho^n}}{2}\right\} \\ A_\rho &= A_\rho^l \cup A_\rho^u.\end{aligned}$$

Moreover, note that repeated application of (3.21), we have

$$\mathbf{E}_i\sqrt{T(W_n^{(i)})} \le \Lambda^n\sqrt{T(W)} \le \frac{\Lambda^n}{2}.$$

Thus, by Markov's inequality,

$$\Pr_i[T(W_n^{(i)}) \ge \alpha] \le \frac{\Lambda^n}{2\sqrt{\alpha}} \tag{3.23}$$

Then, observe that

$$\begin{aligned} H(W) &= \mathbf{E}_i\left[H(W_n^{(i)})\right] \\ &\ge \Pr[A_\rho^l] \cdot \min_{i \in A_\rho^l} H(W_n^{(i)}) + \Pr[A_\rho^u] \cdot \min_{i \in A_\rho^u} H(W_n^{(i)}) \\ &\quad + \Pr[\overline{A_\rho}] \cdot \min_{i \in \overline{A_\rho}} H(W_n^{(i)}) \\ &\ge \Pr[A_\rho^u] \cdot (1 - 2\rho^n). \end{aligned} \tag{3.24}$$

Therefore, letting $t = \Pr_i\left[H(W_n^{(i)}) \le 2\rho^n\right]$, we have

$$\begin{aligned} t &\ge \Pr[A_\rho^l] \\ &= 1 - \Pr[A_\rho^u] - \Pr[\overline{A_\rho}] \\ &\ge 1 - H(W) - \Pr[A_\rho^u] \cdot 2\rho^n - \Pr[\overline{A_\rho}] & (3.25) \\ &\ge 1 - H(W) - 2\rho^n - \frac{1}{2}(\Lambda/\sqrt{\rho})^n, & (3.26) \end{aligned}$$

where (3.25) follows from (3.24), and (3.26) follows from (3.23). Thus, it is clear that if $\rho > \Lambda^2$, then there exists a constant a_ρ such that for $n > a_\rho \lg(1/\epsilon)$, we have

$$t \ge 1 - H(W) - \epsilon.$$

To conclude, note that Lemma 3 implies

$$\begin{aligned} \Pr_i\left[Z_{\max}(W_n^{(i)}) \le 2\rho^n\right] &\ge \Pr_i\left[H(W_n^{(i)}) \le \frac{4\rho^{2n}}{(q-1)^2}\right] \\ &\ge \Pr_i\left[H(W_n^{(i)}) \le 2\left(\frac{\rho^2}{(q-1)^2}\right)^n\right] \\ &\ge 1 - H(W) - \epsilon \end{aligned}$$

for $n > b_\rho \lg(1/\epsilon)$, where $b_\rho = a_{\rho^2/(q-1)^2}$. □

Note that the proofs of Theorem 17 and Lemma 15 follow from arguments similar to those found in the proofs of Proposition 5 and Lemma 8 in [GX13], except that we work with $Z_{\max}$.

3.7 Fine Polarization

Now, we describe the statement of "fine polarization." This is quantified by the following theorem.

Theorem 18. *For any $0 < \delta < \frac{1}{2}$, there exists a constant c_δ that satisfies the following statement: For any q-ary input memoryless channel W and $0 < \epsilon < \frac{1}{2}$, if $n_0 > c_\delta \lg(1/\epsilon)$, then*

$$\Pr_i \left[Z_{\max}(W_{n_0}^{(i)}) \le 2^{-2^{\delta n_0}} \right] \ge 1 - H(W) - \epsilon.$$

The proof follows from arguments similar to those in [AT09, GX13]. For the sake of completeness, and because there are some slight differences in the behavior of the q-ary Bhattacharyya parameters from Section 3.2.7 compared to the binary case, we present a proof here.

Proof. Let $\rho \in (\Lambda^2, 1)$ be a fixed constant, where Λ is the constant described in Theorem 17, and choose $\gamma > \lg(1/\rho)$ such that $\beta = \left(1 + \frac{1}{\gamma}\right)\delta < \frac{1}{2}$. Then, let us set $m = \left\lfloor \frac{n_0}{1+\gamma} \right\rfloor$ and $n = \left\lceil \frac{\gamma n_0}{1+\gamma} \right\rceil$, so that $n_0 = m + n$. Moreover, let $d = \left\lfloor \frac{12 n \lg q}{m \lg(1/\rho)} \right\rfloor$ and choose a constant $a_\rho > 0$ such that

$$a_\rho > \frac{12(\ln 2)(\lg q)}{(1-2\beta)^2 \lg(1/\rho)} \left(1 + \lg\left(\frac{48\gamma \lg q}{\lg(1/\rho)}\right)\right).$$

Now, letting

$$t = \max\left\{2b_\rho \lg(2/\epsilon), \frac{24 \lg(1/\beta) \lg q}{\beta \lg(1/\rho)}, 2a_\rho \lg(2/\epsilon), 1, \frac{1}{\gamma}\right\},$$

we choose

$$n_0 > (1+\gamma)t, \tag{3.27}$$

where b_ρ is the constant described in Theorem 17. Note that this guarantees that

$$m > \max\left\{b_\rho \lg(2/\epsilon), \frac{12 \lg(1/\beta) \lg q}{\beta \lg(1/\rho)}, a_\rho \lg(2/\epsilon)\right\}. \tag{3.28}$$

Then, Theorem 17 implies that there exists a set

$$\mathcal{W}' \subseteq \{W_m^{(i)} : 0 \le i \le 2^m - 1\}$$

such that for all $M \in \mathcal{W}'$, we have $Z_{\max}(M) \le 2\rho^m$ and

$$\Pr_i[W_m^{(i)} \in \mathcal{W}'] \ge 1 - H(W) - \frac{\epsilon}{2}. \tag{3.29}$$

Let T be the set of indices i for which $W_m^{(i)} \in \mathcal{W}'$.

Fix an arbitrary $M \in \mathcal{W}'$. Recursively define $\left\{\tilde{Z}_k^{(i)}\right\}_{0 \le i \le 2^k - 1}$ by $\tilde{Z}_0^{(0)} = Z_{\max}(M)$ and

$$\tilde{Z}_{k+1}^{(i)} = \begin{cases} \left(\tilde{Z}_k^{\lfloor i/2 \rfloor}\right)^2, & i \equiv 1 \pmod 2 \\ q^3 \tilde{Z}_k^{\lfloor i/2 \rfloor}, & i \equiv 0 \pmod 2 \end{cases}.$$

Now, let us define the sets $G_j(n) \subseteq \{i \in \mathbb{Z} : 0 \le i \le 2^n - 1\}$, for $j = 0, 1, \ldots, d-1$ as follows:

$$G_j(n) = \left\{ i : \sum_{\frac{jn}{d} \le k < \frac{(j+1)n}{d}} i_k \ge \beta n/d \right\},$$

where $\overline{i_{n-1} i_{n-2} \cdots i_0}$ is the binary representation of i. Also, let $G(n) = \bigcap_{0 \le j < d} G_j(n)$. Note that if we choose i uniformly among $0, 1, \ldots, 2^n - 1$, then $i_0, i_1, \ldots, i_{n-1}$ are i.i.d. Bernoulli random variables. Thus, Hoeffding's inequality implies that

$$\Pr_{0 \le i < 2^n}[i \in G_j(n)] \ge 1 - \exp(-(1-2\beta)^2 n/2d)$$

for every j. Hence, by the union bound,

$$\Pr_{0 \le i < 2^n}[i \in G(n)] \ge 1 - d\exp(-(1-2\beta)^2 n/2d). \tag{3.30}$$

Now, assume $i \in G(n)$. Note that $\tilde{Z}_{(j+1)n/d}^{\left(\lfloor i/2^{n(d-j-1)/d} \rfloor\right)}$ can be obtained by taking $\tilde{Z}_{jn/d}^{\left(\lfloor i/2^{n(d-j)/d} \rfloor\right)}$ and performing a sequence of n/d operations, each of which is either $z \mapsto z^2$ (squaring) or $z \mapsto q^3 z$ (q^3-fold increase). Since $i \in G_j(n)$, at least $\beta n/d$ of the operations must be squarings. Hence, it is not too difficult to see that the maximum possible value of $\tilde{Z}_{(j+1)n/d}^{\left(\lfloor i/2^{n(d-j-1)/d} \rfloor\right)}$ is obtained when we have $(1-\beta)n/d$ q^3-fold increases followed by $\beta n/d$ squarings. Hence,

$$\lg \tilde{Z}_{(j+1)n/d}^{\left(\lfloor i/2^{n(d-j-1)/d} \rfloor\right)} \le 2^{\beta n/d} \left(\frac{n}{d}(1-\beta)(3\lg q) + \lg \tilde{Z}_{jn/d}^{\left(\lfloor i/2^{n(d-j)/d} \rfloor\right)} \right).$$

Making repeated use of the above inequality, we see that

$$\begin{aligned}
\lg Z(M_n^{(i)}) &\le \lg \tilde{Z}_n^{(i)} \\
&\le 2^{\beta n} \lg Z_{\max}(M) + \frac{n}{d}(1-\beta)(3\lg q)\sum_{k=1}^{d} 2^{\frac{k\beta n}{d}} \\
&\le 2^{\beta n} \lg Z_{\max}(M) + \frac{n}{d}(3\lg q)\frac{(1-\beta)2^{\beta n}}{1-2^{-\frac{\beta n}{d}}} \\
&\le 2^{\beta n}\left(\lg(2\rho^m) + \frac{n}{d}(3\lg q)\right) \qquad (3.31) \\
&\le -2^{\beta n}, \qquad (3.32)
\end{aligned}$$

where (3.31) follows from (3.28) and

$$\begin{aligned}
2^{-\frac{n}{d}\beta} &\le 2^{-\frac{\beta m \lg(1/\rho)}{12\lg q}} \\
&\le \beta,
\end{aligned}$$

while (3.32) follows from (3.28) and

$$\begin{aligned}
\lg(2\rho^m) + \frac{n}{d}(3\lg q) &\le \lg(2\rho^m) + \frac{3n\lg q}{\frac{6n\lg q}{m\lg(1/\rho)}} \\
&\le 1 - m\lg(1/\rho) + \frac{m\lg(1/\rho)}{2} \\
&= 1 - \frac{m\lg(1/\rho)}{2} \\
&\le -1.
\end{aligned}$$

Therefore, for any $0 \le k < 2^{n_0}$ that can be written as $k = 2^n i' + i$, for $0 \le i' < 2^m$ and $0 \le i < 2^n$ such that $i' \in T$ and $i \in G(n)$, we have that for $M = W_m^{(i')}$,

$$\lg Z_{\max}(W_{n_0}^{(k)}) = \lg Z_{\max}(M_n^{(i)}) \le -2^{\beta n} \le -2^{\delta n_0}.$$

Moreover, by (3.29), (3.30), and the union bound, we see that the probability that a uniformly chosen $0 \le k < 2^{n_0}$ is of the above form is at least $1 - H(W) - \frac{\epsilon}{2} - de^{-\frac{(1-2\beta)^2 n}{2d}}$, which is

$$\begin{aligned}
&\geq 1 - H(W) - \frac{\epsilon}{2} - \frac{12n \lg q}{m \lg(1/\rho)} \exp\left(\frac{(1-2\beta)^2 m \lg \rho}{12 \lg q}\right) \\
&\geq 1 - H(W) - \frac{\epsilon}{2} - \frac{48\gamma \lg q}{\lg(1/\rho)} \exp\left(-\frac{(1-2\beta)^2 m \lg(1/\rho)}{12 \lg q}\right) \\
&\geq 1 - H(W) - \frac{\epsilon}{2} - \frac{48\gamma \lg q}{\lg(1/\rho)} \left(\frac{\epsilon}{2}\right)^{\frac{a_\rho (1-2\beta)^2 \lg(1/\rho)}{12(\ln 2)(\lg q)}} \\
&\geq 1 - H(W) - \frac{\epsilon}{2} - \frac{48\gamma \lg q}{\lg(1/\rho)} \left(\frac{\epsilon}{2}\right)^{1+\lg\left(\frac{48\gamma \lg q}{\lg(1/\rho)}\right)} \\
&\geq 1 - H(W) - \frac{\epsilon}{2} - \frac{48\gamma \lg q}{\lg(1/\rho)} \cdot \frac{\epsilon}{2} \cdot \left(\frac{1}{2}\right)^{\lg\left(\frac{48\gamma \lg q}{\lg(1/\rho)}\right)} \\
&= 1 - H(W) - \epsilon.
\end{aligned}$$

So if we take $c_\delta = \max\left\{4(1+\gamma)a_\rho, 4(1+\gamma)b_\rho, 1+\gamma, \frac{1+\gamma}{\gamma}, \frac{24(1+\gamma)\lg(1/\beta)\lg q}{\beta \lg(1/\rho)}\right\}$, then $n_0 > c_\delta \lg(1/\epsilon)$ would guarantee (3.27). This completes the proof. □

As a corollary, we obtain the following result on lossless compression with complexity scaling polynomially in the gap to capacity:

Theorem 8. *Let X be a q-ary source for q prime with side information Y (which means (X, Y) is a correlated random variable). Let $0 < \epsilon < \frac{1}{2}$. Then there exists $N \leq (1/\epsilon)^{c(q)}$ for a constant $c(q) < \infty$ depending only on q and an explicit (constructible in $\mathrm{poly}(N)$ time) matrix $L \in \{0,1\}^{(H(X|Y)+\epsilon)N \times N}$ such that $\vec{X} = (X_0, X_1, \ldots, X_{N-1})^t$, formed by taking N i.i.d. copies $(X_0, Y_0), (X_1, Y_1), \ldots, (X_{N-1}, Y_{N-1})$ of (X, Y), can, with high probability, be recovered from $L \cdot \vec{X}$ and $\vec{Y} = (Y_0, Y_1, \ldots, Y_{N-1})^t$ in $\mathrm{poly}(N)$ time.*

Proof. Let $W = (X; Y)$, and fix $\delta = 0.499$. Also, let $N = 2^{n_0}$. Then, by Theorem 18, for any $n_0 > c_\delta \lg(1/\epsilon)$, we have that

$$\Pr_i \left[Z_{\max}(W_{n_0}^{(i)}) \leq 2^{-2^{\delta n_0}} \right] \geq 1 - H(X) - \epsilon.$$

Moreover, let $N = 2^{n_0}$. Recall the notation in (3.3). Then, letting $\delta' = 2^{-2^{\delta n_0}}$, we have that $\Pr_i[i \in \mathsf{High}_{n_0,\delta'}] \leq H(X|Y) + \epsilon$ and $Z(W_{n_0}^{(i)}) \geq \delta'$ for all $i \in \mathsf{High}_{n_0,\delta'}$. Thus, we can take L to be the linear map G_{n_0} projected onto the coordinates of $\mathsf{High}_{n_0,\delta'}$.

By Lemma 1 and the union bound, the probability that attempting to recover $\vec{X}$ from $L \cdot \vec{X}$ and $\vec{Y}$ results in an error is given by

$$\begin{aligned}\sum_{i \notin \mathsf{High}_{n_0,\delta'}} P_e(W_{n_0}^{(i)}) &\leq \sum_{i \notin \mathsf{High}_{n_0,\delta'}} (q-1) Z_{\max}(W_{n_0}^{(i)}) \\ &\leq (q-1)N\delta' \\ &= (q-1)2^{n_0 - 2^{\delta n_0}}, \end{aligned} \tag{3.33}$$

which is $\leq 2^{-N^{0.49}}$ for $N \geq (1/\epsilon)^{\mu}$ for some positive constant μ (possibly depending on q). Hence, it suffices to take $c(q) = 1 + \max\{c_\delta, \mu\}$.

Finally, the fact that both the construction of L and the recovery of $\vec{X}$ from $L \cdot \vec{X}$ and $\vec{Y}$ can be done in $\mathrm{poly}(N)$ time follows in a similar fashion to the binary case (see the binning algorithm and the successive cancellation decoder in [GX13] for details). Also, the entries of L are all in $\{0, 1\}$ since L can be obtained by taking a submatrix of $B_n K^{\otimes n_0}$, where B_n is a permutation matrix, and $K = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$ (see [Arı10]). □

3.8 Extension to Arbitrary Alphabets

In the previous sections, we have shown polarization and polynomial gap to capacity for polar codes over *prime* alphabets. We now describe how to extend this to obtain channel polarization and the explicit construction of a polar code with polynomial gap to capacity over *arbitrary* alphabets.

The idea is to use the multi-level code construction technique sketched in [cTA09] (and also recently in [LA14] for alphabets of size 2^m). We outline the procedure here. Suppose we have a channel $W = (X; Y)$, where $X \in \mathbb{Z}_q$ and $Y \in \mathcal{Y}$. Moreover, assume that $q = \prod_{i=1}^{s} q_i$ is the prime factorization of q.

Now, we can write $X = (U^{(1)}, U^{(2)}, \ldots, U^{(s)})$, where each $U^{(i)}$ is a random variable distributed over $[q_i]$. We also define the channels $W^{(1)}, W^{(2)}, \ldots, W^{(s)}$ by $W^{(j)} = (U^{(j)}; Y, U^{(1)}, U^{(2)}, \ldots, U^{(j-1)})$. Note that

$$\begin{aligned} H(W) = H(X|Y) &= H(U^{(1)}, U^{(2)}, \ldots, U^{(s)}|Y) \\ &= \sum_{j=1}^{s} H(U^{(j)}|Y, U^{(1)}, U^{(2)}, \ldots, U^{(j-1)}) \\ &= \sum_{j=1}^{s} H(W^{(j)}), \end{aligned}$$

which means that W splits into $W^{(1)}, W^{(2)}, \ldots, W^{(s)}$. Since each $W^{(j)}$ is a channel whose input is over a prime alphabet, one can polarize each $W^{(j)}$ separately using the procedure of the previous sections. More precisely, the encoding procedure is as follows. For N large enough (as specified by Theorem 8), we take N copies $(X_0; Y_0), \ldots, (X_{N-1}; Y_{N-1})$ of W, where $X_i = (U_i^{(1)}, U_i^{(2)}, \ldots U_i^{(s)})$. Then, sequentially for $j = 1, 2, \ldots, s$, we encode $U_0^{(j)}, U_1^{(j)}, \ldots, U_{N-1}^{(j)}$ using $\left\{\left(Y_i, U_i^{(1)}, U_i^{(2)}, \ldots, U_i^{(j-1)}\right)\right\}_{i=0,1,\ldots,N-1}$ as side information (which can be done by the procedure in previous sections, since U_j is a source over a prime alphabet).

For decoding, one can simply use s stages of the successive cancellation decoder. In the j^{th} stage, one uses the successive cancellation decoder for $W^{(j)}$ in order to decode $U_0^{(j)}, U_1^{(j)}, \ldots, U_{N-1}^{(j)}$, assuming that $\left\{U_i^{(k)}\right\}_{k<j}$ has been recovered correctly from the previous stages of successive canellation decoding. Note that the error probability in decoding $X_0, X_1, \ldots, X_{N-1}$ can be obtained by taking a union bound over the error probabilities for each of the s stages of successive cancellation decoding. Since each individual error probability is exponentially small (see (3.33)), it follows that the overall error probability is also negligible.

As a consequence, we obtain Theorem 8 for non-prime q, with the additional modification that the map $\mathbb{Z}_q^N \to \mathbb{Z}_q^{H(X|Y)+\epsilon)N}$ is not linear. Moreover, using the translation from source coding to noisy channel coding (see [Ş12, Sec 2.4]), we obtain the following result for channel coding.

Theorem 9. *Let $q \geq 2$, and let W be any discrete memoryless channel capacity with input alphabet $\mathbb{Z}_q$. Then, there exists an $N \leq (1/\epsilon)^{c(q)}$ for a constant $c(q) < \infty$ depending only on q, as well as a deterministic $\operatorname{poly}(N)$ construction of a q-ary code of block length N and rate at least $1 - H(W) - \epsilon$, along with a deterministic $N \cdot \operatorname{poly}(\log N)$ time decoding algorithm for the code such that the block error probability for communication over W is at most $2^{-N^{0.49}}$. Moreover, when q is prime, the constructed codes are linear.*

Remark 19. *If q is prime, then the q-ary code of Theorem 9 is, in fact, linear.*

Chapter 4

Coding for Interactive Communication

The results of this chapter appear in [HV16].

4.1 Background

The work of Shannon and Hamming applies to the problem of *one-way communication*, in which one party, say Alice, wishes to send a message to another party, say Bob. However, in many applications, underlying (two-party) communications are *interactive*, i.e., Bob's response to Alice may be based on what he received from her previously and vice versa. As in the case of one-way communication, one wishes to make such interactive communications robust to noise by adding some redundancy.

At first sight, it seems plausible that one could use error-correcting codes to encode each round of communication separately. However, this does not work correctly because the channel might corrupt the codeword of one such round of communication entirely and as a result derail the entire future conversation. With the naive approach being insufficient, it is not obvious whether it is possible at all to encode interactive protocols in a way that can tolerate some small constant fraction of errors in an interactive setting. Nonetheless, Schulman [Sch92, Sch93, Sch96] showed that this is possible and numerous follow-up works over the past several years have led to a drastically better understanding of error-correcting coding schemes for interactive communications.

4.1.1 Error Fractions for Interactive Coding

Schulman was the first to consider the question of coding for interactive communication and showed that one can tolerate an adversarial error fraction of $\epsilon = 1/240$ with an unspecified constant communication rate [Sch92, Sch93, Sch96]. Schulman's result also implies that for the easier setting of random errors, one can tolerate any error rate bounded away from $1/2$ by repeating symbols multiple times. Since Schulman's seminal work, there has been a number of subsequent works pinning down the tolerable error fraction. For instance, Braverman and Rao [BR14] showed that any error fraction $\epsilon < 1/4$ can be tolerated in the realm of adversarial errors, provided that one can use larger alphabet sizes, and this bound was shown to be optimal. A series of subsequent works [BE14, GH14, GHS14, EGH15, FGOS15] worked to determine the error rate region under which non-zero communication rates can be obtained for a variety of models, e.g., adversarial errors, random errors, list decoding, adaptivity, and channels with feedback. Unlike the initial coding schemes of [Sch96] and [BR14] that relied on tree codes and as a result required exponential time computations, many of the newer coding schemes are computationally efficient [BK12, BN13, BKN14, GMS14, GH14]. All these results achieve small often unspecified constant communication rate of $\Theta(1)$ which is fixed and independent of amount of noise. Only the works of Kol and Raz [KR13] and Haeupler [Hae14] achieve a communication rate approaching 1 for error fractions going to zero.

4.1.2 Communication Rates of Interactive Coding Schemes

Only recently, however, has this study led to results shedding light on the tradeoff between the achievable communication rate for a given error fraction or amount of noise.

Kol and Raz [KR13] gave a communication scheme for random errors that achieves a communication rate of $1-O(\sqrt{H(\epsilon)})$ for any alternating protocol, where $\epsilon > 0$ is the error rate. They also developed powerful tools to prove upper bounds on the communication rate. Haeupler [Hae14] showed communication schemes that achieve a communication rate of $1-O(\sqrt{\epsilon})$ for any oblivious adversarial channel, including random errors, as well as a communication rate of $1-O(\sqrt{\epsilon \log\log(1/\epsilon)})$ for any fully adaptive adversarial channel. These results apply to alternating protocols as well as adaptively simulated non-alternating protocols (see [Hae14] for a more detailed discussions). Lastly, Haeupler conjectured these rates to be optimal for their respective settings.

On the other hand, in the one-way communication setting, the classical result of Shannon shows that the capacity of a binary symmetric channel with error rate ϵ is $1 - H(\epsilon)$.

Furthermore, for the case of *adversarial* errors, in which an adversary is allowed to introduce any error pattern of up to an error fraction of ϵ, the capacity is known to be $1 - \Theta(H(\epsilon))$ (as suggested by the Gilbert-Varshamov and Hamming bounds). Therefore, there is an almost quadratic gap between the conjectured rate achievable in the interactive setting and the $1 - \Theta(H(\epsilon))$ rate known to be optimal for one-way communications.

4.2 Overview of Results: Capacity of Interactive Communication Channels for Low Error Rates

While the result of [Hae14] is somewhat disappointing in that the (conjectured to be optimal) communication rates are worse than the $1 - \Theta(H(\epsilon))$ rate achievable for non-interactive communication, it does leave open some interesting questions. In particular, the hardest protocols to encode under the underlying coding schemes of Haeupler seem to be "maximally interactive" protocols, which we discuss below. However, most protocols that are likely to show up in real-world applications seem to be far from the worst-case "maximally interactive" case. This leaves open the possibility for some assumptions on the input protocol that would allow coding schemes with better rates:

Question 20. *Is there a reasonable set of assumptions under which a two party protocol can be encoded into a longer protocol that is resilient to an ϵ error fraction of fully adversarial errors with a communication rate of $1 - O(\epsilon \log(1/\epsilon))$?*

Another shortcoming of [Hae14] is that while the coding schemes are rather simple and elegant, they have virtually nothing in common with error-correcting codes and techniques for non-interactive communication that have been developed over the past several decades. This is true for other interactive coding schemes from past works as well, where seemingly disparate methods have been used across several works. More specifically, the early works in the field [Sch96, BR14] used the combinatorial object of *tree codes* to construct coding schemes, while latter works [GHS14, GH14, Hae14] that obtain efficient schemes have used no such objects and are much simpler. Explicit efficient constructions of tree codes have thus far eluded researchers; on the other hand, tree codes are a nice clean combinatorial object that appears to be a natural analogue to the sphere packing interpretation of normal error-correcting codes. Thus, we consider the following goal.

Objective 21. *Find a unifying mathematical theory for coding of (two-way) interactive protocols that relates to coding theory for one-way communication.*

In this thesis, we address both Question 20 and Objective 21. In particular, we show that for a natural and large class of protocols the conjectured capacity gap between the one-way and interactive communication settings disappears. Our primary focus is on protocols for *oblivious adversarial* channels. Such a channel can corrupt any ϵ fraction of bits that are exchanged in the execution of a protocol, and the simulation is required to work, with high probability, for any such error pattern. This is significantly stronger, more interesting, and, as we will see, also much more challenging than the case of independent random errors. We remark that, in contrast to a *fully adaptive adversarial* channel, the decision whether an error happens in a given round is not allowed to depend on the transcript of the execution thus far. This seems to be a minor but crucially necessary restriction (see also Section 4.6).

As mentioned, the conjectured optimal communication rate of $1-O(\sqrt{\epsilon})$ for the oblivious adversarial setting is worse than the $1 - O(H(\epsilon))$ communication rate achievable in the one-way communication settings. However, the conjectured upper bound seems to be tight mainly for "maximally interactive" protocols, i.e., protocols in which the party that is sending bits changes frequently. In particular, *alternating* protocols, in which Alice and Bob take turns sending a single bit, seem to require the most redundancy for a noise-resilient encoding. On the other hand, the usual one-way communication case in which one party just sends a single message consisting of several bits is an example of a "minimally interactive" protocol. It is a natural question to consider what the tradeoff is between achievable communication rate and the level of interaction that takes place. In particular, most natural real-world protocols are rarely "maximally interactive" and could potentially be simulated with communication rates going well beyond $1 - O(\sqrt{\epsilon})$. We seek to investigate this possibility.

Our first contribution is to introduce the notion of *average message length* as a natural measure of the interactivity of a protocol in the context of analyzing communication rates. Loosely speaking, the average message length of an n-round protocol corresponds to the average number of bits a party sends before receiving a reply from the other party. A lower average message length roughly corresponds to more interactivity in a protocol, e.g., a maximally interactive protocol has average message length 1, while a one-way protocol with no interactivity has average message length n. The formal definition of average message length appears as Definition 18 in Section 4.4.

Our second and main contribution in this chapter is to show that for protocols with an average message length of at least some constant in ϵ (but independent of the number of rounds n) one can go well beyond the $1-\Theta(\sqrt{\epsilon})$ communication rate achieved by [Hae14] for channels with oblivious adversarial errors. In fact, we show that for such protocols one can actually achieve a communication rate of $1 - \Theta(H(\epsilon))$, matching the communication rate for one-way communication up to the (unknown) constant in the $H(\epsilon)$ term.

Theorem 22. *For any $\epsilon > 0$ and any n-round interactive protocol Π with average message length $\ell = \Omega(\mathrm{poly}(1/\epsilon))$, it is possible to encode Π into a protocol over the same alphabet which, with probability at least $1 - \exp(-n\epsilon^6)$, simulates Π over an oblivious adversarial channel with an ϵ fraction of errors while achieving a communication rate of $1 - \Theta(H(\epsilon)) = 1 - \Theta(\epsilon \log(1/\epsilon))$.*

Under the (simplifying) assumption of *public shared randomness*, our protocol can furthermore be seen to have the nice property of being *rateless*. This means that the communication rate adapts automatically and only depends on the actual error rate ϵ without having to specify or know in advance what amount of noise to prepare for.

Theorem 23. *Suppose Alice and Bob have access to public shared randomness. For any $\epsilon' > 0$ and any n-round interactive protocol Π with average message length $\ell = \Omega(\mathrm{poly}(1/\epsilon'))$, it is possible to encode Π into protocol Π_{rateless} over the same alphabet such that for any* true error rate *ϵ, executing Π_{rateless} for $n(1 + O(H(\epsilon)) + O(\epsilon' \,\mathrm{polylog}(1/\epsilon')))$ rounds simulates Π with probability at least $1 - \exp(-n\epsilon'^3)$.*

We note that one should think of ϵ' in Theorem 23 as chosen to be very small, in particular, smaller than the smallest amount of noise one expects to encounter. In this case, the communication rate of the protocol simplifies to the optimal $1-O(H(\epsilon))$ for essentially any $\epsilon > \epsilon'$. The only reason for not choosing ϵ' too small is that it very slightly increases the failure probability. As an example, choosing $\epsilon' = o(1)$ suffices to get ratelessness for any constant ϵ and still leads to an essentially exponential failure probability. Alternatively, one can even set $\epsilon' = n^{-1/6}$ which leads to optimal communication rates even for tiny sub-constant true error fractions $\epsilon > n^{-0.2}$ while still achieving a strong sub-exponential failure probability of at most $\exp(-\sqrt{n})$.

4.3 Preliminaries

An *interactive protocol* Π consists of communication performed by two parties, Alice and Bob, over a channel with alphabet Σ. Alice has an input x and Bob has an input y, and the protocol consists of n *rounds*. During each round of a protocol, each party decides whether to listen or transmit a symbol from Σ, based on his input and the player's *transcript* thus far. Alice's *transcript* is defined as a tuple of symbols from Σ, one for each round that has occurred, such that the i^{th} symbol is either (a.) the symbol that Alice sent during the i^{th} round, if she chose to transmit, or (b.) the symbol that Alice received, otherwise.

Moreover, protocols can utilize *randomness*. In the case of *private randomness*, each party is given its own infinite string of independent uniformly random bits as part of its input. In the case of *shared randomness*, both parties have access to a common infinite random string during each round. In general, our protocols will utilize private randomness, unless otherwise specified.

In a *noiseless* setting, we can assume that in any round, exactly one party speaks and one party listens. In this case, the listening party simply receives the symbol sent by the speaking party.

The *communication order* of a protocol refers to the order in which Alice and Bob choose to speak or listen. A protocol is *non-adaptive* if the communication order is fixed prior to the start of the protocol, in which case, whether a party transmits or listens depends only on the round number. A simple type of non-adaptive protocol is an *alternating* protocol, in which one party transmits during odd numbered rounds, while the other party transmits during even numbered rounds. On the other hand, an *adaptive* protocol is one in which the communication order is not fixed prior to the start; therefore, the communication order can vary depending on the transcript of the protocol. In particular, each party's decision whether to speak or listen during a round will depend on his input, randomness, as well as the transcript of the protocol thus far.

For an n-round protocol over alphabet Σ, one can define an associated *protocol tree* of depth n. The protocol tree is a rooted tree in which each non-leaf node of the tree has $|\Sigma|$ children, and the outgoing edges are labeled by the elements of Σ. Each non-leaf node is owned by some player, and the owner of the node has a *preferred* edge that emanates from the node. The preferred edge is a function of the owner's input and any randomness that is allowed. Also, leaf nodes of the protocol tree correspond to ending states.

A proper execution of the protocol corresponds to the unique path from the root of the protocol tree to a leaf node, such that each traversed edge is the preferred edge of the parent node of the edge. In this case, each edge along the path can be viewed as a successive round in which the owner of the parent node transmits the symbol along the edge.

An example of a protocol tree is shown in Figure 4.1.

4.3.1 Communication Channels

For our purposes, the communication between the two parties occurs over a *communication channel* that delivers a possibly corrupted version of the symbol transmitted by the sending party. In this thesis, transmissions will be from a *binary* alphabet, i.e., $\Sigma = \{0, 1\}$.

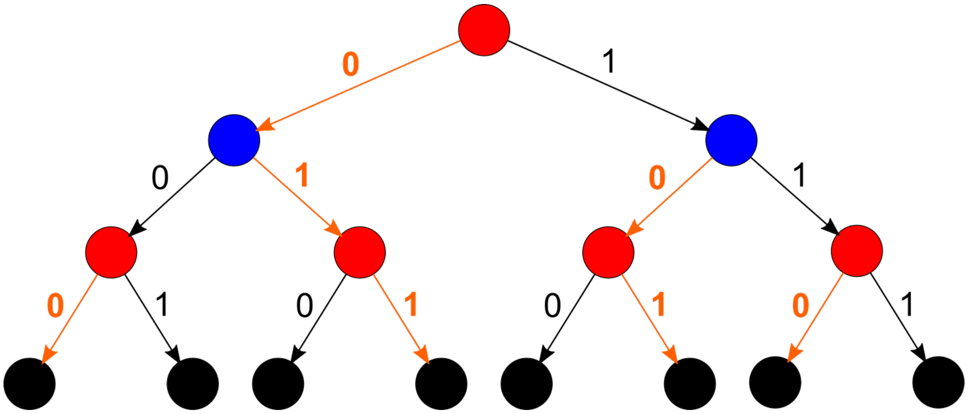

Figure 4.1: An example of a protocol tree for a 3-round interactive protocol. Nodes owned by Alice are colored red, while those owned by Bob are colored blue. Note that Alice always speaks during the first and third rounds, while Bob speaks during the second round. The orange edges are the set of preferred edges for some choice of inputs of Alice and Bob. In this case, a proper execution of the protocol corresponds to the path "011."

In a *random error channel*, each transmission occurs over a binary symmetric channel with crossover probability ϵ. In other words, in each round, if only one party is speaking, then the transmitted bit gets corrupted with probability ϵ.

In this thesis, we mainly consider the *oblivious adversarial channel*, in which an adversary gets to corrupt at most ϵ fraction of the total number of rounds. However, the adversary is restricted to making his decisions prior to the start of the protocol, i.e., the adversary must decide which rounds to corrupt independently of the communication history and randomness used by Alice and Bob. For each round that the adversary decides to corrupt, he can either commit a *flip* error or *replace* error. Suppose a round has one party that speaks and one party that listens. Then, a flip error means that the listening party receives the opposite of the bit that the transmitting party sends. On the other hand, a replace error requires the adversary to specify a symbol $\alpha \in \Sigma$ for the round. In this case, the listening party receives α regardless of which symbol was sent by the transmitting party.

An adaptive adversarial channel allows an adversary to corrupt at most ϵ fraction of the total number of rounds. However, in this case, the adversary does not have to commit to which rounds to corrupt prior to the start of the protocol. Rather, the adversary can decide to corrupt a round based on the communication history thus far, including what is being sent in the current round. Thus, in any round that the adversary chooses to corrupt in which one party transmits and one party receives, the adversary can make the listening party receive any symbol of his choice.

Note that we have not yet specified the behavior for rounds in which both parties speak or both parties listen. Such rounds can occur for *adaptive* protocols when the communication occurs over a noisy communication channel.

If both parties speak during a round, we stipulate that neither party receives any symbol during that round (since neither party is expecting to receive a symbol).

Moreover, we stipulate that in rounds during which both parties listen, the symbols received by Alice and Bob are unspecified. In other words, an arbitrary symbol may be delivered to each of the parties, and we require that the protocol work for any choice of received symbols. Alternatively, one can imagine that the adversary chooses arbitrary symbols for Alice and Bob to receive without this being counted as a corruption (i.e., a free corruption that is not counted toward the budget of ϵ fraction of corruptions). The reason for this model is to disallow the possibility of transmitting information by using silence. An extensive discussion on the appropriateness of this error model can be found in [GHS14].

4.4 Average Message Length and Blocked Protocols

One conceptual contribution of this thesis is to introduce the notion of *average message length* as a natural measure of the level of interactivity of a protocol. While we use it only in the context of analyzing the optimal rate of interactive coding schemes, we believe that this notion and parametrization will also be useful in other settings, such as compression. Next, we define this notion formally.

Definition 18. *The* average message length ℓ *of an n-round interactive protocol Π is the minimum, over all paths in the protocol tree of Π, of the average length in bits of a maximal contiguous block (spoken by a single party) down the path.*

More precisely, given any string $s \in \{0,1\}^n$, there exist integer message lengths $l_0, \dots, l_k > 0$ such that along the path of Π given by s one player (either Alice or Bob) speaks between round $1 + \sum_{j<i} l_j$ and round $\sum_{j \leq i} l_j$ for even i while the other speaks during the remaining intervals, i.e., those for odd i. We then define ℓ_s to be the average of these message lengths $l_0, \dots, l_k$ and define the average message length *of Π to be minimum over all possible inputs, i.e., $\ell = \min_{s \in \{0,1\}^n} \ell_s$.*

An alternate characterization of the amount of interaction in a protocol involves the number of alternations in the protocol:

Definition 19. *An n-round protocol Π is said to be k-*alternating *if any path in the protocol tree of Π can be divided into at most k blocks of consecutive rounds such that only one person (either Alice or Bob) speaks during each block.*

*More precisely, Π is k-*alternating *if, given any string $s \in \{0,1\}^n$, there exist $k' \leq k$ integers $r_0, r_1, \ldots, r_{k'}$ with $0 = r_0 < \cdots < r_{k'} = n$, such that along the path of Π given by s, only one player (either Alice or Bob) speaks for rounds $r_i + 1, \ldots, r_{i+1}$ for any $0 \leq i < k'$.*

It is easy to see that the two notions are essentially equivalent, as an n-round protocol with average message length ℓ is an (n/ℓ)-alternating protocol, and a k-alternating n-round protocol has average message length n/k. Note that an n-round *alternating* protocol has average message length 1, while a *one-way* protocol has average message length n. The average message length can thus be seen as a natural measure for the interactivity of a protocol.

We emphasize that the average message length definition does not require message lengths to be uniform along any path or across paths. In particular, this allows for the length of a response to vary depending on what was communicated before, e.g., the statement the other party has just made—a common phenomenon in many applications. Taking as an example real-world conversations between two people, responses to statements can be as short as a simple “I agree” or much longer, depending on what the conversation has already covered and what the opinion or input of the receiving party is. Thus, a sufficiently large average message length roughly states that while the i^{th} response of a person can be short or long depending on the history of the conversation, no sequence of responses can lead to two parties going back and forth with super short statements for too long a period of time. This flexibility makes the average message length a highly applicable parameter that is reasonably large in most settings of interest. We expect it to be a very useful parametrization for questions going beyond the communication rate considered here.

However, the non-uniformity of protocols with an average message length bound can make the design and analysis of protocols somewhat harder than one would like. Fortunately, adding some dummy rounds of communication in a simple procedure we call *blocking* allows us to transform any protocol with small number of alternations into a much more regularly structured protocol which we refer to as *blocked*.

Definition 20. *An n-round protocol Π is said to be b-*blocked *if for any $1 \leq j \leq \lceil n/b \rceil$, only one person (either Alice or Bob) speaks during all rounds r such that $(j-1)b < r \leq jb$.*

Lemma 16. *Any n-round k-alternating protocol Π can be simulated by a b-blocked protocol Π' that consists of at most $n + kb$ rounds.*

Proof of Lemma 16. Consider the protocol tree of Π, where each node corresponds to a state of the protocol (with the root as the starting state) and each node has at most two edges leaving from it (labeled '0' and '1'). Moreover, each node is colored one of two colors depending on whether Alice or Bob speaks next in the corresponding state, and the edges emanating from the node are colored the same. The leaves of the protocol tree are terminating states of the protocol, and one can view any (possibly corrupted) execution of the protocol as a path from the root to a leaf of the tree, where the edge taken from any node indicates the bit that is transmitted by the sender from the corresponding state.

Now, consider any path down the protocol tree. We can group the edges of the path into maximal groups of consecutive edges of the same color. Now, if any group of edges contains a number of edges that is not a multiple of b, then we add some dummy nodes (with edges) in the middle of the group so that the new number of edges in the group is the next largest multiple of b. It is clear that if we do this for every path down the original protocol tree, then the resulting protocol tree will correspond to a protocol Π' that is b-blocked and simulates Π (i.e., each leaf of Π' corresponds to a leaf of Π).

Moreover, note that the number of groups of edges is at most k, since Π is k-alternating. Also, the number of dummy nodes we add in each group is at most b. It follows that the number of nodes (and edges) down any original path of Π has increased by at most kn in Π'. Thus, the desired claim follows. □

4.5 Warmup: Interactive Coding for Random Errors

As a warmup for the much more difficult adversarial setting, we first consider the setting of random errors, as this will illustrate several ideas including blocking, the use of error-correcting codes, and how to incorporate those with known techniques in coding for interactive communication.

In this section, we suppose that each transmission of Alice and Bob occurs over a binary symmetric channel with an ϵ probability of corruption. Recall that we wish to encode an n-round protocol Π into a protocol $\Pi_{\mathrm{enc}}^{\mathrm{random}}$ such that with high probability over the communication channel, execution of $\Pi_{\mathrm{enc}}^{\mathrm{random}}$ robustly simulates Π. By [Hae14], it is known that one can achieve a communication rate of $1 - O(\sqrt{\epsilon})$. In this section, we show how to go beyond the rate of $1 - O(\sqrt{\epsilon})$ for protocols with at least a constant (in ϵ) average message length.

4.5.1 Trivial Scheme for Non-Adaptive Protocols with Minimum Message Length

The first coding scheme we present for completeness is a completely trivial and straight forward application of error correcting codes which works for *non-adaptive* protocols Π with a guaranteed *minimum* message length. In particular, the coding scheme achieves a communication rate of $1 - O(H(\epsilon))$ for non-adaptive protocols with minimum message length $\Omega((1/\epsilon)\log n)$.

In particular, we assume that Π is a a non-adaptive n-round protocol with message lengths of size $b_1, b_2, \ldots, b_k$, i.e., Alice sends b_1 bits, then Bob sends b_2 bits, and so on. Moreover, we assume that that $b_1, b_2, \ldots, b_k \geq b$, where $b = \Omega((1/\epsilon)\log n)$ is the minimum message length.

Now, we can form the encoded protocol $\Pi_{\text{enc}}^{\text{random}}$ by simply having the transmitting party replace its intended message in Π (of b_i bits) with the encoding (of length, say, b_i') of the message under an error-correcting code of minimum relative distance $\Omega(\epsilon)$ and rate $1 - O(H(\epsilon))$ and then transmitting the resulting codeword. The receiver then decodes the word according to the nearest codeword of the appropriate error-correcting code.

Note that for any given message (codeword) of length b_i', the expected number of corruptions due to the channel is $\epsilon b_i'$. Thus, by Chernoff bound, the probability that the corresponding codeword is corrupted beyond half the minimum distance of the relevant error-correcting code is $e^{-\Omega(\epsilon b')} = n^{-\Omega(1)}$. Since $k = O(n/b) = O(n\epsilon/\log n)$, the union bound implies that the probability that any of the $k < n$ messages is corrupted beyond half the minimum distance is also $n^{-\Omega(1)}$. Thus, with probability $1 - n^{-\Omega(1)}$, $\Pi_{\text{enc}}^{\text{random}}$ simulates the original protocol without error. Moreover, the overall communication rate is clearly $1 - O(H(\epsilon))$ due to the choice of the error-correcting codes.

Remark 24. *Note that the aforementioned trivial coding scheme has the disadvantage of working only for nonadaptive protocols with a certain* minimum *message length, which is a much stronger assumption than average message length. In Section 4.5.2, we show how to get around this problem by converting the input protocol to a* blocked *protocol.*

Another problem with the coding scheme is that the minimum message length is required to be $\Omega_\epsilon(\log n)$. This is in order to ensure that the probability of error survives a union bound, as the trivial coding scheme has no mechanism for recovering if a particular message gets corrupted. This also results in a success probability of only $1 - 1/\text{poly}(n)$ instead of the $1 - \exp(n)$ one would like to have for a coding scheme. Section 4.5.2 shows how to rectify both problems by combining the reduced error probability of a error correcting code failing with any existing interactive coding scheme, such as the one in

[Hae14].

4.5.2 Coding Scheme for Protocols with Average Message Length of $\Omega(\log(1/\epsilon)/\epsilon^2)$

In this section, we build on the trivial scheme discussed earlier to provide an improved coding scheme that handles any protocol Π with an *average message length* of at least $\ell = \Omega(\log(1/\epsilon)/\epsilon^2)$.

The first step will be to transform Π into a protocol that is blocked. Note that the Π is a k-alternating protocol, where $k = n/\ell = O(n\epsilon^2/\log(1/\epsilon))$. Thus, by Lemma 16, we can transform Π into a b-blocked protocol Π_{blk}, for $b = \Theta(\log(1/\epsilon)/\epsilon)$, such that Π_{blk} simulates Π and consists of $n_b = n + kb = n(1 + O(\epsilon))$ rounds.

Now, we view Π_{blk} as a q-ary protocol with n_b/b rounds, where $q = 2^b$. This can be done by grouping the symbols in each b-sized block as a single symbol from an alphabet of size q. Next, we can use the coding scheme of [Hae14] in a blackbox manner to encode this q-ary protocol as a q-ary protocol Π' with $\frac{n_b}{b}(1+\Theta(\sqrt{\epsilon'}))$ rounds such that Π' simulates Π under oblivious random errors with error fraction ϵ' (i.e., each q-ary symbol is corrupted (in any way) with an independent probability of at most ϵ'). We pick $\epsilon' = \epsilon^4$.

Finally, we transform Π' into a *binary* protocol $\Pi_{\text{enc}}^{\text{random}}$ as follows: We expand each q-ary symbol of Π' back into a sequence of b bits and then expand the b bits into $b' > b$ bits using an error-correcting code. In particular, we use an error-correcting code $\mathcal{C} : \{0,1\}^b \to \{0,1\}^{b'}$ with block length $b' = b + (2c + \delta)\log^2(1/\epsilon)$ and minimum distance $2c\log(1/\epsilon)$ for appropriate constants c, δ (such a code is guaranteed to exist by the Gilbert-Varshamov bound). Thus, $\Pi_{\text{enc}}^{\text{random}}$ is a b'-blocked binary protocol with $n_b \cdot \frac{b'}{b}(1 + \Theta(\sqrt{\epsilon'})) = n(1 + O(\epsilon\log(1/\epsilon))$ rounds. Moreover, each b'-sized block of $\Pi_{\text{enc}}^{\text{random}}$ simply simulates each q-ary symbol of Π' and the listening party simply decodes the received b' bits to the nearest codeword of $\mathcal{C}$.

To see that $\Pi_{\text{enc}}^{\text{random}}$ successfully simulates Π in the presence of random errors with error fraction ϵ, observe that a b'-block is decoded incorrectly if and only if more than $d/2$ of the b' bits are corrupted. By the Chernoff bound, the probability of such an event is $< \epsilon^4$ (for appropriate choice of c, δ). Thus, since Π' is known to simulate Π under oblivious errors with error fraction ϵ^4, it follows that $\Pi_{\text{enc}}^{\text{random}}$ satisfies the desired property.

4.6 Conceptual Challenges and Key Ideas

In this section, we wish to provide some intuition for the difficulties in surpassing the $1 - \Theta(\sqrt{\epsilon})$ communication rate for interactive coding when dealing with non-random errors. We do this because the adversarial setting comes with a completely new set of challenges that are somewhat subtle but nonetheless fundamental. As such, the techniques used in the previous section for interactive coding under random errors still provide a good introduction to some of the building blocks in the framework we use to deal with the adversarial setting, but they are not sufficient to circumvent the main technical challenges. Indeed, we show in this section that the adversarial setting inherently requires several completely new techniques to beat the $1 - \Theta(\sqrt{\epsilon})$ communication rate barrier.

We begin by noting that all existing interactive coding schemes encode the input protocol Π into a protocol Π' with a certain type of structure: There are some, a priori specified, communication rounds which simulate rounds of the original protocol (i.e., result in a walk down the protocol tree of Π), while other rounds constitute *redundant information* which is used for error correction. In the case of protocols that use hashing (e.g., [Hae14], [KR13]), this is directly apparent in their description, as rounds in which hashes and control information are communicated constitute redundant information. However, this is also the case for all protocols based on tree codes (e.g., [BR14, GHS14, GH14]): To see this, note that in such protocols, one can simply use an underlying tree code that is linear and systematic, with the non-systematic portion of the tree code then corresponding to redundant rounds.

We next present an argument which shows that, due to the above structure, no existing coding scheme can break the natural $1 - \Omega(\sqrt{\epsilon})$ communication rate barrier, even for protocols with near-linear $o(n)$ average message lengths. This will also provide some intuition about what is required to surpass this barrier.

Suppose that for a (randomized) n-round communication protocol Π, the simulating protocol Π' has the above structure and a communication rate of $1 - \epsilon'$. The simulation Π' thus consists of exactly $N = n/(1 - \epsilon')$ rounds. Note that, since every simulation must have at least n non-redundant rounds, the fraction of redundant rounds in Π' can be at most ϵ'. Given that the position of the redundant rounds is fixed, it is therefore possible to find a window of $(\epsilon/\epsilon')N$ consecutive rounds in Π' which contain at most ϵN redundant rounds, i.e., an ϵ' fraction. Now, consider an oblivious adversarial channel that corrupts all the redundant information in the window along with a few extra rounds. Such an adversary renders any error correction technique useless, while the few extra errors derail the unprotected parts of the communication, thereby rendering essentially all the non-redundant information communicated in this window useless as well—all while corrupting essentially only ϵN rounds in total. This implies that in the remaining $N - (\epsilon/\epsilon')N$

communication rounds outside of this window, there must be at least n non-redundant rounds in order for Π' to be able to successfully simulate Π. However, it follows that $N-(\epsilon/\epsilon')N \geq n = N(1-\epsilon')$ which simplifies to $1-(\epsilon/\epsilon') \geq 1-\epsilon'$, or $\epsilon'^2 \geq \epsilon$, implying that the communication rate of $1-\epsilon'$ can be at most $1-\Omega(\sqrt{\epsilon})$, where ϵ is the fraction of errors applied by the channel.

this window in order to simulate the input protocol Π even in the absence of any further errors. This implies that $n \geq (1-\epsilon')n + (\epsilon/\epsilon')n$, which implies that $\epsilon' \geq \sqrt{\epsilon}$, meaning that the communication rate must be $1-\Omega(\sqrt{\epsilon})$.

One can note that a main reason for the $1-\Omega(\sqrt{\epsilon})$ limitation in the above argument is that the adversary can target the rounds with redundant information in the relevant window. For instance, in the interactive coding scheme of [Hae14], the rounds with control information are in predetermined positions of the encoded protocol, and so, the adversary knows exactly which locations to corrupt.

Our idea for overcoming the aforementioned limitations in the case of an *oblivious* adversarial channel is to employ some type of **information hiding** to hide the locations of the redundant rounds carrying control/verification information. In particular, we randomize the locations of control information bits within the output protocol, which allows us to guard against attacks that target solely the redundant information. In order to allow for this synchronized randomization in the standard *private randomness* model assumed in this chapter, Alice and Bob use the standard trick of first running an error-corrected randomness exchange procedure that allows them to establish some shared randomness hidden from the oblivious adversary that can be used for the rest of the simulation. Note that this inherently does not work for a *fully adaptive* adversary, as the adversary can adaptively choose which locations to corrupt based on any randomness that has been shared over the channel. In fact, we believe that beating the $1-\Omega(\sqrt{\epsilon})$ communication rate barrier against fully adaptive adversaries may be fundamentally impossible for precisely this reason.

Information hiding, while absolutely crucial, does not, however, make use of a larger average message length which, according to the conjectures of [Hae14], is necessary to beat the $1-\Omega(\sqrt{\epsilon})$ barrier. The idea we use for this, as already demonstrated in Section 4.5, is the use of blocking and the subsequent application of error-correcting codes on each such block.

Unfortunately, the same argument as given above shows that a straightforward application of *block* error-correcting codes, as done in Section 4.5, cannot work against an oblivious adversarial channel. The reason is that in such a case, an application of *systematic* block error-correcting codes would be possible as well, and such codes again have pre-specified positions of redundancy which can be targeted by the adversarial channel.

In particular, one could again disable all redundant rounds including the non-systematic parts of block error-correcting codes in a large window of $(\epsilon/\epsilon')N$ rounds and make the remaining communication useless with few extra errors. More concretely, suppose that one simply encodes all blocks of data with a standard block error-correcting code. For such block codes, one needs to specify a priori how much redundancy should be added, and the natural direction would be to set the relative distance to, say, 100ϵ given that one wants to prepare against an error rate of ϵ. However, this would allow the adversary to corrupt a constant fraction (e.g., $1/200$) of error correcting codes beyond their distance, thus making a constant fraction of the communicated information essentially useless. This would lead to a communication rate of $1 - \Theta(1)$. It can again be easily seen that in this tradeoff, the best fixed relative distance one can choose for block error-correcting codes is essentially $\sqrt{\epsilon}$, which would lead to a rate loss of $H(\sqrt{\epsilon})$ for the error-correcting codes but would also allow the adversary to corrupt at most a $\sqrt{\epsilon}$ fraction of all codewords. This would again lead to an overall communication rate of $1 - \tilde{\Omega}(\sqrt{\epsilon})$.

Our solution to the hurdle of having to commit to a fixed amount of redundancy in advance is to use **rateless error-correcting codes**. Unlike block error-correcting codes with fixed block length and minimum distance, rateless codes encode a message into a potentially *infinite* stream of symbols such that having access to enough uncorrupted symbols allows a party to decode the desired message with a resulting communication rate that *adapts* to the true error rate without requiring a priori knowledge of the error rate. Since it is not possible for Alice and Bob to know in advance which data bits the adversary will corrupt, rateless codes allow them to adaptively adjust the amount of redundancy for each communicated block, thereby allowing the correction of errors without incurring too great a loss in the overall communication rate.

4.7 Main Result: Interactive Coding for Oblivious Adversarial Errors

In this section, we develop our main result. We remind the reader that in the oblivious adversarial setting assumed throughout the rest of Chapter 4, the adversary is allowed to corrupt up to an ϵ fraction of the total number of bits exchanged by Alice and Bob. The adversary commits to the locations of these bits before the start of the protocol. Alice and Bob will use randomness in their encoding, and one asks for a coding scheme that allows Alice and Bob to recover the transcript of the original protocol with exponentially high probability in the length of the protocol (over the randomness that Alice and Bob use) for any fixed error pattern chosen by the adversary.

For simplicity in exposition, we assume that the input protocol is *binary*, so that the simulating output protocol will also be binary. However, the results hold virtually as-is for protocols over larger alphabet. We first provide a high-level overview of our construction of an encoded protocol. The pseudocode of the algorithm appears in Figure 4.3.

4.7.1 High-Level Description of Coding Scheme

Let us describe the basic structure of our interactive coding scheme. Suppose Π is an n-round binary input protocol with average message length $\ell \geq \mathrm{poly}(1/\epsilon)$. Using Lemma 16, we first produce a B-blocked binary protocol Π_{blk} with n' rounds that simulates Π.

Our encoded protocol $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ will begin by having Alice and Bob performing a *randomness exchange procedure*. More specifically, Alice will generate some number of bits from her private randomness and encode the random string using an error-correcting code of an appropriate rate and distance. Alice will then transmit the encoding to Bob, who can decode the received string. This allows Alice and Bob to maintain *shared random bits*. The randomness exchange procedure is described in further detail in Section 4.7.3.

Next, $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ will simulate the B-sized blocks (which we call *B-blocks*) of Π_{blk} in order in a structured manner. Each B-block will be encoded as a string of $2B$ bits using a *rateless code*, and the encoded string will be divided into *chunks* of size $b < B$. For a detailed discussion on the encoding procedure via rateless codes, see Section 4.7.4.

Now, $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ will consist of a series of N_{iter} *iterations*. Each iteration consists of transmitting b' rounds, and we call such a b'-sized unit a *mini-block*, where $b' > b$. Each mini-block will consist of b *data bits*, as well as $b' - b$ bits of *control information*. The data bits in successive mini-blocks will taken from the successive b-sized chunks obtained by the encoding under the rateless code. Meanwhile, the control information bits are sent by Alice and Bob in order to check whether they are in sync with each other and to allow a *backtracking* mechanism to tack place if they are not.

For a particular B-block that is being simulated, mini-blocks keep getting sent until the receiving party of the B-block is able to decode the correct B-block, after which Alice and Bob move on to the next B-block in Π.

In addition to data bits, each mini-block also contains $b' - b$ bits of control information. A party's unencoded control information during a mini-block consists of some hashes of his view of the current state of the protocol as well as some backtracking parameters. The aforementioned quantities are encoded using a hash for verification as well as an error-correcting code. Each party sends his encoded control information as part of each mini-block. The locations of the control information within each mini-block will be randomized for the sake of *information hiding*, using bits from the shared randomness of Alice and Bob. This is described in further detail in Section 4.8. Moreover, we note that the hashes used for the control information in each mini-block are seeded using bits from the shared randomness. The structure of each mini-block is shown in Figure 4.2.

After each iteration, Alice and Bob try to decode each other's control information in order to determine whether they are in sync. If not, the parties decide whether to backtrack in a controlled manner (see Section 4.9 for details).

Throughout the protocol, Alice maintains a *block index* c_A (which indicates which block of Π_{blk} she believes is currently being simulated), a *chunk counter* j_A, a *transcript* (of the blocks in Π_{blk} that have been simulated so far) T_A, a *global counter* m (indicating the number of the current iteration), a *backtracking parameter* k_A, as well as a *sync parameter* sync_A. Similarly, Bob maintains c_B, j_B, T_B, m, k_B, and sync_B.

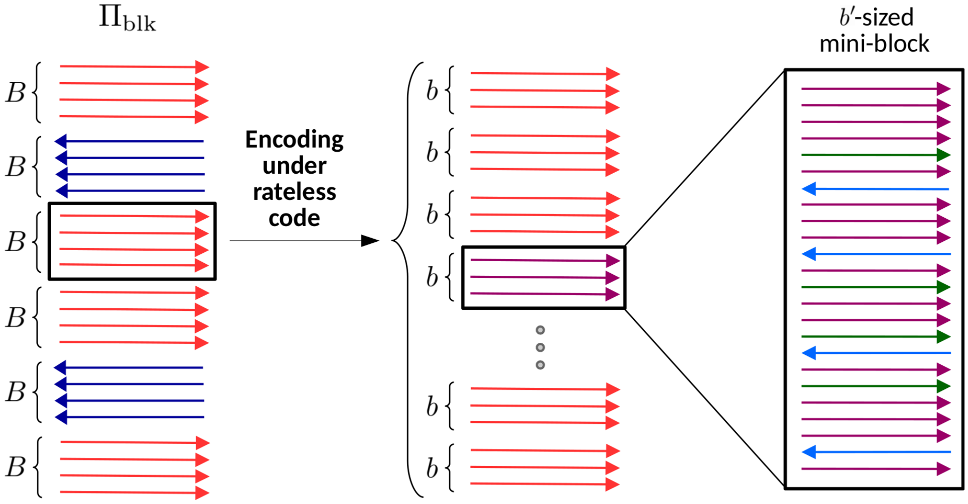

Figure 4.2: Each B-block of Π_{blk} gets encoded into chunks of size b using a rateless code. Every b'-sized mini-block in $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ consists of the b bits of such a chunk, along with $(b' - b)/2$ bits of Alice's control information and $(b' - b)/2$ bits of Bob's control information. The positions of the control information within a mini-block are randomized. Note that rounds with Alice's control information are in green, while rounds with Bob's control information are in light blue.

4.7.2 Parameters

We now set the parameters of the protocol. For convenience, we will define a *loss parameter* $\epsilon' < \epsilon$. Our interactive coding scheme will incur a rate loss of $\Theta(\epsilon' \operatorname{polylog}(1/\epsilon'))$, in addition to the usual rate loss of $\Theta(H(\epsilon))$. Alice and Bob are free to decide on an ϵ' based on what rate loss they are willing to tolerate in the interactive coding scheme. In particular, note that if $\epsilon' = \Theta(\epsilon^2)$, then the rate loss of $\Theta(\epsilon' \operatorname{polylog}(1/\epsilon'))$ is overwhelmed by $\Theta(H(\epsilon))$. For the purposes of Theorem 22, it will suffice to take $\epsilon' = \Theta(\epsilon^2)$ at then end, but for the sake of generality, we maintain ϵ' as a separate parameter.

We now take the average message length threshold to be $\Omega(1/\epsilon'^3)$, i.e., we assume that our input protocol Π has average message length $\ell = \Omega(1/\epsilon'^3)$. Then, Π has at most $\mathsf{alt} = n/\ell = O(n\epsilon'^3)$ alternations. Moreover, we take $B = \Theta(1/\epsilon'^2)$ and $b = s = \Theta(1/\epsilon')$, with $B = sb$. Then, by Lemma 16, note that $n' \leq n + \mathsf{alt} \cdot B = n(1 + O(\epsilon'))$.

We also take $b' = b + 2c\log(1/\epsilon')$, so that within each b'-sized mini-block, each party transmits $c\log(1/\epsilon')$ bits of (encoded) control information.

Finally, we take $N_{\mathsf{iter}} = \frac{n'}{b}(1 + \Theta(\epsilon \log(1/\epsilon))$ iterations. This will guarantee, with high probability, that at the end of the protocol, Alice and Bob have successfully simulated all blocks of Π_{blk}, and therefore, Π. Also, it should be noted that we append trivial blocks of zeros (sent by, say, Alice) to the end of Π_{blk} to simulate in case $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ ever runs out of blocks of Π_{blk} to simulate (because it has reached the bottom of the protocol tree) before N_{iter} iterations of $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$ have been executed.

4.7.3 Randomness Exchange

Alice and Bob will need to have some number of shared random bits throughout the course of the protocol. The random bits will be used for two main purposes: *information hiding* and *seeding hash functions*, which will be discussed in Section 4.8. As it turns out, it will suffice for Alice and Bob to have $l' = O(n\epsilon' \operatorname{polylog}(1/\epsilon'))$ shared random bits for the entirety of the protocol, using some additional tricks.

Thus, in the private randomness model, it suffices for Alice to generate the necessary number of random bits and transmit them to Bob using an error-correcting code. More precisely, Alice generates a uniformly random string $\mathsf{str} \in \{0,1\}^{l'}$, uses an error-correcting code $\mathcal{C}^{\mathsf{exchange}} : \{0,1\}^{l'} \to \{0,1\}^{10\epsilon N_{\mathsf{iter}} b'}$ of relative distance $2/5$ to encode str, and transmits the encoded string to Bob. Since the adversary can corrupt only at most ϵ fraction of all bits, the transmitted string cannot be corrupted beyond half the minimum distance of $\mathcal{C}^{\mathsf{exchange}}$. Hence, Bob can decode the received string and determine str.

Note that the exchange of randomness via the codeword in $\mathcal{C}^{\mathsf{exchange}}$ results in a rate loss of $\Theta(\epsilon)$, which is still overwhelmed by $\Theta(H(\epsilon))$.

4.7.4 Sending Data Bits Using "Rateless" Error-Correcting Codes

To transmit data from blocks of Π_{blk}, we will use an error-correcting code that has incremental distance properties. One can think of this as a rateless code with minimum distance properties. Recall that $b = s = \Theta(1/\epsilon')$ and $B = sb$. In particular, we require an error-correcting code $\mathcal{C}^{\mathsf{rateless}} : \{0,1\}^B \to \{0,1\}^{2B}$ for which the output is divided in to $2s$ chunks of b bits each such that the code restricted to any contiguous block (with cyclic wrap-around) of $> s$ chunks has a certain guaranteed minimum distance. The following lemma guarantees the existence of such a code.

Lemma 17. *For sufficiently large b, s, there exists an error-correcting code $\mathcal{C} : \{0,1\}^{sb} \to \{0,1\}^{2sb}$ such that for any $a = 0, 1, \ldots, 2s-1$ and $j = s+1, s+2, \ldots, 2s$, the code $\mathcal{C}_{a,j} : \{0,1\}^{sb} \to \{0,1\}^{jb}$ formed by restricting $\mathcal{C}$ to the bits $ab, ab+1, \ldots, ab+jb-1$ (modulo $2sb$) has relative distance at least $\delta_j = H^{-1}\left(\frac{j-s}{j} - \frac{1}{4s}\right)$, while $\mathcal{C}$ has relative distance at least $\delta_{2s} = \frac{1}{15}$. (Here, H^{-1} denotes the unique inverse of H that takes values in $[0, 1/2]$.)*

Proof of Lemma 17. We use a slight modification of the random coding argument that is often used to establish the Gilbert-Varshamov bound. Suppose we pick a random linear code. For $s < j \le 2s$, let us consider the probability $P_{a,j}$ that the resulting $\mathcal{C}_{a,j}$ does not have relative distance at least δ_j. Consider any codeword $y \in \{0,1\}^{jb}$ in $\mathcal{C}_{a,j}$. The probability that y has Hamming weight less than δ_j is at most $2^{-jb(1-H(\delta_j))}$. Thus, by the union bound, we have that the probability that $\mathcal{C}_{a,j}$ contains a codeword of Hamming weight less than δ_j is at most

$$\begin{aligned} P_{a,j} = 2^{sb} \cdot 2^{-jb(1-H(\delta_j))} &= 2^{sb-jb\left(1-\frac{j-s}{j}+\frac{1}{4s}\right)} \\ &= 2^{-jb/4s} \\ &\le 2^{-b/4}. \end{aligned}$$

Similarly, P, the probability that $\mathcal{C}$ contains a codeword of Hamming weight less than $\frac{2}{15}s$, is at most

$$P \le 2^{sb} \cdot 2^{-2sb(1-H(2/15))} \le 2^{-sb/4} \le 2^{-b/4}.$$

Therefore, by another application of the union bound, the probability that some $\mathcal{C}_{a,j}$ or $\mathcal{C}$ does not have the required relative distance is at most

$$P + \sum_{\substack{0 \le a \le 2s-1 \\ s < j \le 2s}} P_{a,j} \le (2s^2 + 1) \cdot 2^{-b/4} < 1$$

for sufficiently large b, s. □

Remark 25. *For our purposes, $b = s = \Theta(1/\epsilon')$. Therefore, for suitably small $\epsilon' > 0$, there exists such an error-correcting code $\mathcal{C}$ as guaranteed by Lemma 17. Moreover, it is possible to find a such a code by brute force in time* $\operatorname{poly}(1/\epsilon')$.

Thus, Alice and Bob can agree on a fixed error-correcting code $\mathcal{C}^{\mathsf{rateless}}$ of the type guaranteed by Lemma 17 prior to the start of the algorithm. Now, let us describe how data bits are sent during the iterations of $\Pi_{\text{enc}}^{\text{oblivious}}$. The blocks of $\Pi_{\text{enc}}^{\text{oblivious}}$ are simulated in order as follows.

First, suppose Alice's block index c_A indicates a B-block in Π_{blk} during which Alice is the sender. Then in $\Pi_{\text{enc}}^{\text{oblivious}}$, Alice will transmit up to a maximum of $2s$ chunks (of size b) that will encode the data x from that block. More specifically, Alice will compute $y = \mathcal{C}^{\mathsf{rateless}}(x) \in \{0,1\}^{2B}$ and decompose it as $y = y_0 \circ y_1 \circ \cdots \circ y_{2s-1}$, where $\circ$ denotes concatenation and $y_0, y_1, \ldots, y_{2s-1} \in \{0,1\}^b$.

Recall that each mini-block of $\Pi_{\text{enc}}^{\text{oblivious}}$ contains b data bits (in addition to $b' - b$ control bits). Thus, Alice can send each y_i as the data bits of a mini-block. The chunk that Alice sends in a given iteration depends on the global counter m. In particular, Alice always sends the chunk $y_{m \bmod 2s}$. Moreover, Alice keeps a chunk counter j_A, which is set to 0 during the first iteration in which she transmits a chunk from y and then increases by 1 during each subsequent iteration (until $j_A = 2s$, at which point j_A stops increasing).

On the other hand, suppose Alice's block index c_A indicates a B-block in Π_{blk} during which Alice is the receiver. Then, Alice listens for data during each mini-block. Alice stores her received b-sized chunks as $\widetilde{g}_0, \widetilde{g}_1, \ldots$ and increments her chunk counter j_A after each iteration to keep track of how many chunks she has stored, along with a, an index indicating which y_a she expects the first chunk $\widetilde{g}_0$ to be. Once Alice has received more than s chunks (i.e., $j_A > s$), she starts to keep an estimate $\widetilde{x}$ of the data x that Bob is sending that Alice has by decoding $\widetilde{g}_0 \circ \widetilde{g}_1 \circ \cdots \circ \widetilde{g}_{j_A - 1}$ to the nearest codeword of $\mathcal{C}_{a,j_A}^{\mathsf{rateless}}$. This estimate is updated after each subsequent iteration. As soon as Alice undergoes an iteration in which she receives valid control information suggesting that $\widetilde{x} = x$ (if Alice's estimate $\widetilde{x}$ matches the hash of x that Bob sends as control information, see Section 4.8), she advances her block index c_A and appends her transcript T_A with $\widetilde{x}$.

Note that it is possible that j_A reaches $2s$ and Alice has not yet received valid control information suggesting that he has decoded x. In this case, Alice resets j_A to 0 and also resets a to the current value of m, thereby restarting the listening process. Also, during any iteration, if Alice receives control information suggesting that $j_B < j_A$ (i.e., Alice has been listening for a greater number of iterations than Bob has been transmitting), then again, Alice resets j_A and a and restarts the process.

Remark 26. *The key observation is that using a rateless code allows the amount of redundancy in data that the sender sends to* adapt *to the number of errors being introduced by the adversary, rather than wasting redundant bits or not sending enough of them.*

4.8 Control Information

Alice's unencoded control information in the m^{th} iteration consists of (1.) a hash $h_{A,c}^{(m)} = hash(c_A, S)$ of the block index c_A, (2.) a hash $h_{A,x}^{(m)} = hash(x, S)$ of the data in the current block of Π_{blk} being communicated, (3.) a hash $h_{A,k}^{(m)} = hash(k_A, S)$ of the *backtracking parameter* k_A, (4.) a hash $h_{A,T}^{(m)} = hash(T_A, S)$ of Alice's transcript T_A, (5.) a hash $h_{A,\texttt{MP1}}^{(m)} = hash(T_A[1, \texttt{MP1}], S)$ of Alice's transcript up till the first *meeting point*, (6.) a hash $h_{A,\texttt{MP2}}^{(m)} = hash(T_A[1, \texttt{MP2}], S)$ of Alice's transcript up till the second *meeting point*, (7.) the chunk counter j_A, and (8.) the *sync parameter* sync_A. Here, S refers to a string of fresh random bits used to seed the hash functions (note that S is different for each instance). Thus, we write Alice's unencoded control information as

$$\mathsf{ctrl}_A^{(m)} = \left(h_{A,c}^{(m)}, h_{A,x}^{(m)}, h_{A,k}^{(m)}, h_{A,T}^{(m)}, h_{A,\texttt{MP1}}^{(m)}, h_{A,\texttt{MP2}}^{(m)}, j_A, \mathsf{sync}_A\right).$$

Bob's unencoded control information $\mathsf{ctrl}_B^{(m)}$ is similar in the analogous way.

For the individual hashes, we can use the following Inner Product hash function $hash : \{0,1\}^l \times \{0,1\}^r \to \{0,1\}^p$, where $r = lp$:

$$hash(X, R) = \left(\langle X, R_{[1,l]}\rangle, \langle X, R_{[l+1,2l]}\rangle, \ldots, \langle X, R_{[lp-(l-1),lp]}\rangle\right),$$

where the first argument X is the quantity to be hashed, and the second argument R is a random seed. This choice of hash function guarantees the following property:

Property 4.8.1. *For any $X, Y \in \{0,1\}^l$ such that $X \neq Y$, we have that*

$$\Pr_{R \sim \text{Unif}(\{0,1\}^r)}[hash(X, R) = hash(Y, R)] \leq 2^{-p}.$$

Now, we wish to take output size $p = O(\log(1/\epsilon'))$ for each of the hashes so that the total size of each party's control information in any iteration is $O(\log(1/\epsilon'))$. Note that some of the quantities we hash (e.g., T_A, T_B) actually have size $l = \Omega(n)$. Thus, for the corresponding hash function, we would naively require $r = lp = \Omega(n \log(1/\epsilon'))$ fresh bits of randomness for the seed (per iteration), for a total of $\Omega(N_{\mathsf{iter}} n \log(1/\epsilon'))$ bits of randomness. However, as described in Section 4.7.3, Alice and Bob only have access to $O(n\epsilon' \text{polylog}(1/\epsilon'))$ bits of shared randomness!

To get around this problem, we make use of δ-biased sources to minimize the amount of randomness we need. In particular, we can use the δ-biased sample space of [NN93] to stretch $\Theta(\log(L/\delta))$ independent random bits into a string of $L = \Theta(N_{\mathsf{iter}} n \log(1/\epsilon'))$ pseudorandom bits that are δ-biased. We take $\delta = 2^{-\Theta(N_{\mathsf{iter}} \cdot p)}$. The sample space guarantees that the L pseudorandom bits are $\delta^{\Theta(1)}$-statistically close to being k-wise independent for $k = \log(1/\delta) = \Theta(N_{\mathsf{iter}} \cdot p) = \Theta(N_{\mathsf{iter}} \log(1/\epsilon'))$. Moreover, the Inner Product Hash Function satisfies the following modified collision property, which follows trivially from Property 4.8.1 and the definition of δ-bias:

Property 4.8.2. *For any $X, Y \in \{0,1\}^l$ such that $X \neq Y$, we have that*

$$\Pr_R[hash(X,R) = hash(Y,R)] \leq 2^{-p} + \delta,$$

where R is sampled from a δ-biased source.

As it turns out, this property is good enough for our purposes. Thus, after the randomness exchange, Alice and Bob can simply take $\Theta(\log(L/\delta))$ bits from str and stretch them into an L-bit string $\mathsf{str}_{\text{stretch}}$ as described. Then, for each iteration, Alice and Bob can simply seed their hash functions using bits from $\mathsf{str}_{\text{stretch}}$.

4.8.1 Encoding and Decoding Control Information

Recall that during the m^{th} iteration, Alice's (unencoded) control information is $\mathsf{ctrl}_A^{(m)}$, while Bob's (unencoded) control information is $\mathsf{ctrl}_B^{(m)}$. In this section, we describe the encoding and decoding functions that Alice and Bob use for their control information. We start by listing the properties we desire.

Definition 21. *Suppose $X \in \{0,1\}^l$ and $V \in \{*, \neg, 0, 1\}^l$ for some $l > 0$. Then, we define $\mathsf{Corrupt}_V(X) = Y \in \{0,1\}^l$ as follows:*

$$Y_i = \begin{cases} V_i & \text{if } V_i \in \{0,1\} \\ X_i \oplus 1 & \text{if } V_i = \neg \\ X_i & \text{if } V_i = * \end{cases}.$$

Moreover, we define $\mathsf{wt}(V)$ *to be the number of coordinates of* V *that are not equal to* $*$.

Remark 27. *Note that* V *corresponds to an error pattern. In particular,* $*$ *indicates a position that is not corrupted, while* $\neg$ *indicates a bit flip, and 0/1 indicate a bit that is fixed to the appropriate symbol (see Section 4.3.1 for details about* flip *and* replace *errors). The function* $\mathsf{Corrupt}_V$ *applies the error pattern* V *to the bit string given as an argument. Also, wt*(V) *corresponds to the number of positions that are targeted for corruption.*

We require a seeded encoding function $\mathsf{Enc} : \{0,1\}^l \times \{0,1\}^r \to \{0,1\}^o$ as well as a seeded decoding function $\mathsf{Dec} : \{0,1\}^o \times \{0,1\}^r \to \{0,1\}^l \cup \{\perp\}$ such that the following property holds:

Property 4.8.3. *The following holds:*

1. *For any* $X \in \{0,1\}^l$, $R \in \{0,1\}^r$, *and* $V \in \{*, \neg, 0, 1\}^o$ *such that* $\mathsf{wt}(V) < \frac{1}{8}o$,
$$\mathsf{Dec}(\mathsf{Corrupt}_V(\mathsf{Enc}(X,R)),R) = X.$$

2. *For any* $X \in \{0,1\}^l$ *and* $V \in \{0,1\}^o$ *such that* $\mathsf{wt}(V) \geq \frac{1}{8}o$,
$$\Pr_{R \sim \mathrm{Unif}(\{0,1\}^r)}[\mathsf{Dec}(\mathsf{Corrupt}_V(\mathsf{Enc}(X,R)),R) \notin \{X, \perp\}] \leq 2^{-\Omega(l)}.$$

Remark 28. *The second argument of* Enc *and* Dec *will be a* seed*, which is generated by taking* r *fresh bits from the shared randomness of Alice and Bob. A decoding output of* $\perp$ *indicates a decoding failure. Moreover, (1.) of Property 4.8.3 guarantees that a party can successfully decode the other party's control information if at most a constant fraction of the encoded control information symbols are corrupted (this is then used to prove Lemmas 18 and 19). On the other hand, (2.) of Property 4.8.3 guarantees that if a larger fraction of the encoded control information symbols are corrupted, then the decoding party can detect any possible corruption with high probability (this is then used to establish Lemma 20).*

We now show how to obtain Enc, Dec that satisfy Property 4.8.3. The idea is that Enc consists of a three-stage encoding: (1.) append a hash value to the unencoded control information, (2.) encode the resulting string using an error-correcting code, and (3.) XOR each output bit with a fresh random bit taken from the shared randomness.

For our purposes, we want $l = O(\log(1/\epsilon'))$ to be the number of bits in $\mathsf{ctrl}_A^{(m)}$ (or $\mathsf{ctrl}_B^{(m)}$) and $o = c\log(1/\epsilon')$.

First, we choose a hash function $h : \{0,1\}^l \times \{0,1\}^t \to \{0,1\}^{o'}$ that has the following property:

Property 4.8.4. *Suppose $X, U \in \{0,1\}^l$, where U is not the all-zeros vector, and $W \in \{0,1\}^{o'}$. Then,*

$$\Pr_{R \sim \mathrm{Unif}(\{0,1\}^t)}[h(X+U,R) = h(X,R) + W] \leq 2^{-o'}.$$

In particular, we can use the simple Inner Product Hash Function with $t = l \cdot o'$ and $o' = \Theta(\log(1/\epsilon'))$:

$$h(X,R) = \left(\left\langle X, R_{[1,l]}\right\rangle, \left\langle X, R_{[l+1,2l]}\right\rangle, \ldots, \left\langle X, R_{[l\cdot o'-(l-1), l\cdot o']}\right\rangle\right).$$

Next, we choose a *linear* error-correcting code $\mathcal{C}^{\mathsf{hash}} : \{0,1\}^{l+o'} \to \{0,1\}^o$ of constant relative distance $1/4$ and constant rate.

We now take $r = t + o$ and define Enc as

$$\mathsf{Enc}(X,R) = \mathcal{C}^{\mathsf{hash}}(X \circ h(X, R_{[o+1,r]})) \oplus R_{[1,o]}.$$

Moreover, we define Dec as follows: Given Y, R, let X' be the decoding of $Y + R_{[1,o]}$ under $\mathcal{C}^{\mathsf{hash}}$ (using the nearest codeword of $\mathcal{C}^{\mathsf{hash}}$ and then inverting the map $\mathcal{C}^{\mathsf{hash}}$). We then define

$$\mathsf{Dec}(Y,R) = \begin{cases} X'_{[1,l]} & \text{if } h(X'_{[1,l]}, R_{[o+1,r]}) = X'_{[l+1,l+o']} \\ \perp & \text{if } h(X'_{[1,l]}, R_{[o+1,r]}) \neq X'_{[l+1,l+o']} \end{cases}.$$

Remark 29. *Note that we have $r = O(\log^2(1/\epsilon'))$, which means that over the course of the protocol $\Pi_{\mathrm{enc}}^{\mathrm{oblivious}}$, we will need $O(N_{\mathsf{iter}} r) = O(n\epsilon' \log^2(1/\epsilon'))$ fresh random bits for the purpose of encoding and decoding control information.*

We now prove that the above Enc, Dec satisfy Property 4.8.3.

Proof. Note that if $V \in \{*, \neg, 0, 1\}^o$ satisfies $\mathsf{wt}(V) < \frac{1}{8}o$, then note that the Hamming distance between $\mathsf{Corrupt}_V(\mathsf{Enc}(X,R))$ and $\mathsf{Enc}(X,R)$ is less than $\frac{1}{8}o$. Hence, since $\mathcal{C}^{\mathsf{hash}}$ has relative distance $1/4$, it follows that under the error-correcting code $\mathcal{C}^{\mathsf{hash}}$, $\mathsf{Corrupt}_V(\mathsf{Enc}(X,R)) \oplus R_{[1,o]}$ and $\mathsf{Enc}(X,R) \oplus R_{[1,o]}$ decode to the same element of $\{0,1\}^{l+o'}$, namely, $X \circ h(X,R)$. Part (1.) of Property 4.8.3 therefore holds.

Now, let us establish (2.) of Property 4.8.3. Consider a $V \in \{0,1\}^o$ with $\mathsf{wt}(V) \geq \frac{1}{8}o$. Now, let us enumerate $W^{(1)}, W^{(2)}, \ldots, W^{(2^{\mathsf{wt}(V)})} \in \{0,1\}^o$ as the set of all $2^{\mathsf{wt}(V)}$ vectors in $\{0,1\}^o$ which have a 0 in all coordinates where V has a $*$. Now, observe that the distribution of $\mathsf{Corrupt}_V(\mathsf{Enc}(X, R))$ over $R_1, R_2, \ldots, R_o$ taken i.i.d. uniformly in $\{0, 1\}$ is identical to the distribution of

$$\mathcal{C}^{\mathsf{hash}}(X \circ h(X, R_{[o+1,r]})) \oplus W,$$

where W is chosen uniformly from $\left\{W^{(1)}, W^{(2)}, \ldots, W^{(2^{\mathsf{wt}(V)})}\right\}$. Now, note that for each $W^{(i)}$, there exists a corresponding $U^{(i)} \in \{0, 1\}^{o+l}$ such that under the nearest-codeword decoding of $\mathcal{C}^{\mathsf{hash}}$,

$$\mathcal{C}^{\mathsf{hash}}(X \circ h(X, R_{[o+1,r]}))) \oplus W^{(i)}$$

decodes to $(X \circ h(X, R_{[o+1,r]})) \oplus U^{(i)}$. Thus, we have that

$$\begin{aligned} &\Pr_{R \sim \mathrm{Unif}(\{0,1\}^r)} \left[\mathsf{Dec}(\mathsf{Corrupt}_V(\mathsf{Enc}(X, R)), R) \notin \{X, \perp\}\right] \\ = &\Pr_{\substack{R_{o+1} \ldots, R_r \sim \mathrm{Unif}(\{0,1\}) \\ 1 \leq i \leq 2^{\mathsf{wt}(V)}}} \left[U^{(i)} \neq (0, 0, \ldots, 0) \text{ AND } h\left(X \oplus U^{(i)}_{[1,l]}\right) = h\left(X, R_{[o+1,r]}\right) \oplus U^{(i)}_{[l+1,l+o]}\right], \end{aligned}$$

which, by Property 4.8.4, is at most $2^{-o'}$, thereby establishing (2.) of Property 4.8.3. □

4.8.2 Information Hiding

We now describe how the encoded control information bits are sent within each mini-block. Recall that in the m^{th} iteration, Alice chooses a fresh random seed R^A taken from the shared randomness str and computes her encoded control information $\mathsf{Enc}(\mathsf{ctrl}_A^{(m)}, R^A)$. Similarly, Bob chooses R^B and computes $\mathsf{Enc}(\mathsf{ctrl}_B^{(m)}, R^B)$. Recall that R^A, R^B are known to both Alice and Bob.

As discussed previously, the control information bits in each mini-block are not sent contiguously. Rather, the locations of the control information bits within each b'-sized mini-block are hidden from the oblivious adversary by using the shared randomness to agree on a designated set of $2c\log(1/\epsilon')$ locations. In particular, the locations of the control information bits sent by Alice and Bob during the m^{th} iteration are given by the variables $z_{m,i}^A$ and $z_{m,i}^B$ ($i = 1, \ldots, c\log(1/\epsilon')$), respectively. For each m, these variables are chosen randomly at the beginning using $O(\log^2(1/\epsilon'))$ fresh random bits from the preshared string str. Since there are N_{iter} iterations, this will require a total of $\Theta(N_{\mathsf{iter}} \cdot \log^2(1/\epsilon')) = \Theta(n\epsilon' \log^2(1/\epsilon'))$ random bits from str.

Thus, Alice sends the $c\log(1/\epsilon')$ bits of $\mathsf{Enc}(\mathsf{ctrl}_A^{(m)}, R^A)$ in positions $z_{m,i}^A$ ($i = 1, \ldots, c\log(1/\epsilon')$) of the mini-block of the m^{th} iteration, and similarly, Bob sends the bits of $\mathsf{Enc}(\mathsf{ctrl}_B^{(m)}, R^B)$

in positions $z^B_{m,i}$ $(i = 1, \ldots, c\log(1/\epsilon'))$. Meanwhile, Bob listens for Alice's encoded control information in positions $z^A_{m,i}$ of the mini-block and assembles the received bits as a string $Y \in \{0,1\}^{c\log(1/\epsilon')}$, after which Bob tries to decode Alice's control information by computing $\mathsf{Dec}(Y, R^A)$. Similarly, Alice listens for Bob's encoded control information in locations $z^B_{m,i}$ and tries to decode the received bits.

After each iteration, Alice and Bob use their decodings of each other's control information to decide how to proceed. This is described in detail in Section 4.9.

Remark 30. *The information hiding provided by the randomization of $z^A_{m,i}$ and $z^B_{m,i}$ $(i = 1, \ldots, c\log(1/\epsilon'))$ ensures that an oblivious adversary generally needs to corrupt a constant fraction of bits in a mini-block in order to corrupt a constant fraction of either party's encoded control information bits in that mini-block. Along with Property 4.8.3, this statement is used to prove Lemma 18.*

4.9 Flow of the Protocol and Backtracking

Throughout $\Pi^{\mathrm{oblivious}}_{\mathrm{enc}}$, each party maintains a state that indicates whether both parties are in sync as well as parameters that allow for backtracking in the case that the parties are not in sync. After each iteration, Alice and Bob use their decodings of the other party's control information from that iteration to update their states. We describe the flow of the protocol in detail.

Alice and Bob maintain binary variables sync_A and sync_B, respectively, which indicate the players' individual perceptions of whether they are in sync. Note that $\mathsf{sync}_A = 1$ implies $k_A = 1$ (and similarly, $\mathsf{sync}_B = 1$ implies $k_B = 1$). Moreover, in the case that $\mathsf{sync}_A = 1$ (resp. $\mathsf{sync}_B = 1$), the variable speak_A (resp. speak_B) indicates whether Alice (resp. Bob) speaks in the c_A^{th} (resp. c_B^{th}) block of Π_{blk}, based on the transcript thus far.

Let us describe the protocol from Alice's point of view, as Bob's procedure is analogous. Note that after each iteration, Alice attempts to decode Bob's control information for that iteration. We say that Alice *successfully decodes* Bob's control information if the decoding procedure (see Section 4.8.1) does not output $\perp$. In this case, we write the output of the control information decoder (for the m^{th} iteration) as

$$\widetilde{\mathsf{ctrl}}_B^{(m)} = \left(\widetilde{h}^{(m)}_{B,c}, \widetilde{h}^{(m)}_{B,x}, \widetilde{h}^{(m)}_{B,k}, \widetilde{h}^{(m)}_{B,T}, \widetilde{h}^{(m)}_{B,\mathtt{MP1}}, \widetilde{h}^{(m)}_{B,\mathtt{MP2}}, \widetilde{j}_B, \widetilde{\mathsf{sync}}_B\right).$$

We now split into two cases, based on whether $\mathsf{sync}_A = 1$ or $\mathsf{sync}_A = 0$.

$\underline{\mathsf{sync}_A = 1:}$

The general idea is that whenever Alice thinks she is in sync with Bob (i.e., $\mathsf{sync}_A = 1$), she either (a.) *listens* for data bits from Bob while updating her estimate $\widetilde{x}$ of block c_A of Π_{blk}, if $\mathsf{speak}_A = 0$, or (b.) *transmits*, as data bits of the next iteration, the $(m \bmod 2s)$-th chunk of the encoding of x (the c_A-th B-block of Π_{blk}) under $\mathcal{C}^{\mathsf{rateless}}$, if $\mathsf{speak}_A = 1$ (see Section 4.7.4 for details).

If Alice is *listening* for data bits, then Alice expects that $k_A = k_B = 1$ and either (1.) $c_A = c_B$, $T_A = T_B$ or (2.) $c_A = c_B + 1$, $T_B = T_A[1 \dots (c_B - 1)B]$. Condition (1.) is expected to hold if Alice has still not managed to decode the B-block x that Bob is trying to relay, while (2.) is expected if Alice has managed to decode x and has advanced her transcript but Bob has not yet realized this.

On the other hand, if Alice is *transmitting* data bits, then Alice expects that $k_A = k_B = 1$, as well as either (1.) $c_A = c_B$, $T_A = T_B$, or (2.) $c_B = c_A + 1$, $T_B = T_A \circ x$, or (3.) $c_A = c_B + 1$, $T_B = T_A[1 \dots (c_B - 1)B]$. Condition (1.) is expected to hold if Bob is still listening for data bits and has not yet decoded Alice's x, while (2.) is expected to hold if Bob has already managed to decode x and advanced his block index and transcript, and (3.) is expected to hold if Bob has been transmitting data bits to Alice (for the $(c_A - 1)$-th B-block of Π_{blk}), but Bob has not realized that Alice has decoded the correct B-block and moved on.

Now, if Alice manages to successfully decode Bob's control information in the most recent iteration, then Alice checks whether the hashes $\widetilde{h}^{(m)}_{B,c}$, $\widetilde{h}^{(m)}_{B,k}$, $\widetilde{h}^{(m)}_{B,T}$, $\widetilde{h}^{(m)}_{B,x}$, as well as $\widetilde{\mathsf{sync}}_B$ are consistent with Alice's expectations (as outlined in the previous two paragraphs). If not, then Alice sets $\mathsf{sync}_A = 0$. Otherwise, Alice proceeds normally.

Remark 31. *Note that in general, if a party is trying to transmit the contents x of a B-block and the other party is trying to listen for x, then there is a delay of at least one iteration between the time that the listening party decodes x and the time that the transmitting party receives control information suggesting that the other party has decoded x. However, since $b/B = O(\epsilon')$, the rate loss due to this delay turns out to be just $O(\epsilon')$.*

$\underline{\mathsf{sync}_A = 0}$:

Now, we consider what happens when Alice believes she is out of sync (i.e., $\mathsf{sync}_A = 0$). In this case, Alice uses a meeting point based backtracking mechanism along the lines of [Sch92] and [Hae14]. We sketch the main ideas below:

Specifically, Alice keeps a backtracking parameter k_A that is initialized as 1 when Alice first believes she has gone out of sync and increases by 1 each iteration thereafter.

(Note that k_A is also maintained when $\mathsf{sync}_A = 1$, but it is always set to 1 in this case.) Alice also maintains a counter E_A that counts the number of discrepancies between k_A and k_B, as well as *meeting point counters* v_1 and v_2. The counters E_A, v_1, v_2 are initialized to zero when Alice first sets sync_A to 0.

The parameter k_A measures the amount by which Alice is willing to backtrack in her transcript T_A. More specifically, Alice creates a *scale* $\widetilde{k}_A = 2^{\lfloor \log_2 k_A \rfloor}$ by rounding k_A to the largest power of two that does not exceed it. Then, Alice defines two *meeting points* $\mathtt{MP1}$ and $\mathtt{MP2}$ on this scale to be the two largest multiples of $\widetilde{k}_A B$ not exceeding $|T_A|$. More precisely, $\mathtt{MP1} = \widetilde{k}_A B \left\lfloor \frac{|T_A|}{\widetilde{k}_A B} \right\rfloor$ and $\mathtt{MP2} = \mathtt{MP1} - \widetilde{k}_A B$. Alice is willing to rewind her transcript to either one of $T_A[1 \ldots \mathtt{MP1}]$ and $T_A[1 \ldots \mathtt{MP2}]$, the last two positions in her transcript where the number of B-blocks of Π_{blk} that have been simulated is an integral multiple of $\widetilde{k}_A$.

If Alice is able to successfully decode Bob's control information, then she checks $\widetilde{h}_{B,k}^{(m)}$. If it does not agree with the hash of k_A (suggesting that $k_A \neq k_B$), then Alice increments E_A. Alice also increments E_A if $\widetilde{\mathsf{sync}}_B = 1$.

Otherwise, if $\widetilde{h}_{B,k}^{(m)}$ matches her computed hash of k_A, then Alice checks whether either of $\widetilde{h}_{B,\mathtt{MP1}}^{(m)}, \widetilde{h}_{B,\mathtt{MP2}}^{(m)}$ matches the appropriate hash of $T_A[1 \ldots \mathtt{MP1}]$. If so, then Alice increments her counter v_1, which counts the number of times her *first* meeting point matches one of the meeting points of Bob. If not, then Alice then checks whether either of $\widetilde{h}_{B,\mathtt{MP1}}^{(m)}, \widetilde{h}_{B,\mathtt{MP2}}^{(m)}$ matches the hash of $T_A[1 \ldots \mathtt{MP2}]$ and if so, she increments her counter v_2, which counts the number of times her *second* meeting point matches one of the meeting points of Bob.

In the case that Alice is not able to successfully decode Bob's control information from the most recent iteration (i.e., the decoder outputs $\perp$), she increments E_A.

Regardless of which of the above scenarios holds, Alice then increases k_A by 1 and updates $\widetilde{k}_A$, $\mathtt{MP1}$, and $\mathtt{MP2}$ accordingly.

Next, Alice checks whether to initiate a *transition*. Alice only considers making a transition if $k_A = \widetilde{k}_A \geq 2$ (i.e., k_A is a power of two and is ≥ 2). Alice first decides whether to initiate a *meeting point transition*. If $v_1 \geq 0.2k_A$, then Alice rewinds T_A to $T_A[1 \ldots \mathtt{MP1}]$ and resets $k_A, \widetilde{k}_A, \mathsf{sync}_A$ to 1 and E_A, v_1, v_2 to 0. Otherwise, if $v_2 \geq 0.2k_A$, then Alice rewinds T_A to $T_A[1 \ldots \mathtt{MP2}]$ and again resets $k_A, \widetilde{k}_A, \mathsf{sync}_A$ to 1 and E_A, v_1, v_2 to 0.

If Alice has not made a meeting point transition, then Alice checks whether $E_A \geq 0.2k_A$. If so, Alice undergoes an *error transition*, in which she simply resets $k_A, \widetilde{k}_A, \mathsf{sync}_A$ to 1 and E_A, v_1, v_2 to 0 (without modifying T_A).

Finally, if $k_A = \widetilde{k}_A \geq 2$ but Alice has not made any transition, then she simply resets v_1, v_2 to 0.

Remark 32. *The idea behind meeting point transitions is that if the transcripts T_A and T_B have not diverged too far, then there is a common meeting point up to which the transcripts of Alice and Bob agree. Thus, during the control information of each iteration, both Alice and Bob send hash values of their two meeting points in the hope that there is a match. For a given scale $\widetilde{k}_A$, there are $\widetilde{k}_A$ hash comparisons that are generated. If at least a constant fraction of these comparisons result in a match, then Alice decides to backtrack and rewind her transcript to the relevant meeting point. This ensures that in order for an adversary to cause Alice to backtrack incorrectly, he must corrupt the control information in a constant fraction of iterations.*

4.10 Pseudocode

We are now ready to provide the pseudocode for the protocol $\Pi_{\text{enc}}^{\text{oblivious}}$, which follows the high-level description outlined in Section 4.7.1 and is shown in Figure 4.3. The pseudocode for the helper functions `AliceControlFlow`, `AliceUpdateSyncStatus`, `AliceUpdateControl`, `AliceDecodeControl`, `AliceAdvanceBlock`, `AliceUpdateEstimate`, and `AliceRollback` for Alice is also displayed. Bob's functions `BobControlFlow`, `BobUpdateSyncStatus`, `BobUpdateControl`, `BobDecodeControl`, `BobAdvanceBlock`, `BobUpdateEstimate`, and `BobRollback` are almost identical, except that "A" subscripts are replaced with "B" and are thus omitted. Furthermore, the function `InitializeSharedRandomness` is the same for Alice and Bob.

4.11 Analysis of Coding Scheme for Oblivious Adversarial Channels

Now, we show that the coding scheme presented in Figure 4.3 allows one to tolerate an error fraction of ϵ under an oblivious adversary with high probability.

Global parameters

$b' = b + 2c\log(1/\epsilon')$ $\quad$ $\Pi_{\text{blk}} = B$-blocked simulating protocol for Π (see Lemma 16)

$N_{\mathsf{iter}} = n'(1 + \Theta(\epsilon\log(1/\epsilon)))/b$ $\quad$ $l' = \Theta(n\epsilon' \operatorname{polylog}(1/\epsilon'))$

$\epsilon' = \epsilon^2$ $\quad$ $\mathcal{C}^{\mathsf{hash}} : \{0,1\}^{\Theta(\log(1/\epsilon'))} \to \{0,1\}^{\Theta(\log(1/\epsilon'))}$ (see Section 4.8.1)

$b = s = \Theta(1/\epsilon')$ $\quad$ $\mathcal{C}^{\mathsf{exchange}} : \{0,1\}^{l'} \to \{0,1\}^{10\epsilon N_{\mathsf{iter}} b'}$ (see Section 4.7.3)

$B = sb$ $\quad$ $\mathcal{C}^{\mathsf{rateless}} : \{0,1\}^{B} \to \{0,1\}^{2B}$ (see Lemma 17)

Alice $\qquad\qquad$ **Bob**

Random string exchange

```
Choose a random string $\mathsf{str} \in \{0,1\}^{l'}$
$w \leftarrow \mathcal{C}^{\mathsf{exchange}}(\mathsf{str})$
```

$w \longrightarrow \widetilde{w}$

```
$w' \leftarrow$ nearest codeword of $\mathcal{C}^{\mathsf{exchange}}$ to $\widetilde{w}$
$\mathsf{str} \leftarrow (\mathcal{C}^{\mathsf{exchange}})^{-1}(w')$
```

Initialization

```
$T_A \leftarrow \emptyset$; $x \leftarrow nil$
$k_A, \widetilde{k}_A, c_A, \mathsf{sync}_A \leftarrow 1$
$E_A, v_1, v_2, j_A, \mathsf{speak}_A, a, m, \mathtt{MP1}, \mathtt{MP2} \leftarrow 0$
InitializeSharedRandomness()
if Alice speaks in the first block of $\Pi_{\text{blk}}$ then
    $\mathsf{speak}_A \leftarrow 1$
    $x \leftarrow$ contents of first block of $\Pi_{\text{blk}}$
    $y = y_0 \circ y_1 \circ \cdots \circ y_{2s-1} \leftarrow \mathcal{C}^{\mathsf{rateless}}(x)$
end if
```

```
$T_B \leftarrow \emptyset$; $x \leftarrow nil$
$k_B, \widetilde{k}_B, c_B, \mathsf{sync}_B \leftarrow 1$
$E_B, v_1, v_2, j_B, \mathsf{speak}_B, a, m, \mathtt{MP1}, \mathtt{MP2} \leftarrow 0$
InitializeSharedRandomness()
if Bob speaks in the first block of $\Pi_{\text{blk}}$ then
    $\mathsf{speak}_B \leftarrow 1$
    $x \leftarrow$ contents of first block of $\Pi_{\text{blk}}$
    $y_0 \circ y_1 \circ \cdots \circ y_{2s-1} \leftarrow \mathcal{C}^{\mathsf{rateless}}(x)$
end if
```

Block transmission (repeat N_{iter} times)

```
AliceUpdateControl()
Send $\mathbf{r}[i]$ in slot $z^A_{m,i}$ for $i = 1, \ldots, (b' - b)/2$
Listen during slots $\widetilde{z}^B_{m,i}$ for $i = 1, \ldots, (b' - b)/2$ and write bits to $\widetilde{\mathbf{r}}$
if $\mathsf{sync}_A = 1$ and $\mathsf{speak}_A = 1$ then
    Send bits of $y_{m \bmod 2s}$ in the $b$ remaining slots
else
    Listen during $b$ remaining slots and store as $g_A$
end if
AliceControlFlow()
```

```
BobUpdateControl()
Send $\mathbf{r}[i]$ in slot $z^B_{m,i}$ for $i = 1, \ldots, (b' - b)/2$
Listen during slots $\widetilde{z}^A_{m,i}$ for $i = 1, \ldots, (b' - b)/2$ and write bits to $\widetilde{\mathbf{r}}$
if $\mathsf{sync}_B = 1$ and $\mathsf{speak}_B = 1$ then
    Send bits of $y_{m \bmod 2s}$ in the $b$ remaining slots
else
    Listen during $b$ remaining slots and store as $g_B$
end if
BobControlFlow()
```

End of repeat

Figure 4.3: Encoded protocol $\Pi^{\text{oblivious}}_{\text{enc}}$ for tolerating oblivious adversarial errors.

Algorithm 1 Procedure for Alice to process received data bits and control info from a mini-block

```
function AliceControlFlow
    ▷ Update phase:
    $\widetilde{\mathsf{ctrl}}^{(m)}_B \leftarrow$ AliceDecodeControl
    if $\widetilde{\mathsf{ctrl}}^{(m)}_B \neq \perp$ then
        $\left(\widetilde{h}^{(m)}_{B,c}, \widetilde{h}^{(m)}_{B,x}, \widetilde{h}^{(m)}_{B,k}, \widetilde{h}^{(m)}_{B,T}, \widetilde{h}^{(m)}_{B,\mathtt{MP1}}, \widetilde{h}^{(m)}_{B,\mathtt{MP2}}, \widetilde{j}_B, \widetilde{\mathsf{sync}}_B\right) \leftarrow \widetilde{\mathsf{ctrl}}^{(m)}_B$
        if $\mathsf{sync}_A = 0$ then
            if $\widetilde{h}^{(m)}_{B,k} \neq hash^{(m)}_{B,k}(k_A)$ or $\widetilde{\mathsf{sync}}_B = 1$ then
                $E_A \leftarrow E_A + 1$
            else if $hash^{(m)}_{B,\mathtt{MP1}}(T_A[1 \ldots \mathtt{MP1}]) = \widetilde{h}^{(m)}_{B,\mathtt{MP1}}$ or $hash^{(m)}_{B,\mathtt{MP2}}(T_A[1 \ldots \mathtt{MP1}]) = \widetilde{h}^{(m)}_{B,\mathtt{MP2}}$ then
                $v_1 \leftarrow v_1 + 1$
            else if $hash^{(m)}_{B,\mathtt{MP1}}(T_A[1 \ldots \mathtt{MP2}]) = \widetilde{h}^{(m)}_{B,\mathtt{MP1}}$ or $hash^{(m)}_{B,\mathtt{MP2}}(T_A[1 \ldots \mathtt{MP2}]) = \widetilde{h}^{(m)}_{B,\mathtt{MP2}}$ then
                $v_2 \leftarrow v_2 + 1$
            end if
        end if
    else if $\mathsf{sync}_A = 0$ then
        $E_A \leftarrow E_A + 1$
    end if
    if $\mathsf{sync}_A = 0$ then
        $k_A \leftarrow k_A + 1$
        $\widetilde{k}_A \leftarrow 2^{\lfloor \log_2 k_A \rfloor}$
    end if
    AliceUpdateSyncStatus
    ▷ Transition phase:
    if $k_A = \widetilde{k}_A \geq 2$ and $v_1 \geq 0.2k_A$ then
        AliceRollback($\mathtt{MP1}$)
    else if $k_A = \widetilde{k}_A \geq 2$ and $v_2 \geq 0.2k_A$ then
        AliceRollback($\mathtt{MP2}$)
    else if $k_A = \widetilde{k}_A \geq 2$ and $E_A \geq 0.2k_A$ then
        $a \leftarrow (m+1) \bmod 2s$
        $k_A, \widetilde{k}_A, \mathsf{sync}_A \leftarrow 1$
        $E_A, v_1, v_2, j_A \leftarrow 0$
    else if $k_A = \widetilde{k}_A \geq 2$ then
        $v_1, v_2 \leftarrow 0$
    end if
    $\mathtt{MP1} \leftarrow \tilde{k}_A B \left\lfloor \frac{|T_A|}{\tilde{k}_A B} \right\rfloor$
    $\mathtt{MP2} \leftarrow \mathtt{MP1} - \tilde{k}_A B$
    $m \leftarrow m + 1$
end function
```

Algorithm 2 Procedure for Alice to update sync status

```
function AliceUpdateSyncStatus
    $\mathsf{sync}_A \leftarrow 0$
    if $k_A = 1$ then
        if $\widetilde{\mathsf{ctrl}}_B^{(m)} \neq \perp$ and $\widetilde{h}_{B,k}^{(m)} = hash_{B,k}^{(m)}(1)$ then
            if $\widetilde{\mathsf{sync}}_B = 0$ then
                $\mathsf{sync}_A \leftarrow 1$; $j_A \leftarrow 0$; $a \leftarrow (m+1) \bmod 2s$
            else if $hash_{B,c}^{(m)}(c_A) = \widetilde{h}_{B,c}^{(m)}$ and $hash_{B,T}^{(m)}(T_A) = \widetilde{h}_{B,T}^{(m)}$ then
                $\mathsf{sync}_A \leftarrow 1$
                if $\mathsf{speak}_A = 0$ then
                    if $j_A \leq \widetilde{j}_B$ then
                        AliceUpdateEstimate
                    else
                        $j_A \leftarrow 0$; $a \leftarrow (m+1) \bmod 2s$
                    end if
                else
                    $j_A \leftarrow \min\{j_A + 1, 2s\}$
                end if
            else if $\mathsf{speak}_A = 1$ and $hash_{B,c}^{(m)}(c_A + 1) = \widetilde{h}_{B,c}^{(m)}$ and $hash_{B,T}^{(m)}(T_A \circ x) = \widetilde{h}_{B,T}^{(m)}$ then
                $\mathsf{sync}_A \leftarrow 1$
                AliceAdvanceBlock
            else if Bob speaks in block $(c_A - 1)$ of $\Pi_{\text{blk}}$ and $hash_{B,c}^{(m)}(c_A - 1) = \widetilde{h}_{B,c}^{(m)}$ and
    $hash_{B,T}^{(m)}(T_A[1 \ldots (c_A - 2)B]) = \widetilde{h}_{B,T}^{(m)}$ and $hash_{B,x}^{(m)}(T_A[((c_A - 2)B + 1) \ldots (c_A - 1)B]) = \widetilde{h}_{B,x}^{(m)}$
    then
                $\mathsf{sync}_A \leftarrow 1$
                if $\mathsf{speak}_A = 0$ then
                    $j_A \leftarrow 0$; $a \leftarrow (m+1) \bmod 2s$
                else
                    $j_A \leftarrow \min\{j_A + 1, 2s\}$
                end if
            end if
        else if $\widetilde{\mathsf{ctrl}}_B^{(m)} = \perp$ then
            $\mathsf{sync}_A \leftarrow 1$
            if $\mathsf{speak}_A = 0$ then
                if $j_A \neq 0$ then
                    AliceUpdateEstimate
                else
                    $a \leftarrow (m+1) \bmod 2s$
                end if
            else
                $j_A \leftarrow \min\{j_A + 1, 2s\}$
            end if
        end if
    end if
end function
```

Algorithm 3 Procedure for Alice to update control information

```
function AliceUpdateControl
    $\mathsf{ctrl}_A^{(m)} \leftarrow (hash_{A,m}(c_A), hash_{A,x}^{(m)}(x), hash_{A,k}^{(m)}(k_A), hash_{A,T}^{(m)}(T_A), hash_{A,\texttt{MP1}}^{(m)}(T_A[1\ldots\texttt{MP1}]),$
        $hash_{A,\texttt{MP2}}^{(m)}(T_A[1\ldots\texttt{MP2}]), j_A, \mathsf{sync}_A)$
    $\mathbf{r} \leftarrow \mathcal{C}^{\mathsf{hash}}\left(\mathsf{ctrl}_A^{(m)} \circ hash_{A,\mathrm{ctrl}}^{(m)}\left(\mathsf{ctrl}_A^{(m)}\right)\right) \oplus V_A^{(m)}$
end function
```

Algorithm 4 Procedure for Alice to decode control information sent by Bob

```
function AliceDecodeControl
    $\mathbf{z} \leftarrow$ decoding of $\widetilde{\mathbf{r}} \oplus V_B^{(m)}$ under $\mathcal{C}^{\mathsf{hash}}$ (inverse of $\mathcal{C}^{\mathsf{hash}}$ applied to nearest codeword)
    $\mathbf{z^c} \circ \mathbf{z^h} \leftarrow \mathbf{z}$, where $\mathbf{z^c}$ has length $(b' - b)/2$

    if $hash_{B,\mathrm{ctrl}}^{(m)}(\mathbf{z^c}) = \mathbf{z^h}$ then
        return $\mathbf{z^c}$
    else
        return $\perp$
    end if
end function
```

Algorithm 5 Procedure for Alice to advance the block index and prepare for future transmissions

```
function AliceAdvanceBlock
    if $\mathsf{speak}_A = 1$ then
        $T_A \leftarrow T_A \circ x$
    else
        $T_A \leftarrow T_A \circ \widetilde{x}$
    end if

    $c_A \leftarrow c_A + 1$
    $j_A \leftarrow 0$

    if Alice speaks in block $c_A$ of $\Pi_{\mathrm{blk}}$ then
        $\mathsf{speak}_A \leftarrow 1$
        $x \leftarrow$ contents of block $c_A$ of $\Pi_{\mathrm{blk}}$
        $y = y_0 \circ y_1 \circ \cdots \circ y_{2s-1} \leftarrow \mathcal{C}^{\mathsf{rateless}}(x)$
    else
        $\mathsf{speak}_A \leftarrow 0$
        $a \leftarrow (m+1) \bmod 2s$
        $x \leftarrow nil$
    end if
end function
```

Algorithm 6 Procedure for Alice to update her estimate of the contents of the current block based on past data blocks

```
function AliceUpdateEstimate
    $\widetilde{g}_{j_A} \leftarrow g_A$
    $j_A \leftarrow j_A + 1$

    if $j_A > s$ then
        $\widetilde{x} \leftarrow$ result after decoding $(\widetilde{g}_0, \widetilde{g}_1, \ldots, \widetilde{g}_{j_A - 1})$ via the nearest codeword in $\mathcal{C}^{\mathsf{rateless}}_{a,j_A}$
        if $hash^{(m)}_{B,x}(\widetilde{x}) = \widetilde{h}^{(m)}_{B,x}$ then
            AliceAdvanceBlock
        else if $j_A = 2s$ then
            $j_A \leftarrow 0$
            $a \leftarrow (m+1) \bmod 2s$
        end if
    end if
end function
```

Algorithm 7 Procedure for Alice to backtrack to a previous meeting point

```
function AliceRollback($\mathtt{MP}$)
    $T_A \leftarrow T_A[1 \ldots \mathtt{MP}]$
    $c_A \leftarrow \frac{\mathtt{MP}}{B} + 1$
    $k_A, \widetilde{k}_A, \mathsf{sync}_A \leftarrow 1$
    $E_A, v_1, v_2, j_A \leftarrow 0$

    if Alice speaks in block $c_A$ of $\Pi_{\mathrm{blk}}$ then
        $\mathsf{speak}_A \leftarrow 1$
        $x \leftarrow$ contents of block $c_A$ of $\Pi_{\mathrm{blk}}$
        $y = y_0 \circ y_1 \circ \cdots \circ y_{2s-1} \leftarrow \mathcal{C}^{\mathsf{rateless}}(x)$
    else
        $\mathsf{speak}_A \leftarrow 0$
        $a \leftarrow (m+1) \bmod 2s$
        $x \leftarrow nil$
    end if
end function
```

Algorithm 8 Procedure for Alice and Bob to use exchanged random string to initialize hash functions, information hiding mechanism, and encoding functions for control information

```
function InitalizeSharedRandomness
    $p \leftarrow \Theta(\log(1/\epsilon'))$
    $\delta \leftarrow 2^{-\Theta(N_{\mathsf{iter}} \cdot p)}$
    $L \leftarrow \Theta(N_{\mathsf{iter}} n \log(1/\epsilon'))$
    Let $\mathsf{str} = \mathsf{str}^{\mathrm{loc}} \circ \mathsf{str}'$, where $\mathsf{str}^{\mathrm{loc}}$ is of length $\Theta(N_{\mathsf{iter}} \cdot \log^2(1/\epsilon'))$ and $\mathsf{str}'$ is of length $\Theta(\log(L/\delta))$
    $S \leftarrow \delta$-biased length $L$ pseudorandom string derived from $\mathsf{str}'$ (via the biased sample space of [NN93])
    ▷ Generate locations for information hiding in each iteration:
    for $i = 0$ to $N_{\mathsf{iter}} - 1$ do
        Choose $z^A_{i,1}, z^A_{i,2}, \ldots, z^A_{i,(b'-b)/2}, z^B_{i,1}, z^B_{i,2}, \ldots, z^B_{i,(b'-b)/2}$ to be distinct numbers in $\{1, 2, \ldots, b'\}$ using $O(\log^2(1/\epsilon'))$ fresh random bits from $\mathsf{str}^{\mathrm{loc}}$
    end for
    ▷ Set up parameters for encoding control information during each iteration
    for $i = 0$ to $N_{\mathsf{iter}} - 1$ do
        $V_A^{(i)} \leftarrow (b' - b)/2$ fresh random bits from $\mathsf{str}^{\mathrm{loc}}$
        $V_B^{(i)} \leftarrow (b' - b)/2$ fresh random bits from $\mathsf{str}^{\mathrm{loc}}$
        Initialize $hash^{(i)}_{A,\mathrm{ctrl}}, hash^{(i)}_{B,\mathrm{ctrl}}$ to an inner product hash function with output length $\Theta(\log(1/\epsilon'))$ and seed fixed as $\Theta(\log(1/\epsilon'))$ fresh random bits from $\mathsf{str}^{\mathrm{loc}}$
    end for
    ▷ Initialize hash functions for control information in each iteration:
    for $i = 0$ to $N_{\mathsf{iter}} - 1$ do
        Initialize $hash^{(i)}_{A,c}$, $hash^{(i)}_{A,x}$, $hash^{(i)}_{A,k}$, $hash^{(i)}_{A,T}$, $hash^{(i)}_{A,\mathtt{MP1}}$, $hash^{(i)}_{A,\mathtt{MP2}}$, $hash^{(i)}_{B,c}$, $hash^{(i)}_{B,x}$, $hash^{(i)}_{B,k}$, $hash^{(i)}_{B,T}$, $hash^{(i)}_{B,\mathtt{MP1}}$, $hash^{(i)}_{B,\mathtt{MP2}}$ to be inner product hash functions with output length $\Theta(\log(1/\epsilon'))$ and seed fixed using fresh random bits from $S$
    end for
end function
```

4.11.1 Protocol States and Potential Function

Let us define states for the encoded protocol $\Pi_{\text{enc}}^{\text{oblivious}}$. First, we define

$$\ell^{+} = \lfloor \max\{\ell' \in [1, \min\{|T_A|, |T_B|\}] : T_A[1 \dots \ell'] = T_B[1 \dots \ell']\rfloor$$
$$\ell^{-} = |T_A| + |T_B| - 2\ell^{+}.$$

In other words, ℓ^{+} is the length of the longest common prefix of the transcripts T_A and T_B, while ℓ^{-} is the total length of the parts of T_A and T_B that are not in the common prefix. Also recall that $\delta_{s+1}, \delta_{s+2}, \dots, \delta_{2s}$ are defined as in Lemma 17. Furthermore, we define $\delta_0, \delta_1, \dots, \delta_s = 0$ for convenience.

Now we are ready to define states for the protocol $\Pi_{\text{enc}}^{\text{oblivious}}$ as its execution proceeds.

Definition 22. *At the beginning of an iteration (the start of the code block in Figure 4.3 that is repeated N_{iter} times), the protocol is said to be in one of three possible states:*

- ***Perfectly synced** state: This occurs if $\mathsf{sync}_A = \mathsf{sync}_B = 1$, $k_A = k_B = 1$, $\ell^{-} = 0$, $c_A = c_B$, and $j_A \geq j_B$ if Alice is the sender in block $c_A = c_B$ of Π_{blk} (resp. $j_B \geq j_A$ if Bob is the sender in B-block $c_A = c_B$ of Π_{blk}). In this case, we also define $j = \min\{j_A, j_B\}$.*
- ***Almost synced** state: This occurs if $\mathsf{sync}_A = \mathsf{sync}_B = 1$, $k_A = k_B = 1$, and one of the following holds:*
  1. *$\ell^{-} = B$, $c_B = c_A + 1$, and $T_B = T_A \circ w$, where w represents the contents of the c_A-th B-block of Π_{blk}. In this case, we define $j = j_B$.*
  2. *$\ell^{-} = B$, $c_A = c_B + 1$, and $T_A = T_B \circ w$, where w represents the contents of the c_B-th B-block of Π_{blk}. In this case, we define $j = j_A$.*
  3. *$\ell^{-} = 0$, $c_A = c_B$, $j_B > j_A$, and Alice speaks in B-block $c_A = c_B$ of Π_{blk}. In this case, we define $j = j_B$.*
  4. *$\ell^{-} = 0$, $c_A = c_B$, $j_A > j_B$, and Bob speaks in B-block $c_A = c_B$ of Π_{blk}. In this case, we define $j = j_A$.*
- ***Unsynced** state: This is any state that does not fit into the above two categories.*

We also characterize the control information sent by each party during an iteration based on whether/how it is corrupted.

Definition 23. *For any given iteration, the encoded control information sent by a party is categorized as one of the following:*

- ***Sound control information**: If a party's unencoded control information for an iteration is decoded correctly by the other party (i.e., the output of* Dec *correctly retrieves the intended transmission), and no hash collisions (involving the hashes contained in the control information* $\widetilde{\mathsf{ctrl}}_A^{(m)}$ *or* $\widetilde{\mathsf{ctrl}}_B^{(m)}$*) occur, then the (encoded) control information is considered* sound.

- ***Invalid control information**: If the attempt to decode a party's unencoded control information by the other party results in a failure (i.e.,* Dec *outputs* $\perp$*), then the (encoded) control information is considered* invalid.

- ***Maliciously corrupted control information**: If a party's control information is decoded incorrectly (i.e.,* Dec *does not output* $\perp$*, but the output does not retrieve the intended transmission) or a hash collision (involving the hashes contained in the control information* $\widetilde{\mathsf{ctrl}}_A^{(m)}$ *or* $\widetilde{\mathsf{ctrl}}_B^{(m)}$*) occurs, then the (encoded) control information is considered* maliciously corrupted.

Next, we wish to define a potential function Φ that depends on the current state in the encoded protocol. Before we can do so, we define a few quantities:

Definition 24. *Suppose the protocol is in a perfectly synced state. Then, we define the quantities* err *and* inv *as follows:*

- err *is the total number of data (non-control information) bits that have been corrupted during the last* j *iterations.*

- inv *is the number of iterations among the last* j *iterations for which the control information of at least one party was invalid or maliciously corrupted.*

Definition 25. *Suppose the protocol is in an unsynced state. Then, we define* $\mathsf{mal}_{\mathrm{A}}$ *as follows: At the start of* $\Pi_{\text{enc}}^{\text{oblivious}}$*, we initialize* $\mathsf{mal}_{\mathrm{A}}$ *to 0. Whenever an iteration occurs from a state in which* $\mathsf{sync}_A = 0$*, such that either Alice's or Bob's control information during that iteration is maliciously corrupted,* $\mathsf{mal}_{\mathrm{A}}$ *increases by 1 at the end of line 21 of* `AliceControlFlow` *during that iteration. Moreover, whenever Alice undergoes a* transition *(i.e., one of the "if" conditions in lines 22-29 of* `AliceControlFlow` *is true),* $\mathsf{mal}_{\mathrm{A}}$ *resets to 0.*

The variable $\mathsf{mal}_{\mathrm{B}}$ *is defined in the obvious analagous manner.*

Definition 26. *For the sake of brevity, a variable* var_{AB} *will denote* $\mathrm{var}_A + \mathrm{var}_B$ *(e.g.,* $k_{AB} = k_A + k_B$ *and* $E_{AB} = E_A + E_B$*).*

Now, we are ready to define the potential function Φ.

Definition 27. *Let $C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_{\mathsf{inv}}, C_{\mathsf{mal}}, C, D > 0$ be suitably chosen constants (to be determined by Lemmas 21, 22, 23 and Theorem 33). Then, we define the potential function Φ associated with the execution of $\Pi_{\text{enc}}^{\text{oblivious}}$ according to the state of the protocol (see Definition 22):*

$$\Phi = \begin{cases} \ell^+(1 + C_0 H(\epsilon)) + (jb - C \cdot \mathsf{err} \cdot \log(1/\epsilon)) - Db \cdot \mathsf{inv} & \textit{perfectly synced} \\ \max\{\ell_A, \ell_B\} \cdot (1 + C_0 H(\epsilon)) - (j+1)b & \textit{almost synced} \\ \ell^+(1 + C_0 H(\epsilon)) - C_1 \ell^- + b(C_2 k_{AB} - C_3 E_{AB}) & \textit{unsynced, } (k_A, \mathsf{sync}_A) = (k_B, \mathsf{sync}_B) \\ \qquad -2C_7 B\, \mathsf{mal}_{\mathrm{AB}} - Z_1 & \\ \ell^+(1 + C_0 H(\epsilon)) - C_1 \ell^- + bC_5(-0.8 k_{AB} + 0.9 E_{AB}) & \textit{unsynced, } (k_A, \mathsf{sync}_A) \neq (k_B, \mathsf{sync}_B) \\ \qquad -C_7 B\, \mathsf{mal}_{\mathrm{AB}} - Z_2 & \end{cases},$$

where Z_1 and Z_2 are defined by:

$$Z_1 = \begin{cases} bC_4 & \textit{if } k_A = k_B = 1 \textit{ and } \mathsf{sync}_A = \mathsf{sync}_B = 1 \\ \frac{1}{2} bC_4 & \textit{if } k_A = k_B = 1 \textit{ and } \mathsf{sync}_A = \mathsf{sync}_B = 0 \\ 0 & \textit{otherwise} \end{cases},$$

and

$$Z_2 = \begin{cases} bC_6 & \textit{if } k_A = k_B = 1 \\ 0 & \textit{otherwise} \end{cases}.$$

4.11.2 Bounding Iterations with Invalid or Maliciously Corrupted Control Information

We now prove some lemmas that bound the number of iterations that can have invalid or maliciously corrupted control information.

Lemma 18. *If the fraction of errors in a mini-block is $O(1)$, say, $< \frac{1}{20}$, then with probability at least $1 - \epsilon'^2$, both parties can correctly decode and verify the control symbols sent in the block.*

Proof. Let $\nu < 1/20$ be the fraction of errors in a mini-block. Recall that Alice's control information in the mini-block consists of $c \log(1/\epsilon')$ randomly located bits. Let X be the number of these control bits that are corrupted. Note that $\mathbf{E}[X] = \nu c \log(1/\epsilon')$. Now, since the control information is protected with an error correcting code of distance $c\log(1/\epsilon')/4$, we see that Bob can verify and correctly decode Alice's control symbols as long as $X < c\log(1/\epsilon')/8$. Note that by the Chernoff bound,

$$\begin{aligned}\Pr\left(X > c\log(1/\epsilon')/8\right) &\leq e^{-\frac{\frac{c\log(1/\epsilon')}{8} - \frac{c\log(1/\epsilon')}{20}}{3}} \\ &\leq \epsilon'^{c/40},\end{aligned}$$

which is $< \epsilon'^2/2$ for a suitable constant c. Similarly, the probability that Alice fails to verify and correctly decode Bob's control symbols is $< \epsilon'^2/2$. Thus, the desired statement follows by a union bound. □

Lemma 19. *With probability at least* $1-2^{-\Omega(\epsilon' N_{\mathsf{iter}})}$*, the number of iterations in which some party's control information is invalid but neither party's control information is maliciously corrupted is* $O(\epsilon N_{\mathsf{iter}})$*.*

Proof. First of all, consider the number of iterations of $\Pi_{\text{enc}}^{\text{oblivious}}$ for which the fraction of errors within the iteration is at least $1/20$. Since the total error fraction throughout the protocol is ϵ, we know that at at most $20\epsilon N_{\mathsf{iter}}$ iterations have such an error fraction.

Next, consider any "low-error" iteration in which the error fraction is less than $1/20$. By Lemma 18, the probability that control information of some party is invalid (but neither party's control information is maliciously corrupted) is at most ϵ'^2. Then, by the Chernoff bound, the number of "low-error" iterations with invalid control information is at most $(\epsilon'^2 + \epsilon')N_{\mathsf{iter}} = O(\epsilon' N_{\mathsf{iter}})$ with probability at least $1 - 2^{-\Omega(\epsilon' N_{\mathsf{iter}})}$.

It follows that with probability at least $1 - 2^{-\Omega(\epsilon' N_{\mathsf{iter}})}$, the total number of iterations with invalid control information (but not maliciously corrupted control information) is $O(\epsilon N_{\mathsf{iter}})$. □

Lemma 20. *With probability at least* $1 - 2^{-\Omega(\epsilon'^2 N_{\mathsf{iter}})}$*, the number of iterations in which some party's control information is maliciously corrupted is at most* $O(\epsilon'^2 N_{\mathsf{iter}})$*.*

Proof. Suppose a particular party's control information is maliciously corrupted during a certain iteration (say, the m^{th} iteration). Without loss of generality, assume Alice's control information is maliciously corrupted. Then, we must have one of the following:

1. The number of corrupted bits in the encoded control information of Alice is $> \frac{1}{8}\left(\frac{b'-b}{2}\right)$, i.e., the fraction of control information bits that is corrupted is greater than $\frac{1}{8}$.

2. The number of corrupted bits in the encoded control information of Alice is $< \frac{1}{8}\left(\frac{b'-b}{2}\right)$, but a hash collision occurs for one of $h_{A,c}^{(m)}$, $h_{A,x}^{(m)}$, $h_{A,k}^{(m)}$, $h_{A,T}^{(m)}$, $h_{A,\texttt{MP1}}^{(m)}$, $h_{A,\texttt{MP2}}^{(m)}$.

Note that by Property 4.8.3, case (1.) happens with probability at most

$$2^{-\Theta(\log(1/\epsilon'))} \leq \epsilon'^2,$$

for suitable constants.

Next, we consider the probability that case (2.) occurs. By Property 4.8.2, we have that the probability of a hash collision any specific quantity among c_A, x, k_A, T_A, $T_A[1, \texttt{MP1}]$, $T_A[1, \texttt{MP2}]$ is at most $2^{-\Theta(\log(1/\epsilon'))} + 2^{-\Theta(N_{\mathsf{iter}} \log(1/\epsilon'))} \leq \epsilon'^2$ for appropriate constants. Thus, by a simple union bound, the probability that any one of the aforementioned quantities has a hash collision is at most $6\epsilon'^2$.

A simple union bound between the two events shows that the probability that Alice's control information in a given iteration is maliciously corrupted is at most $7\epsilon'^2$. Similarly, the probability that Bob's control information in a given iteration is maliciously corrupted is also at most $7\epsilon'^2$. Hence, the desired claim follows by the Chernoff bound (recall that there is limited independence, due to the fact that we use pseudorandom bits to seed hash functions, but this is not a problem due to our choice of parameters (see Section 4.8)). □

4.11.3 Evolution of Potential Function During Iterations

We now wish to analyze the evolution of the potential function Φ as the execution of the protocol proceeds. First, we define some notation that will make the analysis easier:

Definition 28. *Suppose we wish to analyze a variable* var *over the course of an iteration. For the purpose of Lemmas 21, 22, and 23, we let* var *denote the value of the variable at the start of the iteration (the start of the code block in Figure 4.3 that is repeated N_{iter} times). Moreover, we let* var′ *denote the value of the variable just after the "update phase" of the iteration (lines 2-21 of* `AliceControlFlow` *and* `BobControlFlow`*), while we will let* var″ *denote the value of the variable at the end of the iteration (at the end of the execution of* `AliceControlFlow` *and* `BobControlFlow`*).*

Moreover, we will use the notation Δvar *to denote* var″ − var*, i.e., the change in the variable over the course of an iteration. For instance,* $\Delta\Phi = \Phi'' - \Phi$.

Definition 29. *During an iteration of $\Pi_{\text{enc}}^{\text{oblivious}}$, Alice is said to undergo a* transition *if one of the "if" conditions in lines 22-29 of* `AliceControlFlow` *is true. The transition is called a* meeting point transition *(or* MP transition*) if either line 23 or line 25 is executed, while* *the transition is called an* error transition *if lines 27-29 are executed. Transitions for Bob are defined similarly, except that one refers to lines in the corresponding* `BobControlFlow` *function.*

Now, we are ready for the main analysis. Lemmas 21, 22, and 23 prove lower bounds on the change in potential, $\Delta\Phi$, over the course of an iteration, depending on (1.) the state of the protocol prior to the iteration and (2.) whether/how control information is corrupted during the iteration.

Lemma 21. *Suppose the protocol is in a perfectly synced state at the beginning of an iteration. Then, the change in potential Φ over the course of the iteration behaves as follows, according to the subsequent state (at the end of the iteration):*

1. *If the subsequent state is perfectly synced or almost synced, then:*
   - *If the control information received by both parties is sound, then $\Delta\Phi \geq b - Ct \cdot \log(1/\epsilon)$, where t is the number of data (non-control) bits that are corrupted in the next iteration.*
   - *If the control information received by at least one party is invalid or maliciously corrupted, then $\Delta\Phi \geq -Ct \cdot \log(1/\epsilon) - (D-1)b \geq -Ct \cdot \log(1/\epsilon) - \min\{C_{\mathsf{inv}}b, C_{\mathsf{mal}}B\}$.*
2. *If the subsequent state is unsynced, then $\Delta\Phi \geq -C_{\mathsf{mal}}B$.*

Proof. Assume that the protocol is currently in a perfectly synced state, and, without loss of generality, suppose that Alice is trying to send data bits corresponding to c_A-th B-block of Π_{blk} to Bob.

For the first part of the lemma statement, assume that the state after the next iteration is perfectly synced or almost synced. At the end of the iteration, Bob updates his estimate of what Alice is sending, and there are three cases:

- Case 1: Bob is still not able to decode the c_A-th B-block that Alice is sending, and j_B does not reset to zero. In this case, it is clear that j increases by 1, while err increases by t. Thus, $\Delta\Phi \geq b - Ct \cdot \log(1/\epsilon)$ if the control information received by both parties is sound, while $\Delta\Phi \geq -Ct \cdot \log(1/\epsilon) - (D-1)b$ otherwise (as inv increases by 1).
- Case 2: Bob is still not able to decode the c_A-th B-block that Alice is sending, but j_B resets to 0 (after increasing to $2s$). Then, note that if both parties receive sound

control information in the next iteration, we have

$$\Delta\Phi \geq (b - Ct \cdot \log(1/\epsilon)) + (Db \cdot \mathsf{inv} + C(\mathsf{err} + t)\log(1/\epsilon) - 2B).$$

Moreover, we must have $\mathsf{err} + t \geq \frac{1}{2}\delta_{2s}(2B) = \frac{1}{15}B$, which implies that

$$Db \cdot \mathsf{inv} + C(\mathsf{err} + t)\log(1/\epsilon) - 2B \geq 0,$$

as desired (for suitably large C).

On the other hand, suppose some party receives invalid or maliciously corrupted control information in the next iteration. Then,

$$\Delta\Phi \geq (-Ct \cdot \log(1/\epsilon) - (D-1)b) + (Db \cdot (\mathsf{inv} + 1) + C(\mathsf{err} + t)\log(1/\epsilon) - 2B).$$

Thus, to prove the lemma, it suffices to show

$$Db \cdot (\mathsf{inv} + 1) + C(\mathsf{err} + t)\log(1/\epsilon) - 2B \geq 0. \tag{4.1}$$

Let j_0 be the last/most recent value of j_B occurring after an iteration in which Bob receives sound control information (or $j_0 = 0$ if such an iteration did not occur). Thus, in the last $2s - j_0 - 1$ iterations, Bob has not received sound control information. This implies that $\mathsf{inv} \geq 2s - j_0 - 1$ and $\mathsf{err} \geq \frac{1}{2}\delta_{j_0} j_0 b$. Thus, we reduce (4.1) to showing the following:

$$D(2s - j_0)b + \frac{C}{2}\delta_{j_0} j_0 b \cdot \log(1/\epsilon) - 2B \geq 0. \tag{4.2}$$

Note that if $j_0 \leq s$, then $\delta_{j_0} = 0$, and so the lefthand side of (4.2) is at least

$$Dsb - 2B = (D-2)B \geq 0,$$

as desired. Hence, we now assume that $j_0 > s$. Then, by Lemma 17, $\delta_{j_0} \geq H^{-1}\left(\frac{j_0 - s}{j_0} - \frac{1}{4s}\right)$ (recall that H^{-1} is the unique inverse of H that takes values in $[0, 1/2]$). Thus, (4.2) reduces to showing

$$\frac{C}{2}H^{-1}\left(\frac{j_0 - s}{j_0} - \frac{1}{4s}\right)\log(1/\epsilon) \geq D - \frac{2s(D-1)}{j_0}. \tag{4.3}$$

Note that if $j_0 \leq \frac{D-1}{D} \cdot 2s$, then (4.3) is clearly true, as the righthand side of (4.3) is nonpositive.

If $j_0 > \frac{D-1}{D} \cdot 2s$, then note that the righthand side of (4.3) is at most 1 (since $j_0 \leq 2s$), while the lefthand side is at least

$$\frac{C}{2} H^{-1} \left(1 - \frac{s}{\frac{D-1}{D} \cdot 2s} - \frac{\epsilon'}{4} \right) \log(1/\epsilon) \geq \frac{C}{2} H^{-1} \left(\frac{D-2}{2(D-1)} - \frac{\epsilon'}{4} \right) \log(1/\epsilon)$$
$$\geq 1.$$

- <u>Case 3</u>: Bob manages to decode the c_A-th B-block and updates his transcript. Then, the protocol either transitions to an almost synced state or remains in a perfectly synced state (if Alice receives maliciously corrupted control information indicating that Bob has already advanced his transcript). Thus,

  $$\Delta\Phi \geq (b - Ct \cdot \log(1/\epsilon)) + B(1 + C_0 H(\epsilon)) + C(\mathsf{err} + t) \log(1/\epsilon) - (j+2)b + Db \cdot \mathsf{inv},$$

  Hence, it suffices to show that

  $$B(1 + C_0 H(\epsilon)) + C(\mathsf{err} + t) \log(1/\epsilon) - (j+2)b + Db \cdot \mathsf{inv} \geq 0. \qquad (4.4)$$

  Note that $j \geq s$. Suppose j_0 is the last/most recent value of j_B occurring after an iteration in which Bob receives sound control information (or $j_0 = 0$ if such an iteration did not occur). Then, $\mathsf{inv} \geq j - j_0$. Hence, (4.4) reduces to showing

  $$B(1 + C_0 H(\epsilon)) + C \cdot \mathsf{err}' \cdot \log(1/\epsilon) - (j+2)b + Db(j - j_0) \geq 0. \qquad (4.5)$$

  Note that if $j_0 \leq s$, then the lefthand side of (4.5) is at least

  $$\begin{aligned} B(1 + C_0 H(\epsilon)) - (j+2)b + Db(j - s) &\geq B(1 + C_0 H(\epsilon)) + (D-1)jb \\ &\quad - DB - 2b \\ &\geq B(1 + C_0 H(\epsilon)) + (D-1)B \\ &\quad - DB - 2b \\ &\geq B(C_0 H(\epsilon) - 2\epsilon') \\ &\geq 0, \end{aligned}$$

  as desired.

  Now, assume $j_0 > s$. Let ϵ_0 be the fraction of errors in the first $j_0 b$ data bits sent since Alice and Bob became perfectly synced (or since the last reset). Then,

  $$\mathsf{err}' \geq \epsilon_0 j_0 b.$$

Hence, the lefthand side of (4.5) is at least

$$B(1 + C_0 H(\epsilon)) + j_0 b(C\epsilon_0 \log(1/\epsilon) - 1) - 2b + (D-1)b(j - j_0). \tag{4.6}$$

Note that if $C\epsilon_0 \log(1/\epsilon) \geq 1$, then the above quantity is clearly nonnegative, as $B \geq b/\epsilon' \geq 2b$. Thus, let us assume that $C\epsilon_0 \log(1/\epsilon) < 1$. Now, recall from our choice of $\mathcal{C}^{\mathsf{rateless}}$ and the fact that Bob had not successfully decoded the blocks sent by Alice before the current iteration, we have $\epsilon_0 \geq \frac{1}{2}\delta_{j_0}$, which implies that

$$\frac{j_0 - s}{j_0} - \frac{1}{4s} = H(\delta_{j_0}) \leq H(2\epsilon_0).$$

Hence,

$$j_0 \leq \frac{s}{1 - H(2\epsilon_0) - \frac{1}{4s}}.$$

Now, (4.6) is at least

$$\begin{aligned}
&B(1 + C_0 H(\epsilon)) + \frac{B(C\epsilon_0 \log(1/\epsilon) - 1)}{1 - H(2\epsilon_0) - \frac{1}{4s}} - 2b \\
&\geq B(1 + C_0 H(\epsilon)) + \frac{B(C\epsilon_0 \log(1/\epsilon) - 1)}{1 - H(2\epsilon_0) - \frac{\epsilon'}{4}} - 2b \\
&\geq B\left(1 + C_0 H(\epsilon) - (1 - C\epsilon_0 \log(1/\epsilon))\left(1 + H(2\epsilon_0) + \frac{\epsilon'}{4} + 2\left(H(2\epsilon_0) + \frac{\epsilon'}{4}\right)^2\right) - 2\epsilon'\right) \\
&\geq B\left(1 + C_0 H(\epsilon) - 1 - H(2\epsilon_0) - \frac{\epsilon'}{4} - 2H(2\epsilon_0)^2 - \epsilon' H(2\epsilon_0) - \frac{\epsilon'^2}{8} + C\epsilon_0 \log(1/\epsilon) - 2\epsilon'\right) \\
&\geq B\left(C_0 H(\epsilon) - 4\epsilon' - 3H(2\epsilon_0) + C\epsilon_0 \log(1/\epsilon)\right). \qquad (4.7)
\end{aligned}$$

Note that if $\epsilon_0 < \epsilon$, then (4.7) is bounded from below by

$$\begin{aligned}
B(C_0 H(\epsilon) - 4\epsilon' - 3H(2\epsilon)) &\geq B\left((4H(\epsilon) - 4\epsilon') + ((C_0 - 4)H(\epsilon) - 3H(2\epsilon))\right) \\
&\geq 0,
\end{aligned}$$

since $H(\epsilon) \geq \epsilon \geq \epsilon'$, $C_0 \geq 10$, and $2H(\epsilon) \geq H(2\epsilon)$.

On the other hand, if $\epsilon_0 \geq \epsilon$, then (4.7) is bounded from below by

$$B\left((4H(\epsilon) - 4\epsilon') + (C\epsilon_0 \log(1/\epsilon_0) - 3H(2\epsilon_0))\right) \geq 0,$$

as long as $C \geq 10$.

This completes the proof of the first part of the lemma.

Next, we prove the second part of the lemma. Assume that the protocol is currently in a perfectly synced state and that the subsequent state is unsynced. Then, note that the control information of at least one party must be maliciously corrupted. Observe that $k''_A = k''_B = 1$, and $\ell^{-''} \leq 2B$, while $E''_A = E''_B = 0$. Thus, if $\mathsf{sync}''_A = \mathsf{sync}''_B$, then

$$\Delta\Phi \geq -jb - 2C_1 B + 2bC_2 - bC_4 \geq -C_{\mathsf{mal}} B,$$

while if $\mathsf{sync}''_A \neq \mathsf{sync}''_B$, then

$$\Delta\Phi \geq -jb - 2C_1 B - 1.6bC_5 - bC_6 \geq -C_{\mathsf{mal}} B,$$

since $jb \leq 2B$. □

Lemma 22. *Suppose the protocol is in an almost synced state at the beginning of an iteration. Then, the change in potential Φ over the course of the iteration behaves as follows, according to the control information received during the iteration:*

- *If the control information received by both parties is sound, then $\Delta\Phi \geq b$.*
- *If the control information received by at least one party is invalid, but neither party's control information is maliciously corrupted, then the potential does not change, i.e., $\Delta\Phi \geq -b \geq -C_{\mathsf{inv}} b$.*
- *If the control information received by at least one party is maliciously corrupted, then $\Delta\Phi \geq -C_{\mathsf{mal}} B$.*

Proof. Assume the protocol lies in an almost synced state. We consider the following cases, according to the subsequent state in the protocol.

- Case 1: The subsequent state is perfectly synced. Then, we must have that $\Delta\Phi \geq (j+1)b \geq b$.
- Case 2: The subsequent state is also almost synced. Then, note that the control information received by some party must be invalid or maliciously corrupted. Moreover, since $\max\{\ell_A, \ell_B\}$ remains unchanged and j can increase by at most 1, it follows that $\Delta\Phi \geq -b \geq -C_{\mathsf{mal}} B$.
- Case 3: The subsequent state is unsynced. Then, observe that the control information received by some party must be maliciously corrupted. Note that $\ell^{+''} \geq$

$\max\{\ell_A, \ell_B\} - B$, and $\ell^{-\prime\prime} \leq 3B$. Moreover, $k''_A = k''_B = 1$. Therefore, if $\mathsf{sync}''_A = \mathsf{sync}''_B$, then

$$\begin{aligned}\Delta\Phi &\geq -B(1 + C_0 H(\epsilon)) - 3C_1 B + 2bC_2 - bC_4 \\ &\geq -C_{\mathsf{mal}} B,\end{aligned}$$

while if $\mathsf{sync}''_A \neq \mathsf{sync}''_B$, then

$$\begin{aligned}\Delta\Phi &\geq -B(1 + C_0 H(\epsilon)) - 3C_1 B - 1.6bC_5 - bC_6 \\ &\geq -C_{\mathsf{mal}} B,\end{aligned}$$

as desired.

□

Lemma 23. *Suppose the protocol is in an unsynced state at the beginning of an iteration. Then, the change in potential Φ over the course of the iteration behaves as follows, according to the control information received during the iteration:*

1. *If the control information received by both parties is sound, then $\Delta\Phi \geq b$.*
2. *If the control information received by at least one party is invalid, but neither party's control information is maliciously corrupted, then $\Delta\Phi \geq -C_{\mathsf{inv}} b$.*
3. *If the control information received by at least one party is maliciously corrupted, then $\Delta\Phi \geq -C_{\mathsf{mal}} B$.*

Proof. We consider several cases, depending on the values of k_A, k_B and what transitions occur before the end of the iteration.

- Case 1: $k_A \neq k_B$.
  - Subcase 1: No transitions occur before the start of the next iteration.
    - a.) If the control information sent by both parties is sound or invalid, then note that $\Delta k_A = \Delta E_A \in \{0, 1\}$ and $\Delta k_B = \Delta E_B \in \{0, 1\}$. Also, at least one of Δk_A, Δk_B must be 1, while ℓ^+, ℓ^-, $\mathsf{mal}_{\mathrm{AB}}$ remain unchanged. Moreover, the state will remain an unsynced state with $k''_A \neq k''_B$. Therefore,

$$\Delta\Phi \geq b(-0.8C_5 + 0.9C_5) \geq b.$$

b.) If at least one party's control information is maliciously corrupted and $k_A, k_B > 1$, then note that the state at the beginning of the next iteration will also be unsynced with $k''_A \neq k''_B$. Also, observe that $\Delta k_A = \Delta k_B = 1$, while ℓ^+, ℓ^- remain unchanged. Thus,

$$\Delta\Phi \geq 2b(-0.8C_5) - 2C_7 B \geq -C_{\mathsf{mal}} B.$$

c.) If at least one party's control information is maliciously corrupted and one of k_A, k_B is 1, then without loss of generality, assume $k_A = 1$ and $k_B > 1$. Note that k_B increases by 1. Also, if k_A does not increase, then ℓ^- can increase by at most B. Hence,

$$\Delta\Phi \geq -0.8bC_5 - 2C_7 B - \max\{0.8bC_5, C_1 B\} \geq -C_{\mathsf{mal}} B.$$

- Subcase 2: Only one of Alice and Bob undergoes a transition before the start of the next iteration. Without loss of generality, assume that Alice makes the transition. Also, let

$$P_1 = \begin{cases} 0.2C_7(k_A+1)B - (1 + C_0 H(\epsilon) + C_1)k_A B & \text{if Alice has an MP trans.} \\ 0 & \text{otherwise} \end{cases} \tag{4.8}$$

Note that $P_1 \geq 0$ for a suitable choice of constants C_0, C_1, C_7. Also observe that if $k_A \geq 3$, then

$$E_A \leq \frac{1}{2}(k_A+1) - 1 + 0.2 \cdot \frac{1}{2}(k_A+1) = 0.6k_A - 0.4 \leq 0.7(k_A - 1), \tag{4.9}$$

since an error transition did not occur when Alice's backtracking parameter was equal to $\frac{1}{2}(k_A+1)$, and an additional $\frac{1}{2}(k_A+1)-1$ iterations have occurred since then. Note that (4.9) also holds if $k_A < 3$ since it must be the case that $E_A = 0$.

a.) Suppose the control information sent by each party is sound. Then, note that $(k''_A, \mathsf{sync}''_A) \neq (k''_B, \mathsf{sync}''_B)$. Moreover, if Alice's transition is a meeting point transition, then we must have $\mathsf{mal}_{\mathsf{A}} \geq 0.2(k_A + 1)$, and the transition can cause Alice's transcript T_A to be rewound by at most $k_A B$ bits, which implies that $\Delta\ell^- \leq k_A B$ and $\Delta\ell^+ \geq -k_A B$.

Thus, if $k_A, k_B > 1$, then by (4.9), we have

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A-1) - 0.9bC_5E_A + (-0.8bC_5 + 0.9bC_5) + P_1\\ &\geq 0.8bC_5(k_A-1) - 0.9bC_5 \cdot 0.7(k_A-1) + 0.1bC_5\\ &\geq 0.27bC_5\\ &\geq b,\end{aligned}$$

while if $k_A = 1$ and $k_B > 1$, then

$$\begin{aligned}\Delta\Phi &\geq -0.8bC_5 + 0.9bC_5 + P_1\\ &\geq 0.1bC_5\\ &\geq b.\end{aligned}$$

Finally, if $k_B = 1$, then $k_A > 1$ and so, by (4.9), we have

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A-1) - 0.9bC_5E_A - bC_6 + P_1\\ &\geq 0.8bC_5(k_A-1) - 0.9bC_5 \cdot 0.7(k_A-1) - bC_6\\ &\geq (0.17C_5 - C_6)b\\ &\geq b.\end{aligned}$$

b.) Suppose the control information sent by at least one party is invalid, but neither party's control information is maliciously corrupted. Again, we note that if Alice's transition is a meeting point transition, then $\mathsf{mal}_{\mathrm{A}} \geq 0.2(k_A+1)$ and $\Delta\ell^- \leq k_A B$ and $\Delta\ell^+ \geq -k_A B$.
First, suppose that $k_B = \mathsf{sync}_B = 1$ and that Bob receives invalid control information. Then, note that $(k''_A, \mathsf{sync}''_A) = (k''_B, \mathsf{sync}''_B) = (1,1)$. Thus, by (4.9),

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A+1) - 0.9bC_5E_A + 2bC_2 - bC_4 + P_1\\ &\geq 0.8bC_5(k_A+1) - 0.9bC_5 \cdot 0.7(k_A-1) + 2bC_2 - bC_4\\ &\geq (2C_2 - C_4 + 1.77C_5)b\\ &\geq -C_{\mathsf{inv}}b.\end{aligned}$$

Next, suppose that $k_B = \mathsf{sync}_B = 1$ but Bob receives sound information. Then, note that $(k''_A, \mathsf{sync}''_A) \neq (k''_B, \mathsf{sync}''_B)$. Hence, by (4.9),

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A-1) - 0.9bC_5E_A - bC_6 + P_1\\ &\geq 0.8bC_5(k_A-1) - 0.9bC_5 \cdot 0.7(k_A-1) - bC_6\\ &\geq (0.17C_5 - C_6)b\\ &\geq -C_{\mathsf{inv}}b.\end{aligned}$$

Finally, suppose that $(k_B, \mathsf{sync}_B) \neq (1,1)$. Then, $\Delta k_B = \Delta E_B = 1$. Thus, if $k_A > 1$, then by (4.9),

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A - 1) - 0.9bC_5E_A + (-0.8bC_5 + 0.9bC_5) + P_1\\ &\geq 0.8bC_5(k_A - 1) - 0.9bC_5 \cdot 0.7(k_A - 1) + 0.1bC_5\\ &\geq 0.27C_5b\\ &\geq -C_{\mathsf{inv}}b,\end{aligned}$$

while if $k_A = 1$, Alice's transition must be an error transition and so,

$$\begin{aligned}\Delta\Phi &\geq -0.8bC_5 + 0.9bC_5\\ &= 0.1C_5b\\ &\geq -C_{\mathsf{inv}}b.\end{aligned}$$

c.) Suppose the control information sent by at least one of the parties is maliciously corrupted. If Alice's transition is a meeting point transition, then $\mathsf{mal}_{\mathrm{A}} \geq 0.2(k_A + 1) - 1$, and T_A can be rewound up to at most k_AB bits during the transition.
First, suppose that $(k''_B, \mathsf{sync}''_B) \neq (1,1)$. Then, $\Delta k_B \leq 1$ and $\Delta\mathsf{mal}_{\mathrm{B}} \leq 1$. Thus, by (4.9), we have

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5(k_A - 1) - 0.9bC_5E_A - 0.8bC_5 - C_7B - bC_6 + (P_1 - C_7B)\\ &\geq 0.8bC_5(k_A - 1) - 0.9bC_5 \cdot 0.7(k_A - 1) - 0.8bC_5 - C_7B - bC_6 - C_7B\\ &\geq -(0.8C_5 + C_6)b - 2C_7B\\ &\geq -C_{\mathsf{mal}}B.\end{aligned}$$

Next, suppose that $(k''_B, \mathsf{sync}''_B) = (1,1)$. Then, since Bob does not undergo a transition, we have $k_B = \mathsf{sync}_B = 1$. Also, the length of T_B can increase by at most B bits over the course of the next iteration. Hence,

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5k_{AB} - 0.9bC_5E_{AB} - C_1B + (P - C_7B) + 2bC_2 - bC_4\\ &\geq 0.8bC_5(k_A + 1) - 0.9bC_5 \cdot 0.7(k_A - 1) - C_1B - C_7B + 2bC_2 - bC_4\\ &\geq (2C_2 - C_4 + 1.6C_5)b - (C_1 + C_7)B\\ &\geq -C_{\mathsf{mal}}B.\end{aligned}$$

– Subcase 3: Both Alice and Bob undergo transitions before the start of the next iteration. Again, note that note that $E_A \leq 0.7(k_A - 1)$, due to (4.9). Similarly, $E_B \leq 0.7(k_B - 1)$. Also, we define P_1 as in (4.8) and define P_2 analogously:

$$P_2 = \begin{cases} 0.2C_7(k_B + 1)B - (1 + C_0H(\epsilon) + C_1)k_BB & \text{if Bob has an MP trans.}\\ 0 & \text{otherwise}\end{cases}.$$

Observe that $P_1, P_2 \geq 0$ for a suitable choice of constants C_0, C_1, C_7.

First, suppose that no party receives maliciously corrupted control information. Then, note that if Alice undergoes a meeting point transition, then $\mathsf{mal}_{\mathrm{A}} \geq 0.2(k_A + 1)$, and the transition can cause T_A to be rewound by at most $k_A B$ bits. Similarly, if Bob undergoes a meeting point transition, then $\mathsf{mal}_{\mathrm{B}} \geq 0.2(k_B + 1)$, and the transition can cause T_B to be rewound by at most $k_B B$ bits. Thus, regardless of the types of transitions that Alice and Bob make, we have

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5 k_{AB} - 0.9bC_5 E_{AB} + P_1 + P_2 + 2bC_2 - bC_4 \\ &\geq 0.8bC_5 k_{AB} - 0.9bC_5 \cdot 0.7((k_A - 1) + (k_B - 1)) + 2bC_2 - bC_4 \\ &\geq (2C_2 - C_4 + 1.6C_5)b \\ &\geq b,\end{aligned}$$

Now, suppose some party receives maliciously corrupted control information. We instead have $\mathsf{mal}_{\mathrm{A}} \geq 0.2(k_A + 1) - 1$ and $\mathsf{mal}_{\mathrm{B}} \geq 0.2(k_A + 1) - 1$. Thus,

$$\begin{aligned}\Delta\Phi &\geq 0.8bC_5 k_{AB} - 0.9bC_5 E_{AB} + (P_1 - C_7 B) + (P_2 - C_7 B) + 2bC_2 - bC_4 \\ &\geq (2C_2 - C_4 + 1.6C_5)b - 2C_7 B \\ &\geq -C_{\mathsf{mal}} B,\end{aligned}$$

as desired.

- Case 2: $k_A = k_B = 1$.
  - Subcase 1: $\mathsf{sync}_A = \mathsf{sync}_B = 1$. Then, note that if both parties receive sound control information, then $\mathsf{sync}''_A = \mathsf{sync}''_B = 0$. Thus,

    $$\Delta\Phi = -\Delta Z_1 = \frac{1}{2} bC_4 \geq b.$$

    On the other hand, if some party receives invalid control information but neither party receives maliciously corrupted control information, then note that either $\mathsf{sync}''_A = \mathsf{sync}''_B = 1$, in which case,

    $$\Delta\Phi = 0 \geq -C_{\mathsf{inv}} b,$$

    or $\mathsf{sync}''_A \neq \mathsf{sync}''_B$, in which case,

    $$\Delta\Phi \geq -2bC_2 + bC_4 - 1.6bC_5 - bC_6 \geq -C_{\mathsf{inv}} b.$$

Finally, consider the case in which some party receives maliciously corrupted information. Then, if $\mathsf{sync}''_A = \mathsf{sync}''_B$, note that $\Delta\ell^- \le 2$. Thus, if the subsequent state is unsynced, then

$$\Delta\Phi \ge -2C_1 B \ge -C_{\mathsf{mal}} B,$$

while if the subsequent state is perfectly or almost synced, then

$$\Delta\Phi \ge -2bC_2 + bC_4 - (2s+1)b \ge -C_{\mathsf{mal}} B.$$

Otherwise, if $\mathsf{sync}''_A \ne \mathsf{sync}''_B$, then $\Delta\ell^- \le 1$, and so,

$$\Delta\Phi \ge -C_1 B - 2bC_2 + bC_4 - 1.6bC_5 - bC_6 \ge -C_{\mathsf{mal}} B.$$

- <u>Subcase 2</u>: $\mathsf{sync}_A = \mathsf{sync}_B = 0$. First, suppose both parties receive sound control information. Then, either both parties do not undergo any transitions, in which case,

  $$\Delta\Phi \ge 2bC_2 + \frac{1}{2}bC_4 \ge b,$$

  or both parties undergo a meeting point transition, in which case the subsequent state is perfectly synced, and so,

  $$\Delta\Phi \ge -2bC_2 + \frac{1}{2}bC_4 \ge b.$$

  Next, consider the case in which some party receives invalid control information, but neither party receives maliciously corrupted control information. Suppose, without loss of generality, that Alice receives invalid control information. Then, $k''_A = \mathsf{sync}''_A = 1$. Note that if $k''_B = 2$, then

  $$\Delta\Phi \ge -2bC_2 + \frac{1}{2}bC_4 - 2.4bC_5 \ge -C_{\mathsf{inv}} b.$$

  Otherwise, if $k''_B = 1$, then either the subsequent state is perfectly synced, in which case

  $$\Delta\Phi \ge -2bC_2 + \frac{1}{2}bC_4 \ge -C_{\mathsf{inv}} b,$$

  or the subsequent state is almost synced, in which case

  $$\Delta\Phi \ge B(1 + C_0 H(\epsilon)) - 2bC_2 + \frac{1}{2}bC_4 - b \ge -C_{\mathsf{inv}} b,$$

or the subsequent state is unsynced, in which case

$$\Delta\Phi \geq -\frac{1}{2}bC_4 \geq -C_{\mathsf{inv}}b.$$

Finally, consider the case in which some party receives maliciously corrupted control information. If $k''_A = k''_B = 2$, then

$$\Delta\Phi \geq 2bC_2 - 4C_7B + \frac{1}{2}bC_4 \geq -C_{\mathsf{mal}}B.$$

On the other hand, if $k''_A = k''_B = 1$, then $\Delta\ell^- \leq 2$. Thus, if the subsequent state is unsynced, then

$$\Delta\Phi \geq -2(1 + C_0H(\epsilon) + C_1)B - \frac{1}{2}bC_4 \geq -C_{\mathsf{mal}}B,$$

while if the subsequent state is perfectly or almost synced, then

$$\Delta\Phi \geq -2(1 + C_0H(\epsilon) + C_1)B + \frac{1}{2}bC_4 - b \geq -C_{\mathsf{mal}}B.$$

If $k''_A \neq k''_B$, then without loss of generality, assume that $k''_A = 2$ and $k''_B = 1$. We then have

$$\Delta\Phi \geq -(1 + C_0H(\epsilon) + C_1)B - 2bC_2 + \frac{1}{2}bC_4 - 2.4bC_5 - C_7B \geq -C_{\mathsf{mal}}B.$$

- <u>Subcase 3</u>: $\mathsf{sync}_A \neq \mathsf{sync}_B$. Without loss of generality, assume that $\mathsf{sync}_A = 1$ and $\mathsf{sync}_B = 0$.

  First, suppose that neither party receives maliciously corrupted control information. Then, $k''_A = \mathsf{sync}''_A = k''_B = \mathsf{sync}''_B = 1$. Thus, if the subsequent state is unsynced, then we have

  $$\Delta\Phi \geq 1.6bC_5 + bC_6 + 2bC_2 - bC_4 \geq b,$$

  while if the subsequent state is perfectly or almost synced, then

  $$\Delta\Phi \geq 1.6bC_5 + bC_6 - b \geq b.$$

  Next, suppose that some party receives maliciously corrupted control information. Note that $k''_A = 1$. If $\mathsf{sync}_A = 1$ and $k''_B = 2$, then $\Delta\ell^- \leq 1$, and so,

  $$\Delta\Phi \geq -C_1B - 0.8bC_5 - C_7B + bC_6 \geq -C_{\mathsf{mal}}B.$$

If $\mathsf{sync}_A = 1$ and $k''_B = 1$, then either the subsequent state is unsynced, in which case,

$$\Delta\Phi \geq -C_1B-(1+C_0H(\epsilon)+C_1)B+0.8bC_5+bC_6+2bC_2-bC_4 \geq -C_{\mathsf{mal}}B,$$

or the subsequent state is perfectly/almost synced, in which case,

$$\Delta\Phi \geq -C_1B-(1+C_0H(\epsilon)+C_1)B+1.6bC_5+bC_6-(2s+1)b \geq -C_{\mathsf{mal}}B.$$

Finally, suppose $\mathsf{sync}_A = 0$. Then, note that

$$\Delta\Phi \geq -(1+C_0H(\epsilon)+C_1)B - 0.8bC_5 - C_7B \geq -C_{\mathsf{mal}}B.$$

- Case 3: The protocol is in an unsynced state, and $k_A = k_B > 1$.
  - Subcase 1: Suppose neither Alice nor Bob undergoes a transition before the start of the next iteration. Then, we have $\Delta k_A = \Delta k_B = 1$. If the control information received by both parties is either sound or invalid, then we have

    $$\Delta\Phi \geq 2bC_2 \geq b.$$

    On the other hand, if some party's control information is maliciously corrupted, then

    $$\Delta\Phi \geq 2bC_2 - 2bC_3 - 4BC_7 \geq -C_{\mathsf{mal}}B.$$
  - Subcase 2: Suppose both Alice and Bob undergo a transition, and suppose at least one of the transitions is a meeting point transition.

    a.) Suppose $\ell^{-\prime\prime} = 0$ and $k_A+1 = k_B+1 \leq \frac{4\ell^-}{B}$. Then, note that ℓ^+ decreases by at most $k_AB = k_BB$. Thus,

    $$\begin{aligned}\Delta\Phi &\geq -k_AB(1+C_0H(\epsilon)) + C_1\ell^- - 2C_2b(k_A-1) - C_4b \\ &\geq -k_AB(1+C_0H(\epsilon)) + C_1 \cdot \frac{B(k_A+1)}{4} - 2C_2b(k_A-1) - C_4b \\ &= k_AB\left(\frac{C_1}{4} - C_0H(\epsilon) - \frac{2C_2b}{B} - 1\right) + \frac{C_1B}{4} + (2C_2-C_4)b \\ &\geq b.\end{aligned}$$

    b.) Suppose $\ell^{-\prime\prime} \neq 0$. Without loss of generality, assume that Alice has made a meeting point transition. Note that if Alice has made an incorrect meeting point transition, then it is clear that $\mathsf{mal}'_{\mathrm{A}} \geq 0.2(k_A+1)$. On the other hand, if she has made a correct transition, then Bob has made an incorrect transition, since $\ell^{-\prime\prime} \neq 0$, and so, $\mathsf{mal}'_{\mathrm{B}} \geq 0.2(k_A + 1)$. Since $\mathsf{mal}'_{\mathrm{A}} = \mathsf{mal}'_{\mathrm{B}}$, it follows that $\mathsf{mal}'_{\mathrm{AB}} \geq 0.4(k_A + 1)$ in either case. Thus, if the control information in the current round is not maliciously corrupted, then $\mathsf{mal}_{\mathrm{AB}} \geq 0.4(k_A + 1)$, and so,

$$\begin{aligned}\Delta\Phi &\geq -k_A B(1 + C_0 H(\epsilon) + C_1) - 2C_2 b(k_A - 1) \\ &\quad + 2C_7 B \cdot 0.4(k_A + 1) - C_4 b \\ &\geq k_A B \left(0.8C_7 - C_0 H(\epsilon) - C_1 - \frac{2C_2 b}{B} - 1\right) \\ &\quad + (2C_2 - C_4)b + 0.8C_7 B \\ &\geq b.\end{aligned}$$

Otherwise, if some party's control information in the current round is corrupted, then $\mathsf{mal}_{\mathrm{AB}} \geq 0.4(k_A + 1) - 2$, and so,

$$\begin{aligned}\Delta\Phi &\geq k_A B \left(0.8C_7 - C_0 H(\epsilon) - C_1 - \frac{2C_2 b}{B} - 1\right) \\ &\quad + (2C_2 - C_4)b - 3.2C_7 B \\ &\geq -C_{\mathsf{mal}} B.\end{aligned}$$

c.) Suppose that $\ell^{-\prime\prime} = 0$ but $k_A + 1 = k_B + 1 > \frac{4\ell^-}{B}$. Then observe that there must have been at least

$$\frac{1}{4}(k_A + 1) - 0.2 \cdot \frac{1}{2}(k_A + 1) - 0.2 \cdot \frac{1}{2}(k_A + 1) = 0.05(k_A + 1) \quad (4.10)$$

maliciously corrupted rounds among the past k_A rounds. This is because there were $\frac{1}{4}(k_A + 1)$ iterations taking place as Alice's backtracking parameter increased from $\frac{1}{4}(k_A + 1)$ to $\frac{1}{2}(k_A + 1)$, of which at most $0.2 \cdot \frac{1}{2}(k_A + 1)$ iterations could have had invalid control information for Alice, and at most $0.2 \cdot \frac{1}{2}(k_A + 1)$ iterations could have had sound control information for Alice (since Alice did not undergo a meeting point transmission when her backtracking parameter reached $\frac{k_A + 1}{2}$). Thus, $\mathsf{mal}_{\mathrm{AB}} \geq 2 \cdot 0.05(k_A + 1) = 0.1(k_A + 1)$ and so,

$$\begin{aligned}\Delta\Phi &\geq -k_A B(1 + C_0 H(\epsilon)) - 2bC_2(k_A - 1) + C_7 B \cdot \mathsf{mal}_{\mathrm{AB}} - C_4 b \\ &\geq k_A B \left(0.1C_7 - C_0 H(\epsilon) - \frac{2C_2 b}{B} - 1\right) + (2C_2 - C_4)b + 0.1C_7 B \\ &\geq b.\end{aligned}$$

- Subcase 3: Suppose both Alice and Bob undergo error transitions. Then, $E'_A \geq 0.2(k_A + 1)$ and $E'_B \geq 0.2(k_B + 1) = 0.2(k_A + 1)$. Therefore, if both parties receive sound control information, then $E_A, E_B \geq 0.2(k_A + 1)$, and so,

  $$\begin{aligned}\Delta\Phi &\geq C_3 b E_{AB} - 2C_2 b(k_A - 1) - C_4 b \\ &\geq C_3 b(0.4k_A + 0.4) - 2C_2 b(k_A - 1) - C_4 b \\ &\geq (0.4C_3 - 2C_2)k_A b + (2C_2 + 0.4C_3 - C_4)b \\ &\geq (0.8C_3 - C_4)b \\ &\geq b.\end{aligned}$$

  On the other hand, if some party receives invalid or maliciously corrupted control information, then $E_A, E_B \geq 0.2(k_A + 1) - 1$, and so,

  $$\begin{aligned}\Delta\Phi &\geq C_3 b E_{AB} - 2C_2 b(k_A - 1) - C_4 b \\ &\geq (0.4C_3 - 2C_2)k_A b + (2C_2 - 1.6C_3 - C_4)b \\ &\geq (-1.2C_3 - C_4)b \\ &\geq -C_{\mathsf{inv}} b.\end{aligned}$$

- Subcase 4: Suppose only one of Alice and Bob undergoes a transition before the next iteration. Without loss of generality, assume Alice undergoes the transition.

  a.) Suppose the transition is an error transition. If both parties' control information is sound, then observe that $E_A \geq 0.2(k_A + 1)$. Thus,

  $$\begin{aligned}\Delta\Phi &\geq -2bC_2 k_A + bC_3 E_A - 0.8bC_5(k_A + 2) \\ &\geq -2bC_2 k_A + bC_3(0.2k_A + 0.2) - 0.8bC_5(k_A + 2) \\ &\geq k_A b(0.2C_3 - 0.8C_5 - 2C_2) + (0.2C_3 - 1.6C_5)b \\ &\geq b.\end{aligned}$$

  Otherwise, if some party's control information is invalid, but neither party's control information is maliciously corrupted, then $E_A \geq 0.2(k_A + 1) - 1 = 0.2k_A - 0.8$, and so,

  $$\begin{aligned}\Delta\Phi &\geq -2bC_2 k_A + bC_3 E_A - 0.8bC_5(k_A + 2) \\ &\geq -2bC_2 k_A + bC_3(0.2k_A - 0.8) - 0.8bC_5(k_A + 2) \\ &\geq k_A b(0.2C_3 - 0.8C_5 - 2C_2) - (0.8C_3 + 1.6C_5)b \\ &\geq -C_{\mathsf{inv}} b.\end{aligned}$$

Finally, if some party's control information is maliciously corrupted, then again, we have $E_A \geq 0.2k_A - 0.8$. Thus,

$$\begin{aligned}\Delta\Phi &\geq -2bC_2k_A + bC_3E_A - 0.8bC_5(k_A+2) - C_7B \\ &\geq k_Ab(0.2C_3 - 0.8C_5 - 2C_2) - (0.8C_3 + 1.6C_5)b - C_7B \\ &\geq -C_{\mathsf{mal}}B.\end{aligned}$$

b.) Suppose the transition is a meeting point transition. Then, since only one of the two players is transitioning, either (1.) Alice is incorrectly transitioning, meaning that $\mathsf{mal}'_{\mathrm{A}}, \mathsf{mal}'_{\mathrm{B}} \geq 0.2(k_A+1)$, or (2.) Bob should have also been transitioning, meaning that $\mathsf{mal}'_{\mathrm{A}}, \mathsf{mal}'_{\mathrm{B}} \geq \frac{1}{2}(k_A+1) - 0.2(k_A+1) - 0.2(k_A+1) \geq 0.1(k_A+1)$. Either way, $\mathsf{mal}'_{\mathrm{A}}, \mathsf{mal}'_{\mathrm{B}} \geq 0.1(k_A+1)$. Hence, if neither party's control information in the current round is maliciously corrupted, then $\mathsf{mal}_{\mathrm{A}}, \mathsf{mal}_{\mathrm{B}} \geq 0.1(k_A+1)$, and so,

$$\begin{aligned}\Delta\Phi &\geq -2bC_2k_A - 0.8bC_5(k_A+2) + 2C_7B\cdot\mathsf{mal}_{\mathrm{A}} + C_7B\cdot\mathsf{mal}_{\mathrm{B}} \\ &\quad - k_AB(1 + C_0H(\epsilon) + C_1) \\ &\geq -2bC_2k_A - 0.8bC_5(k_A+2) + 0.3C_7B(k_A+1) - k_AB(1 + C_0H(\epsilon) + C_1) \\ &\geq k_AB\left(0.3C_7 - C_1 - C_0H(\epsilon) - 2C_2\frac{b}{B} - 0.8C_5\frac{b}{B} - 1\right) - 1.6bC_5 + 0.3C_7B \\ &\geq b.\end{aligned}$$

Otherwise, if there is maliciously corrupted control information in the current round, then $\mathsf{mal}_{\mathrm{A}}, \mathsf{mal}_{\mathrm{B}} \geq 0.1(k_A+1) - 1 = 0.1k_A - 0.9$, and so,

$$\begin{aligned}\Delta\Phi &\geq -2bC_2k_A - 0.8bC_5(k_A+2) + 2C_7B\cdot\mathsf{mal}_{\mathrm{A}} + C_7B\cdot\mathsf{mal}_{\mathrm{B}} - C_7B \\ &\quad - k_AB(1 + C_0H(\epsilon) + C_1) \\ &\geq -2bC_2k_A - 0.8bC_5(k_A+2) + 3C_7B(0.1k_A - 0.9) - C_7B \\ &\quad - k_AB(1 + C_0H(\epsilon) + C_1) \\ &\geq k_AB\left(0.3C_7 - C_1 - C_0H(\epsilon) - 2C_2\frac{b}{B} - 0.8C_5\frac{b}{B} - 1\right) - 1.6bC_5 - 2.7C_7B \\ &\geq -C_{\mathsf{mal}}B,\end{aligned}$$

as desired.

□

Now, we are ready to prove the main theorem of the section, which implies Theorem 22 for the choice $\epsilon' = \epsilon^2$.

Theorem 33. *For any sufficiently small $\epsilon > 0$ and n-round interactive protocol Π with average message length $\ell = \Omega(1/\epsilon'^3)$, the protocol $\Pi_{\text{enc}}^{\text{oblivious}}$ given in Figure 4.3 successfully simulates Π, with probability $1-2^{-\Omega(\epsilon'^2 N_{\textsf{iter}})}$, over an oblivious adversarial channel with an ϵ error fraction while achieving a communication rate of $1-\Theta(\epsilon\log(1/\epsilon)) = 1-\Theta(H(\epsilon))$.*

Proof. Recall that Π_{blk} has n' rounds, where $n' = n(1+O(\epsilon'))$. Let $N_{\textsf{mal}}$ be the number of iterations of $\Pi_{\text{enc}}^{\text{oblivious}}$ in which some party's control information is maliciously corrupted. Moreover, let $N_{\textsf{inv}}$ be the number of iterations in which some party's control information is invalid but neither party's control information is maliciously corrupted. Finally, let $N_{\textsf{sound}}$ be the number of iterations starting at an unsynced or almost synced state such that both parties receive sound control information.

Now, by Lemma 20, we know that with probability $1-2^{-\Omega(\epsilon'^2 N_{\textsf{iter}})}$, $N_{\textsf{mal}} = O(\epsilon'^2 N_{\textsf{iter}})$. Also, by Lemma 19, $N_{\textsf{inv}} = O(\epsilon N_{\textsf{iter}})$ with probability $1-2^{-\Omega(\epsilon' N_{\textsf{iter}})}$. Recall that the total number of data bits that can be corrupted by the adversary throughout the protocol is at most $\epsilon b N_{\textsf{iter}}$. Since $N_{\textsf{iter}} = N_{\textsf{sound}} + N_{\textsf{inv}} + N_{\textsf{mal}}$, Lemmas 21, 22, and 23 imply that at the end of the execution of $\Pi_{\text{enc}}^{\text{oblivious}}$, the potential function Φ satisfies

$$\begin{aligned}
\Phi &\geq bN_{\textsf{sound}} - C\epsilon bN_{\textsf{iter}}\log(1/\epsilon) - C_{\textsf{inv}}bN_{\textsf{inv}} - C_{\textsf{mal}}BN_{\textsf{mal}} \\
&= b(N_{\textsf{iter}} - N_{\textsf{inv}} - N_{\textsf{mal}}) - C\epsilon bN_{\textsf{iter}}\log(1/\epsilon) - C_{\textsf{inv}}bN_{\textsf{inv}} - C_{\textsf{mal}}BN_{\textsf{mal}} \\
&= bN_{\textsf{iter}} - C\epsilon bN_{\textsf{iter}}\log(1/\epsilon) - (C_{\textsf{inv}}+1)bN_{\textsf{inv}} - (C_{\textsf{mal}}B+b)N_{\textsf{mal}} \\
&= bN_{\textsf{iter}} - C\epsilon bN_{\textsf{iter}}\log(1/\epsilon) - O(\epsilon)\cdot(C_{\textsf{inv}}+1)bN_{\textsf{iter}} - O(\epsilon'^2)\cdot(C_{\textsf{mal}}B+b)N_{\textsf{iter}} \\
&= bN_{\textsf{iter}}(1 - O(\epsilon)\cdot(C_{\textsf{inv}}+1) - O(\epsilon'^2)\cdot(C_{\textsf{mal}}s+1) - C\epsilon\log(1/\epsilon)) \\
&= bN_{\textsf{iter}}(1 - O(\epsilon\log(1/\epsilon))) \\
&= b\cdot\frac{n'}{b}(1+\Theta(\epsilon\log(1/\epsilon))) \\
&\geq n'(1+C_0H(\epsilon)) + (C_0+1)B.
\end{aligned}$$

Now, in order to complete the proof, it suffices to show that $\ell^+ \geq n'$. We consider several cases, based on the ending state:

- If the ending state is perfectly synced, then note that $jb - C\cdot\textsf{err}\cdot\log(1/\epsilon) \leq 2B$. Thus,
$$\ell^+ \geq \frac{\Phi - 2B}{1+C_0H(\epsilon)} \geq n'.$$
- If the ending state is almost synced, then note that
$$\ell^+ \geq \frac{\Phi}{1+C_0H(\epsilon)} - B \geq n'.$$

- If the ending state is unsynced and $(k_A, \mathsf{sync}_A) = (k_B, \mathsf{sync}_B)$, then first consider the case $k_A = k_B = 1$. In this case,

  $$\Phi \leq \ell^+(1 + C_0 H(\epsilon)) + 2bC_2,$$

  and so,

  $$\ell^+ \geq \frac{\Phi - 2bC_2}{1 + C_0 H(\epsilon)} \geq n'.$$

  Now, consider the case $k_A = k_B \geq 2$. Note that either $\ell^- \geq \frac{B}{4}(k_A + 1)$ or

  $$\mathsf{mal}_{\mathrm{AB}} \geq 2 \cdot \mathsf{mal}_{\mathrm{A}} \geq 2\left(\frac{1}{2}\widetilde{k}_A - 0.2\widetilde{k}_A - 0.2\widetilde{k}_A\right) \geq 0.2\widetilde{k}_A \geq 0.1(k_A + 1)$$

  (see (4.10)). If the former holds, then

  $$\begin{aligned} \Phi &\leq \ell^+(1 + C_0 H(\epsilon)) - C_1 \ell^- + bC_2 k_{AB} \\ &\leq \ell^+(1 + C_0 H(\epsilon)) - C_1 \cdot \frac{B}{4}(k_A + 1) + 2bC_2 k_A \\ &\leq \ell^+(1 + C_0 H(\epsilon)). \end{aligned}$$

  Otherwise, if the latter holds, then

  $$\begin{aligned} \Phi &\leq \ell^+(1 + C_0 H(\epsilon)) + bC_2 k_{AB} - 2C_7 B \mathsf{mal}_{\mathrm{AB}} \\ &\leq \ell^+(1 + C_0 H(\epsilon)) + 2bC_2 k_A - 2C_7 B(0.1(k_A + 1)) \\ &\leq \ell^+(1 + C_0 H(\epsilon)). \end{aligned}$$

  Either way,

  $$\ell^+ \geq \frac{\Phi}{1 + C_0 H(\epsilon)} \geq n'.$$

- If the ending state is unsynced and $k_A \neq k_B$, then consider the following. Note that if $k_A = 1$, then $E_A = 0 \leq 0.6k_A - 0.4$. On the other hand, if $k_A \geq 2$, then

  $$\begin{aligned} E_A &\leq 0.2\widetilde{k}_A + (k_A - \widetilde{k}_A) \\ &= k_A - 0.8\widetilde{k}_A \\ &\leq k_A - 0.8\left(\frac{k_A + 1}{2}\right) \\ &\leq 0.6k_A - 0.4. \end{aligned}$$

Either way, $E_A \leq 0.6k_A - 0.4$. Similarly, $E_B \leq 0.6k_B - 0.4$. Thus,

$$\begin{aligned}\Phi &\leq \ell^+(1 + C_0H(\epsilon)) + bC_5(-0.8k_{AB} + 0.9E_{AB}) \\ &\leq \ell^+(1 + C_0H(\epsilon)) + bC_5(-0.8k_{AB} + 0.9((0.6k_A - 0.4) + (0.6k_B - 0.4))) \\ &\leq \ell^+(1 + C_0H(\epsilon)).\end{aligned}$$

Thus,

$$\ell^+ \geq \frac{\Phi}{1 + C_0H(\epsilon)} \geq n'.$$

□

Finally, we prove Theorem 23.

Proof. Consider the same protocol $\Pi_{\text{enc}}^{\text{oblivious}}$ as in Theorem 33, except that we discard the random string exchange procedure at the beginning of the protocol. Since Alice and Bob have access to public shared randomness, they can instead initialize str to a common random string of the appropriate length and continue with the remainder of $\Pi_{\text{enc}}^{\text{oblivious}}$. Moreover, in this case, ϵ' is a parameter that is set as part of the input. Then, it is clear that the analysis of Theorem 33 still goes through. In this case, we have that the total number of rounds is

$$N_{\mathsf{iter}}\, b' = \frac{n'b'}{b}(1 + O(\epsilon \log(1/\epsilon))) = n(1 + O(H(\epsilon)) + O(\epsilon' \operatorname{polylog}(1/\epsilon'))),$$

while the success probability is $1 - 2^{-\Omega(\epsilon'^2 N_{\mathsf{iter}})} = 1 - 2^{-\Omega(\epsilon'^3 n)}$, as desired. □

Remark 34. *It is routine to verify that the constants $C_0, C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_{\mathsf{inv}}, C_{\mathsf{mal}}, C, D > 0$ can be chosen appropriately such that the relevant inequalities in Lemmas 21, 22, 23, and Theorem 33 all hold.*

Chapter 5

List Decodability

The results of this chapter were published in [CGV13].

5.1 Introduction

This work is motivated by the list decodability properties of random linear codes for correcting a large fraction of errors, approaching the information-theoretic maximum limit. We prove a near-optimal bound on the rate of such codes, by making a connection to and establishing improved bounds on the restricted isometry property of random submatrices of Hadamard matrices.

A q-ary error correcting code $\mathcal{C}$ of block length n is a subset of $[q]^n$, where $[q]$ denotes any alphabet of size q. The rate of such a code is defined to be $(\log_q |\mathcal{C}|)/n$. A good code $\mathcal{C}$ should be large (rate bounded away from 0) and have its elements (codewords) well “spread out.” The latter property is motivated by the task of recovering a codeword $c \in \mathcal{C}$ from a noisy version r of it that differs from c in a bounded number of coordinates. Since a random string $r \in [q]^n$ will differ from c on an expected $(1 - 1/q)n$ positions, the information-theoretically maximum fraction of errors one can correct is bounded by the limit $(1-1/q)$. In fact, when the fraction of errors exceeds $\frac{1}{2}(1-1/q)$, it is not possible to unambiguously identify the close-by codeword to the noisy string r (unless the code has very few codewords, i.e., a rate approaching zero).

In the model of list decoding, however, recovery from a fraction of errors approaching the limit $(1 - 1/q)$ becomes possible. Under list decoding, the goal is to recover a small list of all codewords of $\mathcal{C}$ differing from an input string r in at most ρn positions, where ρ

is the error fraction (our interest in this thesis being the case when ρ is close to $1 - 1/q$). This requires that $\mathcal{C}$ have the following sparsity property, called (ρ, L)-*list decodability*, for some small L : for every $r \in [q]^n$, there are at most L codewords within Hamming distance ρn from r. We will refer to the parameter L as the "list size" — it refers to the maximum number of codewords that the decoder may output when correcting a fraction ρ of errors. Note that (ρ, L)-list decodability is a strictly combinatorial notion, and does not promise an efficient algorithm to compute the list of close-by codewords. In this work, we only focus on this combinatorial aspect, and study a basic trade-off between between ρ, L, and the rate for the important class of random linear codes, when $\rho \to 1 - 1/q$. We describe the prior results in this direction and state our results next.

For integers $q, L \geq 2$, a random q-ary code of rate $R = 1 - h_q(\rho) - 1/L$ is (ρ, L)-list decodable with high probability. Here $h_q \colon [0, 1 - 1/q] \to [0, 1]$ is the q-ary entropy function: $h_q(x) = x \log_q(q-1) - x \log_q x - (1-x) \log_q(1-x)$. This follows by a straightforward application of the probabilistic method, based on a union bound over all centers $r \in [q]^n$ and all $(L+1)$-element subsets S of codewords that all codewords in S lie in the Hamming ball of radius ρn centered at r. For $\rho = 1 - 1/q - \epsilon$, where we think of q as fixed and $\epsilon \to 0$, this implies that a random code of rate $\Omega_q(\epsilon^2)$ is $(1 - 1/q - \epsilon, O_q(1/\epsilon^2))$-list decodable. (Here and below, the notation Ω_q and O_q hide constant factors that depend only on q.)

Understanding list decodable codes at the extremal radii $\rho = 1 - 1/q - \epsilon$, for small ϵ, is of particular significance mainly due to numerous applications that depend on this regime of parameters. For example, one can mention hardness amplification of Boolean functions [STV01], construction of hardcore predicates from one-way functions [GL89], construction of pseudorandom generators [STV01] and randomness extractors [Tre01], inapproximability of NP witnesses [KS99], and approximating the VC dimension [MU01]. Moreover, *linear* list decodable codes are further appealing due to their symmetries, succinct description, and efficient encoding. For some applications, linearity of list decodable codes is of crucial importance. For example, the black-box reduction from list decodable codes to capacity achieving codes for additive noise channels in [GS10], or certain applications of Trevisan's extractor [Tre01] (e.g., [Che10, § 3.6, § 5.2]) rely on linearity of the underlying list decodable code. Furthermore, list decoding of linear codes features an interplay between linear subspaces and Hamming balls and their intersection properties, which is of significant interest from a combinatorial perspective.

This work is focused on random *linear* codes, which are subspaces of $\mathbb{F}_q^n$, where $\mathbb{F}_q$ is the finite field with q elements. A random linear code $\mathcal{C}$ of rate R is sampled by picking $k = Rn$ random vectors in $\mathbb{F}_q^n$ and letting $\mathcal{C}$ be their $\mathbb{F}_q$-span. Since the codewords of $\mathcal{C}$ are now not all independent (in fact they are not even 3-wise independent), the above naive argument only proves the (ρ, L)-list decodability property for codes of rate $1 - h_q(\rho) - 1/\log_q(L+1)$ [ZP82].[1] For the setting $\rho = 1 - 1/q - \epsilon$, this implies a list size bound of $\exp(O_q(1/\epsilon^2))$ for random linear codes of rate $\Omega_q(\epsilon^2)$, which is exponentially worse than for random codes. Understanding if this exponential discrepancy between general and linear codes is inherent was raised an open question by [Eli91]. Despite much research, the exponential bound was the best known for random linear codes (except for the case of $q = 2$, and even for $q = 2$ only an existence result was known; see the related results section below for more details).

Our main result in this work closes this gap between random linear and random codes, up to polylogarithmic factors in the rate. We state a simplified version of the main theorem (Theorem 43) below.

Theorem 35 (Main, simplified)**.** *Let q be a prime power, and let $\epsilon > 0$ be a constant parameter. Then for some constant $a_q > 0$ only depending on q and all large enough integers n, a random linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of rate $a_q\epsilon^2/\log^3(1/\epsilon)$ is $(1-1/q-\epsilon, O(1/\epsilon^2))$-list decodable with probability at least 0.99. (One can take $a_q = \Omega(1/\log^4 q)$.)*

We remark that both the rate and list size are close to optimal for list decoding from a $(1-1/q-\epsilon)$ fraction of errors. For rate, this follows from the fact the q-ary "list decoding capacity" is given by $1 - h_q(\rho)$, which is $O_q(\epsilon^2)$ for $\rho = 1 - 1/q - \epsilon$. For list size, a lower bound of $\Omega_q(1/\epsilon^2)$ is known — this follows from [Bli86] for $q = 2$, and was shown for all q in [GV10, Bli08]. We have also assumed that the alphabet size q is fixed and have not attempted to obtain the best possible dependence of the constants on the alphabet size.

5.1.1 Related Results

We now discuss some other previously known results concerning list decodability of random linear codes.

First, it is well known that a random linear code of rate $\Omega_q(\epsilon^4)$ is $(1-1/q-\epsilon, O(1/\epsilon^2))$-list decodable with high probability. This follows by combining the Johnson bound for list decoding (see, for example, [GS01]) with the fact that such codes lie on the Gilbert-Varshamov bound and have relative distance $1-1/q-\epsilon^2$ with high probability. This result gets the correct quadratic dependence in list size, but the rate is worse.

[1] The crux of the argument is that any L non-zero vectors in $\mathbb{F}_q^k$ must have a subset of $\log_q(L+1)$ linearly independent vectors, and these are mapped independently by a random linear code. This allows one to effectively substitute $\log_q(L+1)$ in the place of L in the argument for fully random codes.

Second, for the case of $q = 2$, the existence of (ρ, L)-list decodable binary linear codes of rate $1 - h(\rho) - 1/L$ was proved in [GHSZ02]. For $\rho = 1/2 - \epsilon$, this implies the existence of binary linear codes of rate $\Omega(\epsilon^2)$ list decodable with list size $O(1/\epsilon^2)$ from an error fraction $1/2 - \epsilon$. This matches the bounds for random codes, and is optimal up to constant factors. However, there are two shortcomings with this result: (i) it only works for $q = 2$ (the proof makes use of this in a crucial way, and extensions of the proof to larger q have been elusive), and (ii) the proof is based on the semi-random method. It only shows the existence of such a code while failing to give any sizeable lower bound on the probability that a random linear code has the claimed list decodability property.

Motivated by this state of affairs, in [GHK11], the authors proved that a random q-ary linear code of rate $1 - h_q(\rho) - C_{\rho,q}/L$ is (ρ, L)-list decodable with high probability, for some $C_{\rho,q} < \infty$ that depends on ρ, q. This matches the result for completely random codes up to the leading constant $C_{\rho,q}$ in front of $1/L$. Unfortunately, for $\rho = 1 - 1/q - \epsilon$, the constant $C_{\rho,q}$ depends exponentially[2] on $1/\epsilon$. Thus, this result only implies an exponential list size in $1/\epsilon$, as opposed to the optimal $O(1/\epsilon^2)$ that we seek.

Summarizing, for random linear codes to achieve a polynomial in $1/\epsilon$ list size bound for error fraction $1 - 1/q - \epsilon$, the best lower bound on rate was $\Omega(\epsilon^4)$. We are able to show that random linear codes achieve a list size growing quadratically in $1/\epsilon$ for a rate of $\tilde{\Omega}(\epsilon^2)$. One downside of our result is that we do not get a probability bound of $1 - o(1)$, but only $1 - \gamma$ for any desired constant $\gamma > 0$ (essentially our rate bound degrades by a $\log(1/\gamma)$ factor).

Finally, there are also some results showing limitations on list decodability of random codes. It is known that both random codes and random linear codes of rate $1 - h_q(\rho) - \eta$ are, with high probability, *not* $(\rho, c_{\rho,q}/\eta)$-list decodable [Rud11, GN12]. For arbitrary (not necessarily random) codes, the best lower bound on list size is $\Omega(\log(1/\eta))$ [Bli86, GN12].

Remark 36. *We note that subsequent to our result, an improved version of our coding result was obtained in [Woo13], where it is shown that the rate of a random linear code can be improved to $\Omega(\epsilon^2/\log(q))$ while achieving $((1-1/q)(1-\epsilon), O(1/\epsilon^2))$-list decodability with probability $1 - o(1)$, thereby obtaining the optimal dependence of rate on ϵ. While [Woo13] does make use of the simplex encoding technique used here, it bypasses the use of RIP-2 and instead controls a related L_1 norm to achieve a simpler proof of the list decodability result. However, as a result, it does not improve the number of row samples of a DFT matrix needed to obtain RIP-2, a question that is interesting in its own right.*

[2] The constant $C_{\rho,q}$ depends exponentially on $1/\delta_\rho$, where $q^{-\delta_\rho n}$ is an upper bound on the probability that two random vectors in $\mathbb{F}_q^n$ of relative Hamming weight at most ρ, chosen independently and uniformly among all possibilities, sum up (over $\mathbb{F}_q^n$) to a vector of Hamming weight at most ρ. When $\rho = 1 - 1/q - \epsilon$, we have $\delta_\rho = \Theta_q(\epsilon^2)$ which makes the list size exponentially large.

5.1.2 Proof Technique

The proof of our result uses a different approach from the earlier works on list decodability of random linear codes [ZP82, Eli91, GHSZ02, GHK11]. Our approach consists of three steps.

Step 1: Our starting point is a relaxed version of the Johnson bound for list decoding that only requires the *average* pairwise distance of L codewords to be large (where L is the target list size), instead of the minimum distance of the code.

Technically, this extension is easy and pretty much follows by inspecting the proof of the Johnson bound. This has recently been observed for the binary case by [Che11]. Here, we give a proof of the relaxed Johnson bound for a more general setting of parameters, and apply it in a setting where the usual Johnson bound is insufficient. Furthermore, as a side application, we show how the average version can be used to bound the list decoding radius of codes which do not have too many codewords close to any codeword — such a bound was shown via a different proof in [GKZ08], where it was used to establish the list decodability of binary Reed-Muller codes up to their distance.

Step 2: Prove that the L-wise average distance property of random linear codes is implied by the order L restricted isometry property (RIP-2) of random submatrices of the Hadamard matrix (or in general, matrices related to the Discrete Fourier Transform).

This part is also easy technically, and our contribution lies in making this connection between restricted isometry and list decoding. The restricted isometry property has received much attention lately due to its relevance to compressed sensing (cf. [Can08, CRT06a, CRT06b, CT06, Don06]) and is also connected to the Johnson-Lindenstrauss dimension reduction lemma [BDDW08, AL13, KW11]. Our work shows another interesting application of this concept.

Step 3: Prove the needed restricted isometry property of the matrix obtained by sampling rows of the Hadamard matrix.

This is the most technical part of our proof. Let us focus on $q = 2$ for simplicity, and let H be the $N \times N$ Hadamard (Discrete Fourier Transform) matrix with $N = 2^n$, whose (x, y)'th entry is $(-1)^{\langle x,y \rangle}$ for $x, y \in \{0, 1\}^n$. We prove that (the scaled version of) a random submatrix of H formed by sampling a subset of $m = O(k \log^3 k \log N)$ rows of H satisfies RIP of order k with probability 0.99. This means that every k columns of this sampled matrix M are nearly orthogonal — formally, every $m \times k$ submatrix of M has all its k singular values close to 1.

For random matrices $m \times N$ with i.i.d Gaussian or (normalized) ± 1 entries, it is relatively easy to prove RIP-2 of order k when $m = O(k \log N)$ [BDDW08]. Proving such a bound for submatrices of the Discrete Fourier Transform (DFT) matrix (as conjectured in [RV08]) has been an open problem for many years. The difficulty is that the entries within a row are no longer independent, and not even triple-wise independent. The best proven upper bound on m for this case was $O(k \log^2 k(\log k + \log\log N)\log N)$, improving an earlier upper bound $O(k \log^6 N)$ of [CT06]. We improve the bound to $O(k \log^3 k \log N)$ — the key gain is that we do *not* have the $\log\log N$ factor. This is crucial for our list decoding connection, as the rate of the code associated with the matrix will be $(\log N)/m$, which would be $o(1)$ if $m = \Omega(\log N \log\log N)$. We will take $k = L = \Theta(1/\epsilon^2)$ (the target list size), and the rate of the random linear code will be $\Omega(1/(k \log^3 k))$, giving the bounds claimed in Theorem 35. We remark that any improvement of the RIP bound toward the best known lower bound of $m = \Omega(k \log N)$ [BLM15], a challenging open problem, would immediately translate into an improvement on the list decoding rate of random linear codes via our reductions.

Our RIP-2 proof for row-subsampled DFT matrices proceeds along the lines of [RV08], and is based on upper bounding the expectation of the supremum of a certain *Gaussian process* [LT91, Chap. 11]. The index set of the Gaussian process is $\mathcal{B}_2^{k,N}$, the set of all k-sparse unit vectors in $\mathbb{R}^N$, and the Gaussian random variable G_x associated with $x \in \mathcal{B}_2^{k,N}$ is a Gaussian linear combination of the squared projections of x on the rows sampled from the DFT matrix (in the binary case these are just squared Fourier coefficients)[3]. The key to analyzing the Gaussian process is an understanding of the associated (pseudo)-metric X on the index set, defined by $\|x - x'\|_X^2 = \mathbf{E}_G |G_x - G_{x'}|^2$. This metric is difficult to work with directly, so we upper bound distances under X in terms of distances under a different metric X'. The principal difference in our analysis compared to [RV08] is in the choice of X' — instead of the max norm used in [RV08], we use an L_p norm for large finite p applied to the sampled Fourier coefficients. We then estimate the covering numbers for X' and use Dudley's theorem to bound the supremum of the Gaussian process.

It is worth pointing out that, as we prove in this work, for low-rate random linear codes the average-distance quantity discussed in Step 1 above is substantially larger than the minimum distance of the code. This allows the relaxed version of the Johnson bound attain better bounds than what the standard (minimum-distance based) Johnson bound would obtain on list decodability of random linear codes. While explicit examples of linear codes surpassing the standard Johnson bound are already known in the literature

[3] We should remark that our setup of the Gaussian process is slightly different from [RV08], where the index set is k-element subsets of $[N]$, and the associated Gaussian random variable is the spectral norm of a random matrix. Moreover, in [RV08] the number of rows of the subsampled DFT matrix is a random variable concentrating around its expectation, contrary to our case where it is a fixed number. We believe that the former difference in our setup may make the proof accessible to a broader audience.

(see [GGR11] and the references therein), a by-product of our result is that in fact *most* linear codes (at least in the low-rate regime) surpass the standard Johnson bound. However, an interesting question is to see whether there are codes that are list decodable even beyond the relaxed version of the Johnson bound studied in this work.

Remark 37. *We note that in a subsequent work of Haviv and Regev [HR16], the authors further improve on the number of row samples needed in the subsampled DFT matrix to obtain RIP-2. They show that it suffices to take $O(k \log^2 k \log N)$ row samples, which improves on our result of $O(k \log^3 k \log N)$. Note that for the list decoding problem, this implies a rate of $\Omega(\epsilon^2 / \log^2(1/\epsilon))$, which again provides logarithmic improvements but falls short of $\Omega(\epsilon^2)$ rate provided by [Woo13].*

Furthermore, we remark that also subsequent to our work, Bourgain [Bou14] obtained a result showing that $O(k \log k \log^2 N)$ samples, which is incomparable to our result as well as the result of [RV08]. However, the bound of [HR16] strictly improves upon this result.

Organization of Chapter 5. The rest of Chapter 5 is organized as follows. After fixing some notation, in Section 5.2 we prove the average-case Johnson bound that relates a lower bound on average pair-wise distances of subsets of codewords in a code to list decoding guarantees on the code. We also show, in Section 5.2.3, an application of this bound on proving list decodability of "locally sparse" codes, which is of independent interest and simplifies some earlier list decoding results. In Section 5.3, we prove our main theorem on list decodability of random linear codes by demonstrating a reduction from RIP-2 guarantees of DFT-based complex matrices to average distance of random linear codes, combined with the Johnson bound. Finally, the RIP-2 bounds on matrices related to random linear codes are proved in Section 5.4.

Notation. Throughout this chapter, we will be interested in list decodability of q-ary codes. We will denote an alphabet of size q by $[q] := \{1, \dots, q\}$. For linear codes, the alphabet will be $\mathbb{F}_q$, the finite field with q elements (when q is a prime power). However, whenever there is a need to identify $\mathbb{F}_q$ with $[q]$ and vice versa (for example, to form the simplex encoding in Definition 32), we implicitly assume a fixed, but arbitrary, bijection between the two sets.

We use the notation $\mathbb{I} := \sqrt{-1}$. When $f \leq Cg$ (resp., $f \geq Cg$) for some absolute constant $C > 0$, we use the shorthand $f \lesssim g$ (resp., $f \gtrsim g$). We use the notation $\log(\cdot)$ when the base of logarithm is not of significance (e.g., $f \lesssim \log x$). Otherwise the base is subscripted as in $\log_b(x)$. The natural logarithm is denoted by $\ln(\cdot)$.

For a matrix M and a multiset of rows T, define M_T to be the matrix with $|T|$ rows, formed by the rows of M picked by T (in some arbitrary order). Each row in M_T may be repeated for the appropriate number of times specified by T.

5.2 Average-Distance Based Johnson Bound

In this section, we show how the average pair-wise distances between subsets of codewords in a q-ary code translate into list decodability guarantees on the code.

Recall that the relative Hamming distance between strings $x, y \in [q]^n$, denoted $\delta(x, y)$, is defined to be the fraction of positions i for which $x_i \neq y_i$. The relative distance of a code $\mathcal{C}$ is the minimum value of $\delta(x, y)$ over all pairs of codewords $x \neq y \in \mathcal{C}$. We define list decodability as follows.

Definition 30. *A code $\mathcal{C} \subseteq [q]^n$ is said to be (ρ, ℓ)-list decodable if $\forall y \in [q]^n$, the number of codewords of $\mathcal{C}$ within relative Hamming distance less than ρ is at most ℓ.*[4]

The following definition captures a crucial function that allows one to generically pass from distance property to list decodability.

Definition 31 (Johnson radius)**.** *For an integer $q \geq 2$, the Johnson radius function $J_q : [0, 1 - 1/q] \to [0, 1]$ is defined by*

$$J_q(x) := \frac{q-1}{q}\left(1 - \sqrt{1 - \frac{qx}{q-1}}\right) .$$

The well known Johnson bound in coding theory states that a q-ary code of relative distance δ is $(J_q(\delta - \delta/L), L)$-list decodable (see for instance [GS01]). Below we prove a version of this bound which does not need every pair of codewords to be far apart but instead works when the average distance of every set of codewords is large. The proof of this version of the Johnson bound is a simple modification of earlier proofs, but working with this version is a crucial step in our near-tight analysis of the list decodability of random linear codes.

Theorem 38 (Average-distance Johnson bound)**.** *Let $\mathcal{C} \subseteq [q]^n$ be a q-ary code and $L \geq 2$ an integer. If the average pairwise relative Hamming distance of every subset of L codewords of $\mathcal{C}$ is at least δ, then $\mathcal{C}$ is $(J_q(\delta - \delta/L), L - 1)$-list decodable.*

Thus, if one is interested in a bound for list decoding with list size L, it is enough to consider the average pairwise Hamming distance of subsets of L codewords.

[4] We require that the radius is strictly less than ρ instead of at most ρ for convenience.

5.2.1 Geometric Encoding of q-ary Symbols

We will give a geometric proof of the above result. For this purpose, we will map vectors in $[q]^n$ to complex vectors and argue about the inner products of the resulting vectors.

Definition 32 (Simplex encoding)**.** *The simplex encoding maps $x \in [q]$ to a vector $\varphi(x) \in \mathbb{C}^{q-1}$. The coordinate positions of this vector are indexed by the elements of $[q-1] := \{1, 2, \ldots, q-1\}$. Namely, for every $\alpha \in [q-1]$, we define $\varphi(x)(\alpha) := \omega^{x\alpha}$ where $\omega = e^{2\pi\mathbb{I}/q}$ is the primitive qth complex root of unity.*

For complex vectors $\vec{v} = (v_1, v_2, \ldots, v_m)$ and $\vec{w} = (w_1, w_2, \ldots, w_m)$, we define their inner product $\langle \vec{v}, \vec{w} \rangle = \sum_{i=1}^{m} v_i w_i^*$. From the definition of the simplex encoding, the following immediately follows:

$$\langle \varphi(x), \varphi(y) \rangle = \begin{cases} q-1 & \text{if } x = y, \\ -1 & \text{if } x \neq y. \end{cases} \tag{5.1}$$

We can extend the above encoding to map elements of $[q]^n$ into $\mathbb{C}^{n(q-1)}$ in the natural way by applying this encoding to each coordinate separately. From the above inner product formula, it follows that for $x, y \in [q]^n$ we have

$$\langle \varphi(x), \varphi(y) \rangle = (q-1)n - q\delta(x,y)n \; . \tag{5.2}$$

Similarly, we overload the notation to matrices with entries over $[q]$. Let M be a matrix in $[q]^{n \times N}$. Then, $\varphi(M)$ is an $n(q-1) \times N$ complex matrix obtained from M by replacing each entry with its simplex encoding, considered as a column complex vector.

Finally, we extend the encoding to *sets* of vectors (i.e., codes) as well. For a set $\mathcal{C} \subseteq [q]^n$, $\varphi(\mathcal{C})$ is defined as a $(q-1)n \times |\mathcal{C}|$ matrix with columns indexed by the elements of $\mathcal{C}$, where the column corresponding to each $c \in \mathcal{C}$ is set to be $\varphi(c)$.

5.2.2 Proof of Average-Distance Johnson Bound

We now prove the Johnson bound based on average distance.

Proof (of Theorem 38). Suppose $\{c_1, c_2, \ldots, c_L\} \subseteq [q]^n$ are such that their average pair-wise relative distance is at least δ, i.e.,

$$\sum_{1 \le i < j \le L} \delta(c_i, c_j) \ge \delta \cdot \binom{L}{2} \; . \tag{5.3}$$

We will prove that $c_1, c_2, \ldots, c_L$ cannot all lie in a Hamming ball of radius less than $J_q(\delta - \delta/L)n$. Since every subset of L codewords of $\mathcal{C}$ satisfy (5.3), this will prove that $\mathcal{C}$ is $(J_q(\delta - \delta/L), L-1)$-list decodable.

Suppose, for contradiction, that there exists $c_0 \in [q]^n$ such that $\delta(c_0, c_i) \leq \rho$ for $i = 1, 2, \ldots, L$ and some $\rho < J_q(\delta - \delta/L)$. Recalling the definition of $J_q(\cdot)$, note that the assumption about ρ implies

$$\left(1 - \frac{q\rho}{q-1}\right)^2 > 1 - \frac{q\delta}{q-1} + \frac{q}{q-1}\frac{\delta}{L} \, . \tag{5.4}$$

For $i = 1, 2, \ldots, L$, define the vector $v_i = \varphi(c_i) - \beta\varphi(c_0) \in \mathbb{C}^{n(q-1)}$, for some parameter β to be chosen later. By (5.2) and the assumptions about $c_0, c_1, \ldots, c_L$, we have $\langle \varphi(c_i), \varphi(c_0)\rangle \geq (q-1)n - q\rho n$, and $\sum_{1 \leq i < j \leq L} \langle \varphi(c_i), \varphi(c_j)\rangle \leq \binom{L}{2}\big((q-1)n - q\delta n\big)$. We have

$$\begin{aligned}
0 &\leq \left\langle \sum_{i=1}^{L} v_i, \ \sum_{i=1}^{L} v_i \right\rangle = \sum_{i=1}^{L} \langle v_i, v_i \rangle + 2 \cdot \sum_{1 \leq i < j \leq L} \langle v_i, v_j \rangle \\
&\leq L\big(n(q-1) + \beta^2 n(q-1) - 2\beta(n(q-1) - q\rho n)\big) + \\
&\quad + L(L-1)\big(n(q-1) - q\delta n + \beta^2 n(q-1) - 2\beta(n(q-1) - q\rho n)\big) \\
&= L^2 n(q-1)\left(\frac{q}{q-1}\frac{\delta}{L} + \left(1 - \frac{q\delta}{q-1} + \beta^2 - 2\beta\Big(1 - \frac{q\rho}{q-1}\Big)\right)\right)
\end{aligned}$$

Picking $\beta = 1 - \frac{q\rho}{q-1}$ and recalling (5.4), we see that the above expression is negative, a contradiction. □

5.2.3 An Application: List Decodability of Reed-Muller and Locally Sparse Codes

Our average-distance Johnson bound implies the following combinatorial result for the list decodability of codes that have few codewords in a certain vicinity of every codeword. The result allows one to translate a bound on the number of codewords in balls centered at codewords to a bound on the number of codewords in an arbitrary Hamming ball of smaller radius. An alternate proof of the below bound (using a "deletion" technique) was given by Gopalan, Klivans, and Zuckerman in [GKZ08], where they used it to argue the list decodability of (binary) Reed-Muller codes up to their relative distance. A mild strengthening of the deletion lemma was later used in [GGR11] to prove combinatorial bounds on the list decodability of tensor products and interleavings of binary linear codes.

Lemma 24. *Let $q \geq 2$ be an integer and $\eta \in (0, 1 - 1/q]$. Suppose $\mathcal{C}$ is a q-ary code such that for every $c \in \mathcal{C}$, there are at most A codewords of relative distance less than η from c (including c itself). Then, for every positive integer $L \geq 2$, $\mathcal{C}$ is $(J_q(\eta - \eta/L), AL - 1)$-list decodable.*

Note that setting $A = 1$ above gives the usual Johnson bound for a code of relative distance at least η.

Proof. We will lower bound the average pairwise relative distance of every subset of AL codewords of $\mathcal{C}$, and then apply Theorem 38.

Let $c_1, c_2, \ldots, c_{AL}$ be distinct codewords of $\mathcal{C}$. For $i = 1, 2, \ldots, AL$, the sum of relative distances of c_j, $j \neq i$, from c_i is at least $(AL - A)\eta$ since there are at most A codewords at relative distance less than η from c_i. Therefore

$$\frac{1}{\binom{AL}{2}} \cdot \sum_{1 \leq i < j \leq AL} \delta(c_i, c_j) \geq \frac{AL \cdot (AL - A)\eta}{AL(AL - 1)} = \frac{A(L-1)}{AL - 1}\eta \ .$$

Setting $\eta' = \frac{A(L-1)\eta}{AL-1}$, Theorem 38 implies that $\mathcal{C}$ is $(J_q(\eta' - \frac{\eta'}{AL}), AL - 1)$-list decodable. But $\eta' - \frac{\eta'}{AL} = \eta - \eta/L$, so the claim follows. □

5.3 Proof of the List Decoding Result

In this section, we prove our main result on list decodability of random linear codes. The main idea is to use the *restricted isometry property (RIP)* of complex matrices arising from random linear codes for bounding average pairwise distances of subsets of codewords. Combined with the average-distance based Johnson bound shown in the previous section, this proves the desired list decoding bounds. The RIP-2 condition that we use in this work is defined as follows.

Definition 33. *We say that a complex matrix $M \in \mathbb{C}^{m \times N}$ satisfies RIP-2 of order k with constant δ if, for any k-sparse vector $x \in \mathbb{C}^N$, we have*[5]

$$(1 - \delta)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1 + \delta)\|x\|_2^2.$$

Generally we think of δ as a small positive constant, say $\delta = 1/2$.

[5] We stress that in this work, we crucially use the fact that the definition of RIP that we use is based on the Euclidean (ℓ_2) norm.

Since we will be working with list decoding radii close to $1 - 1/q$, we derive a simplified expression for the Johnson bound in this regime; namely, the following:

Theorem 39. *Let $\mathcal{C} \subseteq [q]^n$ be a q-ary code and $L \geq 2$ an integer. If the average pairwise relative Hamming distance of every subset of L codewords of $\mathcal{C}$ is at least $(1-1/q)(1-\epsilon)$, then $\mathcal{C}$ is $((1 - 1/q)(1 - \sqrt{\epsilon + 1/L}), L - 1)$-list decodable.*

Proof. The proof is nothing but a simple manipulation of the bound given by Theorem 38. Let $\delta := (1 - 1/q)(1 - \epsilon)$. Theorem 38 implies that $\mathcal{C}$ is $(J_q(\delta(1 - 1/L)), L - 1)$-list decodable. Now,

$$\begin{aligned} J_q(\delta(1-1/L)) &= \frac{q-1}{q}\left(1 - \sqrt{1 - \frac{q}{q-1}\cdot\frac{q-1}{q}(1-\epsilon)\Big(1-\frac{1}{L}\Big)}\right) \\ &= \frac{q-1}{q}\left(1 - \sqrt{\epsilon + \frac{1}{L} - \frac{\epsilon}{L}}\right) \geq \frac{q-1}{q}\left(1 - \sqrt{\epsilon + \frac{1}{L}}\right). \end{aligned}$$ □

In order to prove lower bounds on average distance of random linear codes, we will use the simplex encoding of vectors (Definition 32), along with the following simple geometric lemma.

Lemma 25. *Let $c_1, \ldots, c_L \in [q]^n$ be q-ary vectors. Then, the average pairwise distance δ between these vectors satisfies*

$$\delta := \sum_{1\leq i<j\leq L} \delta(c_i, c_j)/\binom{L}{2} = \frac{L^2(q-1)n - \left\| \sum_{i\in[L]} \varphi(c_i)\right\|_2^2}{qL(L-1)n}.$$

Proof. The proof is a simple application of (5.2). The second norm on the right hand side can be expanded as

$$\begin{aligned} \Big\| \sum_{i\in[L]} \varphi(c_i)\Big\|_2^2 &= \sum_{i,j\in[L]} \langle \varphi(c_i), \varphi(c_j)\rangle \\ &\overset{(5.2)}{=} \sum_{i,j\in[L]} \Big((q-1)n - qn\delta(c_i,c_j)\Big) \\ &= L^2(q-1)n - 2qn \sum_{1\leq i<j\leq L} \delta(c_i, c_j) \\ &= L^2(q-1)n - 2qn\binom{L}{2}\delta, \end{aligned}$$

and the bound follows. □

Now we are ready to formulate our reduction from RIP-2 to average distance of codes.

Lemma 26. *Let $\mathcal{C} \subseteq [q]^n$ be a code and suppose $\varphi(\mathcal{C})/\sqrt{(q-1)n}$ satisfies RIP-2 of order L with constant $1/2$. Then, the average pairwise distance between every L codewords of $\mathcal{C}$ is at least $\left(1-\frac{1}{q}\right)\left(1-\frac{1}{2(L-1)}\right)$.*

Proof. Consider any set S of L codewords, and the real vector $x \in \mathbb{R}^{|\mathcal{C}|}$ with entries in $\{0,1\}$ that is exactly supported on the positions indexed by the codewords in S. Obviously, $\|x\|_2^2 = L$. Thus, by the definition of RIP-2 (Definition 33), we know that, defining $M := \varphi(\mathcal{C})$,

$$\|Mx\|_2^2 \leq 3L(q-1)n/2. \tag{5.5}$$

Observe that $Mx = \sum_{i\in[L]} \varphi(c_i)$. Let δ be the average pairwise distance between codewords in S. By Lemma 25 we conclude that

$$\begin{aligned}\delta &= \frac{L^2(q-1)n - \left\| \sum_{i\in[L]} \varphi(c_i) \right\|_2^2}{2q\binom{L}{2}n} \\ &\overset{(5.5)}{\geq} \frac{(L^2-1.5L)(q-1)n}{qL(L-1)n} \\ &= \frac{q-1}{q}\left(1-\frac{1}{2(L-1)}\right).\end{aligned}$$

□

We remark that, for our applications, the exact choice of the RIP constant in the above result is arbitrary, as long as it remains an absolute constant (although the particular choice of the RIP constant would also affect the constants in the resulting bound on average pairwise distance). Contrary to applications in compressed sensing, for our application it also makes sense to have RIP-2 with constants larger than one, since the proof only requires the upper bound in Definition 33.

By combining Lemma 26 with the simplified Johnson bound of Theorem 39, we obtain the following corollary.

Theorem 40. *Let $\mathcal{C} \subseteq [q]^n$ be a code and suppose $\varphi(\mathcal{C})/\sqrt{(q-1)n}$ satisfies RIP-2 of order L with constant $1/2$. Then $\mathcal{C}$ is $\left(\left(1-\frac{1}{q}\right)\left(1-\sqrt{\frac{1.5}{L-1}}\right), L-1\right)$-list decodable.*

Remark 41. *Theorem 40 is a direct corollary of Lemma 26 and Theorem 39, that in turn follow from mathematically simple proofs and establish more general connections between the notion of average distance of codes, list decodability, and RIP. However, it is possible to directly prove Theorem 40 without establishing such independently interesting connections. We present the direct proof below.*

Direct proof of Theorem 40. Let $\epsilon := \sqrt{\frac{1.5}{L-1}}$ and $M := \varphi(\mathcal{C})/\sqrt{(q-1)n}$. Let $S \subseteq \mathcal{C}$ be a set of L codewords, and suppose for the sake of contradiction that there is a vector $w \in [q]^n$ that is close in Hamming distance to all the L codewords in S. Namely, that for each $c \in S$ we have

$$\delta(w,c) < \big(1 - \frac{1}{q}\big)(1-\epsilon). \tag{5.6}$$

Let M' be the $(q-1)n \times L$ submatrix of M formed by removing all the columns of M corresponding to codewords of $\mathcal{C}$ outside the set S, and define $v := \varphi(w)/\sqrt{(q-1)n}$, considered as a row vector. RIP implies that for every non-zero vector $x \in \mathbb{R}^L$,

$$\frac{\|M'x\|_2^2}{\|x\|_2^2} \leq 3/2.$$

That is, if σ denotes the largest singular value of M', we have $\sigma^2 \leq 3/2$. Let $u := vM'$. From (5.6) combined with (5.2), we know that all the entries of u are greater than ϵ. Thus, $\|u\|_2^2 > \epsilon^2 L > 3/2$. On the other hand, $\|v\|_2 = 1$. This means that $\|vM'\|_2^2/\|v\|_2^2 > 3/2$, contradicting the bound on σ (maximum singular value of M'). □

Now, the matrix $\varphi(\mathcal{C})$ for a linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ has a special form. It is straightforward to observe that, when $q = 2$, the matrix is an incomplete Hadamard-Walsh transform matrix with $2^{\tilde{k}}$ columns, where $\tilde{k}$ is the dimension of the code. In general $\varphi(\mathcal{C})$ turns out to be related to a Discrete Fourier Transform matrix. Specifically, we have the following observation.

Observation 42. *Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be an $[n, \tilde{k}]$ linear code with a generator matrix $G \in \mathbb{F}_q^{\tilde{k} \times n}$, and define $N := q^{\tilde{k}}$. Consider the matrix of* linear forms $\mathsf{Lin} \in \mathbb{F}_q^{N \times N}$ *with rows and columns indexed by elements of $\mathbb{F}_q^{\tilde{k}}$ and entries defined by*

$$\mathsf{Lin}(x,y) := \langle x, y \rangle,$$

where $\langle \cdot, \cdot \rangle$ is the finite-field inner product over $\mathbb{F}_q^{\tilde{k}}$. Let $T \subseteq \mathbb{F}_q^{\tilde{k}}$ be the multiset of columns of G. Then, $\varphi(\mathcal{C}) = \varphi(\mathsf{Lin}_T)$ (Recall, from Definition 32, that the former simplex encoding $\varphi(\mathcal{C})$ is applied to the matrix enumerating the codewords of $\mathcal{C}$, while the latter, $\varphi(\mathsf{Lin}_T)$, is applied to the entries of a submatrix of Lin. *Also recall from the notations section that* Lin_T *denotes the submatrix of* Lin *obtained by choosing all the rows of* Lin *indexed by the elements of the multiset T, with possible repetitions).*

When G is uniformly random, $\mathcal{C}$ becomes a random linear code and $\varphi(\mathcal{C})$ can be sampled by the following process: Arrange n uniformly random rows of Lin, *sampled independently and with replacement, as rows of a matrix M. Then, replace each entry of*

M by its simplex encoding, seen as a column vector in $\mathbb{C}^{q-1}$. The resulting complex matrix is $\varphi(\mathcal{C})$.

The RIP-2 condition for random complex matrices arising from random linear codes is proved in Theorem 48 of Section 5.4. We now combine this theorem with the preceding results of this section to prove our main theorem on list decodability of random linear codes.

Theorem 43 (Main)**.** *Let q be a prime power, and let $\epsilon, \gamma > 0$ be constant parameters. Then for all large enough integers n, a random linear code $\mathcal{C} \subseteq \mathbb{F}_q^n$ of rate R, for some*

$$R \gtrsim \frac{\epsilon^2}{\log(1/\gamma)\log^3(q/\epsilon)\log q}$$

is $((1-1/q)(1-\epsilon), O(1/\epsilon^2))$-list decodable with probability at least $1-\gamma$.

Proof. Let $\mathcal{C} \subseteq \mathbb{F}_q^n$ be a uniformly random linear code associated to a random $Rn \times n$ generator matrix G over $\mathbb{F}_q$, for a rate parameter $R \leq 1$ to be determined later. Consider the random matrix $M = \varphi(\mathcal{C}) = \varphi(\mathsf{Lin}_T)$ from Observation 42, where $|T| = n$. Recall that M is a $(q-1)n \times N$ complex matrix, where $N = q^{Rn}$. Let $L := 1 + \lceil 1.5/\epsilon^2 \rceil = \Theta(1/\epsilon^2)$. By Theorem 48, for large enough N (thus, large enough n) and with probability $1-\gamma$, the matrix $M/\sqrt{(q-1)n}$ satisfies RIP-2 of order L with constant $1/2$, for some choice of $|T|$ bounded by

$$n = |T| \lesssim \log(1/\gamma) L \log(N) \log^3(qL). \tag{5.7}$$

Suppose n is large enough and satisfies (5.7) so that the RIP-2 condition holds. By Theorem 40, this ensures that the code $\mathcal{C}$ is $((1-1/q)(1-\epsilon), O(1/\epsilon^2))$-list decodable with probability at least $1-\gamma$.

It remains to verify the bound on the rate of $\mathcal{C}$. We observe that, whenever the RIP-2 condition is satisfied, G must have rank exactly Rn, since otherwise, there would be distinct vectors $x, x' \in \mathbb{F}_q^{Rn}$ such that $xG = x'G$. Thus in that case, the columns of M corresponding to x and x' become identical, implying that M cannot satisfy RIP-2 of any nontrivial order. Thus we can assume that the rate of $\mathcal{C}$ is indeed equal to R. Now we have

$$\begin{aligned} R &= \log_q |\mathcal{C}|/n = \log N/(n \log q) \\ &\overset{(5.7)}{\gtrsim} \frac{\log N}{\log(1/\gamma) L \log(N) \log^3(qL) \log q}. \end{aligned}$$

Substituting $L = \Theta(1/\epsilon^2)$ into the above expression yields the desired bound. □

5.4 Restricted Isometry Property of DFT-Based Matrices

In this section, we prove RIP-2 for random incomplete Discrete Fourier Transform matrices. However, we first prove some technical ingredients that we will later use in the proof.

The original definition of RIP-2 given in Definition 33 considers all complex vectors $x \in \mathbb{C}^n$. Below we show that it suffices to satisfy the property only for real-valued vectors x.

Proposition 44. *Let $M \in \mathbb{C}^{m\times N}$ be a complex matrix such that $M^\dagger M \in \mathbb{R}^{N\times N}$ and for any k-sparse vector $x \in \mathbb{R}^N$, we have*

$$(1-\delta)\|x\|_2^2 \leq \|Mx\|_2^2 \leq (1+\delta)\|x\|_2^2.$$

Then, M satisfies RIP-2 of order k with constant δ.

Proof. Let $x = a + \mathbb{I}b$, for some $a, b \in \mathbb{R}^N$, be any complex vector. We have $\|x\|_2^2 = \|a\|_2^2 + \|b\|_2^2$, and

$$\begin{aligned}
\left|\|Mx\|_2^2 - \|x\|_2^2\right| &= \left|x^\dagger M^\dagger M x - \|x\|_2^2\right| \\
&= \left|(a^\dagger - \mathbb{I}b^\dagger)M^\dagger M(a+\mathbb{I}b) - \|x\|_2^2\right| \\
&= \left|a^\dagger M^\dagger M a + b^\dagger M^\dagger M b + \mathbb{I}(a^\dagger M^\dagger M b - b^\dagger M^\dagger M a) - \|x\|_2^2\right| \\
&\overset{(\star)}{=} \left|a^\dagger M^\dagger M a + b^\dagger M^\dagger M b - \|x\|_2^2\right| \\
&= \left|a^\dagger M^\dagger M a - \|a\|_2^2 + b^\dagger M^\dagger M b - \|b\|_2^2\right| \\
&\overset{(\star\star)}{\leq} \delta\|a\|_2^2 + \delta\|b\|_2^2 \\
&= \delta\|x\|_2^2,
\end{aligned}$$

where $(\star)$ is due to the assumption that $M^\dagger M$ is real, which implies that $a^\dagger M^\dagger M b$ and $b^\dagger M^\dagger M a$ are conjugate real numbers (and thus, equal), and $(\star\star)$ is from the assumption that the RIP-2 condition is satisfied by M for real-valued vectors and the triangle inequality. □

As a technical tool, we use the standard symmetrization technique summarized in the following proposition for bounding deviation of summation of independent random variables from the expectation. The proof is a simple convexity argument (see, e.g., [LT91, Lemma 6.3] and [Ver12, Lemma 5.70]).

Proposition 45. *Let $(X_i)_{i\in[m]}$ be a finite sequence of independent random variables in a Banach space, and $(\epsilon_i)_{i\in[m]}$ and $(g_i)_{i\in[m]}$ be sequences of independent Rademacher (i.e., each uniformly random in $\{-1,+1\}$) and standard Gaussian random variables, respectively. Then,*

$$\mathbf{E}\Big\|\sum_{i\in[m]}(X_i-\mathbf{E}[X_i])\Big\| \lesssim \mathbf{E}\Big\|\sum_{i\in[m]}\epsilon_i X_i\Big\| \lesssim \mathbf{E}\Big\|\sum_{i\in[m]} g_i X_i\Big\|.$$

More generally, for a stochastic process $(X_i^{(\tau)})_{i\in[m],\tau\in\mathcal{T}}$ where $\mathcal{T}$ is an index set,

$$\mathbf{E}\sup_{\tau\in\mathcal{T}}\Big\|\sum_{i\in[m]}\big(X_i^{(\tau)}-\mathbf{E}[X_i^{(\tau)}]\big)\Big\| \lesssim \mathbf{E}\sup_{\tau\in\mathcal{T}}\Big\|\sum_{i\in[m]}\epsilon_i X_i^{(\tau)}\Big\| \lesssim \mathbf{E}\sup_{\tau\in\mathcal{T}}\Big\|\sum_{i\in[m]} g_i X_i^{(\tau)}\Big\|.$$

The following bound will be used in the proof of Claim 49, a part of the proof of Lemma 27.

Proposition 46. *Let $(\epsilon_i)_{i\in[m]}$ be a sequence of independent Rademacher random variables, and $(a_{ij})_{i,j\in[m]}$ be a sequence of complex coefficients with magnitude bounded by K. Then,*

$$\left|\mathbf{E}\Big(\sum_{i,j\in[m]} a_{ij}\epsilon_i\epsilon_j\Big)^s\right| \le (4Kms)^s.$$

Proof. By linearity of expectation, we can expand the moment as follows.

$$\mathbf{E}\Big(\sum_{i,j\in[m]} a_{ij}\epsilon_i\epsilon_j\Big)^s = \sum_{\substack{(i_1,\dots i_s)\in[m]^s\\(j_1,\dots j_s)\in[m]^s}}\Big(a_{i_1j_1}\cdots a_{i_sj_s}\mathbf{E}\Big[\epsilon_{i_1}\cdots\epsilon_{i_s}\epsilon_{j_1}\cdots\epsilon_{j_s}\Big]\Big).$$

Observe that $\mathbf{E}[\epsilon_{i_1}\cdots\epsilon_{i_s}\epsilon_{j_1}\cdots\epsilon_{j_s}]$ is equal to 1 whenever all integers in the sequence

$$(i_1,\dots,i_s,j_1,\dots,j_s)$$

appear an even number of times. Otherwise the expectation is zero. Denote by $S\subseteq[m]^{2s}$ the set of sequences $(i_1,\dots,i_s,j_1,\dots,j_s)$ that make the expectation non-zero. Then,

$$\left|\mathbf{E}\Big(\sum_{i,j\in[m]} a_{ij}\epsilon_i\epsilon_j\Big)^s\right| = \left|\sum_{(i_1,\dots i_s,j_1,\dots j_s)\in S} a_{i_1j_1}\cdots a_{i_sj_s}\right| \le K^s|S|.$$

One way to generate a sequence $\sigma\in S$ is as follows. Pick s coordinate positions of σ out of the $2s$ available positions, fill out each position by an integer in $[m]$, duplicate each integer at an available unpicked slot (in some fixed order), and finally permute the s positions of σ that were not originally picked. Obviously, this procedure can generate every sequence in S (although some sequences may be generated in many ways). The number of combinations that the combinatorial procedure can produce is bounded by $\binom{2s}{s} m^s(s!) \leq (4ms)^s$. Therefore, $|S| \leq (4ms)^s$ and the bound follows. □

We will use the following technical statement in the proof of Lemma 27.

Proposition 47. *Suppose for real numbers $a > 0$, $\mu \in [0,1]$, $\delta \in (0,1]$, we have*

$$a \cdot \Big(\frac{a}{1+a}\Big)^{\frac{1}{1+\mu}} \leq \frac{\delta^{\frac{2+\mu}{1+\mu}}}{4}.$$

Then, $a \leq \delta$.

Proof. Let $\delta' := \delta^{\frac{2+\mu}{1+\mu}}/4^{\frac{1}{1+\mu}} \geq \delta^{\frac{2+\mu}{1+\mu}}/4$. From the assumption, we have

$$a \cdot \Big(\frac{a}{1+a}\Big)^{\frac{1}{1+\mu}} \leq \delta' \Rightarrow a^{2+\mu} \leq \delta^{2+\mu}(1+a)/4. \tag{5.8}$$

Consider the function

$$f(a) := a^{2+\mu} - \delta^{2+\mu} a/4 - \delta^{2+\mu}/4.$$

The proof is complete if we show that, for every $a > 0$, the assumption $f(a) \leq 0$ implies $a \leq \delta$; or equivalently, $a > \delta \Rightarrow f(a) > 0$. Note that $f(0) < 0$, and $f''(a) > 0$ for all $a > 0$. The function f attains a negative value at zero and is convex at all points $a > 0$. Therefore, it suffices to show that $f(\delta) > 0$. Now,

$$f(\delta) = \delta^{2+\mu} - \delta^{3+\mu}/4 - \delta^{2+\mu}/4 \geq (3\delta^{2+\mu} - \delta^{3+\mu})/4.$$

Since $\delta \leq 1$, the last expression is positive, and the claim follows. □

Now, we are ready to prove the following theorem, which establishes the RIP-2 property for random incomplete DFT matrices.

Theorem 48. *Let T be a random multiset of rows of Lin, where $|T|$ is fixed and each element of T is chosen uniformly at random, and independently with replacement. Then, for every $\delta, \gamma > 0$, and assuming $N \geq N_0(\delta, \gamma)$, with probability at least $1-\gamma$ the matrix $\varphi(\mathsf{Lin}_T)/\sqrt{(q-1)|T|}$ (with $(q-1)|T|$ rows) satisfies RIP-2 of order k with constant δ for a choice of $|T|$ satisfying*

$$|T| \lesssim \frac{\log(1/\gamma)}{\delta^2} k \log(N) \log^3(qk). \tag{5.9}$$

The proof extends and closely follows the original proof in [RV08]. However we modify the proof at a crucial point to obtain a strict improvement over their original analysis which is necessary for our list decoding application. We present our improved analysis in this section.

Proof (of Theorem 48). Let $M := \varphi(\mathsf{Lin}_T)$. Each row of M is indexed by an element of T and some $\alpha \in \mathbb{F}_q^*$, where in the definition of simplex encoding (Definition 32), we identify $\mathbb{F}_q^*$ with $[q-1]$ in a fixed but arbitrary way. Recall that $T \subseteq \mathbb{F}_q^{\tilde{k}}$, where $N = q^{\tilde{k}}$. Denote the row corresponding to $t \in T$ and $\alpha \in \mathbb{F}_q^*$ by $M_{t,\alpha}$, and moreover, denote the set of k-sparse unit vectors in $\mathbb{C}^N$ by $\mathcal{B}_2^{k,N}$.

In order to show that $M/\sqrt{(q-1)|T|}$ satisfies RIP of order k, we need to verify that for any $x = (x_1, \dots, x_N) \in \mathcal{B}_2^{k,N}$,

$$|T|(q-1)(1-\delta) \leq \|Mx\|_2^2 \leq |T|(q-1)(1+\delta). \tag{5.10}$$

In light of Proposition 44, without loss of generality we can assume that x is real-valued (since the inner product between any pair of columns of M is real-valued).

For $i \in \mathbb{F}_q^n$, denote the ith column of M by M^i. For $x = (x_1, \dots, x_N) \in \mathcal{B}_2^{k,N}$, define the random variable

$$\begin{aligned} \Delta_x &:= \|Mx\|_2^2 - |T|(q-1) \qquad (5.11)\\ &= \sum_{\substack{(i,j) \in \mathsf{supp}(x) \\ i \neq j}} x_i x_j \langle M^i, M^j \rangle, \end{aligned}$$

where the second equality holds since each column of M has ℓ_2 norm $\sqrt{(q-1)|T|}$ and $\|x\|_2 = 1$. Thus, the RIP condition (5.10) is equivalent to

$$\Delta := \sup_{x \in \mathcal{B}_2^{k,N}} |\Delta_x| \leq \delta |T|(q-1). \tag{5.12}$$

Recall that Δ is a random variable depending on the randomness in T. The proof of the RIP condition involves two steps. First, bounding Δ in expectation, and second, a tail bound. The first step is proved, in detail, in the following lemma.

Lemma 27. *Let $\delta' > 0$ be a real parameter. Then, $\mathbf{E}[\Delta] \leq \delta' |T|(q-1)$ for a choice of $|T|$ bounded as follows:*

$$|T| \lesssim k \log(N) \log^3(qk)/\delta'^2.$$

Proof. We begin by observing that the columns of M are orthogonal in expectation; i.e., for any $i, j \in \mathbb{F}_q^n$, we have

$$\mathbf{E}_T\langle M^i, M^j\rangle = \begin{cases} |T|(q-1) & i = j, \\ 0 & i \neq j. \end{cases}$$

This follows from (5.2) and the fact that the expected relative Hamming distance between the columns of Lin corresponding to i and j, when $i \neq j$, is exactly $1 - 1/q$. It follows that for every $x \in \mathcal{B}_2^{k,N}$, $\mathbf{E}[\Delta_x] = 0$, namely, the stochastic process $\{\Delta_x\}_{x\in\mathcal{B}_2^{k,N}}$ is centered.

Recall that we wish to estimate

$$\begin{aligned} \mathcal{E} &:= \mathbf{E}_T\Delta \\ &= \mathbf{E}_T \sup_{x\in\mathcal{B}_2^{k,N}} \left| \sum_{t\in T}\sum_{\alpha\in\mathbb{F}_q^*} \langle M_{t,\alpha}, x\rangle^2 - |T|(q-1) \right|. \end{aligned} \tag{5.13}$$

Suppose the chosen multiset of the rows of Lin is written as a random sequence $T = (t_1, t_2, \ldots, t_{|T|})$. The random variables $\sum_{\alpha\in\mathbb{F}_q^*}\langle M_{t_i,\alpha}, x\rangle^2$, for different values of i, are independent. Therefore, we can use the standard symmetrization technique on summation of independent random variables in a stochastic process (Proposition 45) and conclude from (5.13) that

$$\mathcal{E} \lesssim \mathcal{E}_1 := \mathbf{E}_T\mathbf{E}_{\mathcal{G}} \sup_{x\in\mathcal{B}_2^{k,N}} \left(\sum_{t\in T} g_t \sum_{\alpha\in\mathbb{F}_q^*} \langle M_{t,\alpha}, x\rangle^2 \right), \tag{5.14}$$

where $\mathcal{G} := (g_t)_{t\in T}$ is a sequence of independent standard Gaussian random variables. Denote the term inside $\mathbf{E}_T$ in (5.14) by $\mathcal{E}_T$; namely,

$$\mathcal{E}_T := \mathbf{E}_{\mathcal{G}} \sup_{x\in\mathcal{B}_2^{k,N}} \left(\sum_{t\in T} g_t \sum_{\alpha\in\mathbb{F}_q^*} \langle M_{t,\alpha}, x\rangle^2 \right).$$

Now we observe that, for any fixed T, the quantity $\mathcal{E}_T$ defines the supremum of a Gaussian process. The Gaussian process $\{G_x\}_{x\in\mathcal{B}_2^{k,N}}$ induces a pseudo-metric $\|\cdot\|_X$ on

$\mathcal{B}_2^{k,N}$ (and more generally, $\mathbb{C}^N$), where for $x, x' \in \mathcal{B}_2^{k,N}$, the (squared) distance is given by

$$\begin{aligned}
\|x - x'\|_X^2 &:= \mathbf{E}_G |G_x - G_{x'}|^2 \\
&= \sum_{t \in T} \left(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x \rangle^2 - \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x' \rangle^2 \right)^2 \\
&= \sum_{t \in T} \left(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle \langle M_{t,\alpha}, x - x' \rangle \right)^2 . \qquad (5.15)
\end{aligned}$$

By Cauchy-Schwarz, (5.15) can be bounded as

$$\begin{aligned}
\|x - x'\|_X^2 &\leq \sum_{t \in T} \left(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2 \right) \left(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x - x' \rangle^2 \right) \qquad (5.16) \\
&\leq \sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2 \max_{t \in T} \left(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x - x' \rangle^2 \right) . \qquad (5.17)
\end{aligned}$$

Here is where our analysis differs from [RV08]. When $q = 2$, (5.17) is exactly how the Gaussian metric is bounded in [RV08]. We obtain our improvement by bounding the metric in a different way. Specifically, let $\eta \in (0, 1]$ be a positive real parameter to be determined later and let $r := 1 + \eta$ and $s := 1 + 1/\eta$ such that $1/r + 1/s = 1$. We assume that η is so that s becomes an integer. We use Hölder's inequality with parameters r and s along with (5.16) to bound the metric as follows:

$$\|x - x'\|_X \leq \left(\sum_{t \in T} \Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2 \Big)^r \right)^{1/2r} \left(\sum_{t \in T} \Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x - x' \rangle^2 \Big)^s \right)^{1/2s} . \qquad (5.18)$$

Since $\|x\|_2 = 1$, x is k-sparse, and $|M_{t,\alpha}| = 1$ for all choices of (t, α), Cauchy-Schwarz implies that $\langle M_{t,\alpha}, x \rangle^2 \leq k$ and thus, using the triangle inequality, we know that

$$\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2 \leq 4qk.$$

Therefore, for every $t \in T$, seeing that $r = 1 + \eta$, we have

$$\Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2\Big)^r \leq (4qk)^\eta \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2,$$

which, applied to the bound (5.18) on the metric, yields

$$\|x - x'\|_X \leq$$
$$(4qk)^{\eta/2r} \underbrace{\left(\sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x + x' \rangle^2\right)^{1/2r}}_{\mathcal{E}_2} \left(\sum_{t \in T} \Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x - x' \rangle^2\Big)^s\right)^{1/2s}. \quad (5.19)$$

Now,

$$\mathcal{E}_2 \;\leq\; 2\left(\sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x \rangle^2 + \sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x' \rangle^2\right) \leq 4\mathcal{E}'_T, \quad (5.20)$$

where we have defined

$$\mathcal{E}'_T := \sup_{x \in \mathcal{B}_2^{k,N}} \sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x \rangle^2. \quad (5.21)$$

Observe that, by the triangle inequality,

$$\mathcal{E}'_T \leq \sup_{x \in \mathcal{B}_2^{k,N}} \left| \sum_{t \in T} \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x \rangle^2 - |T|(q-1) \right| + |T|(q-1). \quad (5.22)$$

Plugging (5.21) back in (5.19), we so far have

$$\|x - x'\|_X \leq 4(4qk)^{\eta/2r} {\mathcal{E}'_T}^{1/2r} \left(\sum_{t \in T} \Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, x - x' \rangle^2\Big)^s\right)^{1/2s}. \quad (5.23)$$

For a real parameter $u > 0$, define $N_X(u)$ as the minimum number of spheres of radius u required to cover $\mathcal{B}_2^{k,N}$ with respect to the metric $\|\cdot\|_X$. We can now apply Dudley's theorem on supremum of Gaussian processes (cf. [LT91, Theorem 11.17]) and deduce that

$$\mathcal{E}_T \lesssim \int_0^\infty \sqrt{\log N_X(u)}\, du. \quad (5.24)$$

In order to make the metric $\|\cdot\|_X$ easier to work with, we define a related metric $\|\cdot\|_{X'}$ on $\mathcal{B}_2^{k,N}$, according to the right hand side of (5.23), as follows:

$$\|x-x'\|_{X'}^{2s} := \sum_{t\in T}\Big(\sum_{\alpha\in\mathbb{F}_q^*}\langle M_{t,\alpha}, x-x'\rangle^2\Big)^s. \tag{5.25}$$

Let K denote the diameter of $\mathcal{B}_2^{k,N}$ under the metric $\|\cdot\|_{X'}$. Trivially, $K \le 2|T|^{1/2s}\sqrt{qk}$. By (5.23), we know that

$$\|x-x'\|_X \le 4(4qk)^{\eta/2r}{\mathcal{E}'_T}^{1/2r}\|x-x'\|_{X'}. \tag{5.26}$$

Define $N_{X'}(u)$ similar to $N_X(u)$, but with respect to the new metric X'. The preceding upper bound (5.26) thus implies that

$$N_X(u) \le N_{X'}(u/(4(4qk)^{\eta/2r}{\mathcal{E}'_T}^{1/2r})). \tag{5.27}$$

Now, using this bound in (5.24) and after a change of variables, we have

$$\mathcal{E}_T \lesssim (4qk)^{\eta/2r}{\mathcal{E}'_T}^{1/2r}\int_0^\infty \sqrt{\log N_{X'}(u)}du. \tag{5.28}$$

Now we take an expectation over T. Note that (5.22) combined with (5.13) implies

$$\mathbf{E}_T\mathcal{E}'_T \le \mathcal{E} + |T|(q-1). \tag{5.29}$$

Using (5.24), we get

$$\begin{aligned}
\mathcal{E}^{2r} &\overset{(5.14)}{\lesssim} \mathcal{E}_1^{2r} = (\mathbf{E}_T\mathcal{E}_T)^{2r} \le \mathbf{E}_T\mathcal{E}_T^{2r} \\
&\lesssim (4qk)^\eta\mathbf{E}_T\left((\mathcal{E}'_T)^{1/2r}\int_0^\infty\sqrt{\log N_{X'}(u)}du\right)^{2r} \\
&\le (4qk)^\eta(\mathbf{E}_T\mathcal{E}'_T)\max_T\left(\int_0^\infty\sqrt{\log N_{X'}(u)}du\right)^{2r} \\
&\overset{(5.29)}{\le} (4qk)^\eta(\mathcal{E}+|T|(q-1))\max_T\left(\int_0^\infty\sqrt{\log N_{X'}(u)}du\right)^{2r}.
\end{aligned}$$

Define

$$\bar{\mathcal{E}} := \mathcal{E}\cdot\left(\frac{\mathcal{E}}{\mathcal{E}+|T|(q-1)}\right)^{1/(1+2\eta)}. \tag{5.30}$$

Therefore, recalling that $r = 1 + \eta$, the above inequality simplifies to

$$\bar{\mathcal{E}} \lesssim (4qk)^\eta \max_T \left(\int_0^K \sqrt{\log N_{X'}(u)} du \right)^{1+1/(1+2\eta)}, \tag{5.31}$$

where we have replaced the upper limit of integration by the diameter of $\mathcal{B}_2^{k,N}$ under the metric $\| \cdot \|_{X'}$ (obviously, $N_{X'}(u) = 1$ for all $u \geq K$).

Now we estimate $N_{X'}(u)$ in two ways. The first estimate is the simple volumetric estimate (cf. [RV08]) that gives

$$\log N_{X'}(u) \lesssim k \log(N/k) + k \log(1 + 2K/u). \tag{5.32}$$

This estimate is useful when u is small. For larger values of u, we use a different estimate as follows.

Claim 49. $\log N_{X'}(u) \lesssim |T|^{1/s} (\log N) qks/u^2$.

Proof. We use the method used in [RV08] (originally attributed to B. Maurey, cf. [Car85, § 1]) and empirically estimate any fixed real vector $x = (x_1, \ldots, x_N) \in \mathcal{B}_2^{k,N}$ by an m-sparse random vector Z, for sufficiently large m. The vector Z is an average

$$Z := \frac{\sqrt{k}}{m} \sum_{i=1}^m Z_i, \tag{5.33}$$

where each Z_i is a 1-sparse vector in $\mathbb{C}^N$ and $\mathbf{E}[Z_i] = x/\sqrt{k}$. The Z_i are independent and identically distributed.

The way each Z_i is sampled is as follows. Let $x' := x/\sqrt{k}$ so that $\|x'\|_1 = \frac{\|x\|_1}{\sqrt{k}} \leq 1$. With probability $1 - \|x'\|$, we set $Z_i := 0$. With the remaining probability, Z_i is sampled by picking a random $j \in \mathsf{supp}(x)$ according to the probabilities defined by absolute values of the entries of x', and setting $Z_i = \mathrm{sgn}(x'_j) e_j$, where e_j is the jth standard basis vector[6]. This ensures that $\mathbf{E}[Z_i] = x'$. Thus, by linearity of expectation, it is clear that $\mathbf{E}[Z] = x$. Now, consider

$$\mathcal{E}_3 := \mathbf{E} \|Z - x\|_{X'}.$$

If we pick m large enough to ensure that $\mathcal{E}_3 \leq u$, regardless of the initial choice of x, then we can conclude that for every x, there exists a Z of the form (5.33) that is at distance at most u from x (since there is always some fixing of the randomness that attains the expectation). In particular, the set of balls centered at all possible realizations of Z would

[6] Note that, since we have assumed x is a real vector, $\mathrm{sgn}(\cdot)$ is always well-defined.

cover $\mathcal{B}_2^{k,N}$. Since the number of possible choices of Z of the form (5.33) is at most $(2N+1)^m$, we have

$$\log N_{X'}(u) \lesssim m \log N. \tag{5.34}$$

In order to estimate the number of independent samples m, we use symmetrization again to estimate the deviation of Z from its expectation x. Namely, since the Z_i are independent, by the symmetrization technique stated in Proposition 45 we have

$$\mathcal{E}_3 \lesssim \frac{\sqrt{k}}{m} \cdot \mathbf{E} \left\| \sum_{i=1}^{m} \epsilon_i Z_i \right\|_{X'}, \tag{5.35}$$

where $(\epsilon_i)_{i \in [m]}$ is a sequence of independent Rademacher random variables in $\{-1, +1\}$. Now, consider

$$\begin{aligned} \mathcal{E}_4 &:= \mathbf{E} \left\| \sum_{i=1}^{m} \epsilon_i Z_i \right\|_{X'}^{2s} \\ &= \mathbf{E} \sum_{t \in T} \Big(\sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, \sum_{i=1}^{m} \epsilon_i Z_i \rangle^2 \Big)^s \\ &= \sum_{t \in T} \mathbf{E} \left(\sum_{\alpha \in \mathbb{F}_q^*} \Big(\sum_{i=1}^{m} \epsilon_i \langle M_{t,\alpha}, Z_i \rangle \Big)^2 \right)^s \\ &= \sum_{t \in T} \mathbf{E} \left(\sum_{i,j=1}^{m} \epsilon_i \epsilon_j \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, Z_i \rangle \langle M_{t,\alpha}, Z_j \rangle^* \right)^s . \end{aligned} \tag{5.36}$$

Since the entries of the matrix M are bounded in magnitude by 1, we have

$$\Big| \sum_{\alpha \in \mathbb{F}_q^*} \langle M_{t,\alpha}, Z_i \rangle \langle M_{t,\alpha}, Z_j \rangle^* \Big| \leq q.$$

Using this bound and Proposition 46, (5.36) can be simplified as

$$\mathcal{E}_4 = \mathbf{E} \left\| \sum_{i=1}^{m} \epsilon_i Z_i \right\|_{X'}^{2s} \leq |T| (4qms)^s,$$

and combined with (5.35), and using Jensen's inequality,

$$\mathcal{E}_3 \lesssim |T|^{1/2s} \sqrt{4qks/m}.$$

Therefore, we can ensure that $\mathcal{E}_3 \leq u$, as desired, for some large enough choice of m; specifically, for some $m \lesssim |T|^{1/s} qks/u^2$. Now from (5.34), we get

$$\log N_{X'}(u) \lesssim |T|^{1/s} (\log N) qks/u^2. \tag{5.37}$$

Claim 49 is now proved. □

Now we continue the proof of Lemma 27. Break the integration in (5.31) into two intervals. Consider

$$\mathcal{E}_5 := \underbrace{\int_0^A \sqrt{\log N_{X'}(u)} du}_{\mathcal{E}_6} + \underbrace{\int_A^K \sqrt{\log N_{X'}(u)} du}_{\mathcal{E}_7},$$

where $A := K/\sqrt{qk}$. We claim the following bound on $\mathcal{E}_5$.

Claim 50. $\mathcal{E}_5 \lesssim |T|^{1/2s} \sqrt{(\log N) qks} \log(qk)$.

Proof. First, we use (5.32) to bound $\mathcal{E}_6$ as follows.

$$\mathcal{E}_6 \lesssim A\sqrt{k \log(N/k)} + \sqrt{k} \int_0^A \sqrt{\ln(1 + 2K/u)} du. \tag{5.38}$$

Observe that $2K/u \geq 1$, so $1 + 2K/u \leq 4K/u$. Thus,

$$\begin{aligned} \int_0^A \sqrt{\ln(1 + 2K/u)}\, du &\leq \int_0^A \sqrt{\ln(4K/u)}\, du \\ &= 2K \int_0^{A/2K} \sqrt{\ln(2/u)}\, du \\ &= 2K \left(\frac{A}{2K} \sqrt{\ln(4K/A)} + \sqrt{\pi} \left(1 - \operatorname{erf}\left(\sqrt{\ln(4K/A)} \right) \right) \right) \\ &= A\sqrt{\ln(4K/A)} + 2\sqrt{\pi} K \operatorname{erfc}\left(\sqrt{\ln(4K/A)} \right), \end{aligned} \tag{5.39}$$

where $\operatorname{erf}(\cdot)$ is the Gaussian error function $\operatorname{erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$, and $\operatorname{erfc}(x) := 1 - \operatorname{erf}(x)$, and we have used the integral identity

$$\int \sqrt{\ln(1/x)} dx = -\frac{\sqrt{\pi}}{2} \operatorname{erf}\left(\sqrt{\ln(1/x)} \right) + x\sqrt{\ln(1/x)} + C$$

that can be verified by taking derivatives of both sides. Let us use the following upper bound

$$(\forall x > 0) \quad \text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \leq \frac{2}{\sqrt{\pi}} \int_x^\infty \frac{t}{x} e^{-t^2} dt = \frac{1}{\sqrt{\pi}} \cdot \frac{e^{-x^2}}{x},$$

and plug it into (5.39) to obtain

$$\begin{aligned} \int_0^A \sqrt{\ln(1+2K/u)}\, du &\leq A\sqrt{\ln(4K/A)} + 2\sqrt{\pi}K \left(\frac{1}{\sqrt{\pi}} \cdot \frac{A}{4K} \cdot \frac{1}{\sqrt{\ln(4K/A)}} \right) \\ &= A\sqrt{\ln(4K/A)} + \frac{A}{2\sqrt{\ln(4K/A)}} \\ &\lesssim A\sqrt{\log(qk)} \lesssim |T|^{1/2s}\sqrt{\log(qk)}), \end{aligned}$$

where the last inequality holds since $A = K/\sqrt{qk} \lesssim |T|^{1/2s}$. Therefore, by (5.38) we get

$$\mathcal{E}_6 \lesssim |T|^{1/2s}\sqrt{k}(\sqrt{\log N} + \sqrt{\log(qk)}). \tag{5.40}$$

On the other hand, we use Claim 49 to bound $\mathcal{E}_7$.

$$\begin{aligned} \mathcal{E}_7 &\lesssim \sqrt{|T|^{1/s}(\log N)qks} \int_A^K du/u \\ &\lesssim |T|^{1/2s}\sqrt{(\log N)qks}\log(qk). \end{aligned} \tag{5.41}$$

Combining (5.40) and (5.41), we conclude that for every fixed T,

$$\mathcal{E}_5 = \mathcal{E}_6 + \mathcal{E}_7 \lesssim |T|^{1/2s}\sqrt{(\log N)qks}\log(qk).$$

Claim 50 is now proved. □

By combining Claim 50 and (5.31), we have

$$\begin{aligned} \bar{\mathcal{E}} &\lesssim (4qk)^\eta \max_T \mathcal{E}_5^{1+1/(1+2\eta)} \\ &\lesssim (4qk)^\eta \Big(|T|^{1/2s}\sqrt{(\log N)qks}\log(qk) \Big)^{1+1/(1+2\eta)} \\ &= (4qk)^\eta |T|^{\eta/(1+2\eta)} \Big(\sqrt{(\log N)qks}\log(qk) \Big)^{1+1/(1+2\eta)}. \end{aligned} \tag{5.42}$$

By Proposition 47 (setting $a := \mathcal{E}/(|T|(q-1))$ and $\mu := 2\eta$), and recalling the definition (5.30) of $\bar{\mathcal{E}}$, in order to ensure that $\mathcal{E} \leq \delta'(q-1)|T|$, it suffices to have

$$\bar{\mathcal{E}} \leq \delta'^{\frac{2(1+\eta)}{1+2\eta}} |T|(q-1)/4. \tag{5.43}$$

Using (5.42), and after simple manipulations, (5.43) can be ensured for some

$$|T| \lesssim \frac{(4qk)^{2\eta}}{\eta} k(\log N)\log^2(qk)/\delta'^2.$$

This expression is minimized for some $\eta = 1/\Theta(\log(qk))$, which gives

$$|T| \lesssim k(\log N)\log^3(qk)/\delta'^2.$$

This concludes the proof of Lemma 27. □

Now we turn to the tail bound on the random variable Δ and estimate the appropriate size of T required to ensure that $\Pr[\Delta > \delta|T|(q-1)] \leq \gamma$. We observe that the tail bound proved in [RV08] uses the bound on $\mathbf{E}[\Delta]$ as a black box. In particular, the following lemma, for $q = 2$, is implicit in the proof of Theorem 3.9 in [RV08]. The extension to arbitrary alphabet size q and our slightly different sub-sampling process is straightforward. However, for completeness, we include a detailed proof.

Lemma 28. *[RV08, implicit] Suppose that, for some $\delta' > 0$, $\mathbf{E}[\Delta] \leq \delta'|T|(q-1)$. Then, there are absolute constants c_1, c_2, c_3 such that for every $\lambda \geq 1$,*

$$\Pr[\Delta > (c_1 + c_2\lambda)\delta'|T|(q-1)] \leq 6\exp(-\lambda^2),$$

provided that

$$|T|/k \geq c_3\lambda/\delta'. \tag{5.44}$$

Before we prove Lemma 28, we recall the following concentration theorem used by [RV08]:

Theorem 51 (Theorem 3.8 of [RV08])**.** *There is an absolute constant $C_{\mathrm{RV}} > 0$ such that the following holds. Let $Y_1, \ldots, Y_r$ be independent symmetric variables taking values in some Banach space. Assume $\|Y_j\| \leq R$ for all j, and let $Y := \|\sum_{i=1}^r Y_i\|$. Then, for any integers $l \geq Q$ and any $\tau > 0$, it holds that*

$$\Pr[Y \geq 8Q\mathbf{E}[Y] + 2Rl + \tau] \leq \Big(\frac{C_{\mathrm{RV}}}{Q}\Big)^l + 2\exp\Big(-\frac{\tau^2}{256Q\mathbf{E}[Y]^2}\Big).$$

From this theorem, we derive the following corollary.

Corollary 52. *There are absolute constants $C_1, C_2 > 0$ such that the following holds. Let $Y_1, \dots, Y_r$ be independent symmetric variables taking values in some Banach space. Assume $\|Y_j\| \leq R$ for all j, and let $Y := \| \sum_{i=1}^r Y_i \|$. Moreover, assume that $\mathbf{E}[Y] \leq E$ for some $E > 0$. Then, for every $\lambda \geq 1$, we have*

$$\Pr[Y \geq (C_1 + C_2 \lambda) E] \leq 3 \exp(-\lambda^2),$$

provided that $E \geq \lambda R$.

Proof. We properly set up the parameters of Theorem 51. Let $\tau := 16\sqrt{Q}\lambda E$. Suppose $R > 0$ (otherwise, the conclusion is trivial). Let $Q := \lceil e C_{\mathrm{RV}} \rceil$ so that

$$\left(\frac{C_{\mathrm{RV}}}{Q} \right)^l \leq \exp(-l). \tag{5.45}$$

Let $l := Q \lceil \tau/(2R) \rceil = Q \lceil 8\sqrt{Q}\lambda E/R \rceil \geq \lambda^2$, where the inequality is because of the assumption $E/R \geq \lambda$. The coefficient Q also ensures that $l \geq Q$. Note that

$$R \leq E/\lambda \leq E\lambda \leq \tau \Rightarrow 2Rl \leq 2RQ(\tau/(2R) + 1) = Q\tau + 2QR \leq 3Q\tau. \tag{5.46}$$

Thus,

$$\Pr[Y \geq 8QE + 2Rl + \tau] \leq \Pr[Y \geq 8Q\mathbf{E}[Y] + 2Rl + \tau] \leq 3 \exp(-\lambda^2),$$

where the second inequality follows from Theorem 51 and by observing the choice of τ, the bound (5.45), and the lower bounds on l. Finally,

$$8QE + 2Rl + \tau \overset{(5.46)}{\leq} 8QE + (3Q+1)\tau = 8QE + 16(3Q+1)\sqrt{Q}\lambda E =: (C_1 + C_2\lambda)E,$$

where $C_1 := 8Q$ and $C_2 := 16(3Q+1)\sqrt{Q}$. The result now follows since

$$\Pr[Y \geq (C_1 + C_2\lambda)E] \leq \Pr[Y \geq 8QE + 2Rl + \tau].$$

□

Now, we are ready for the proof of Lemma 28.

Proof of Lemma 28. We closely follow the proof of Theorem 3.9 in [RV08]. In order to prove the desired tail bound, we shall apply Corollary 52 on norm of an independent summation of matrices. Recall that $N = q^{\tilde{k}}$. Let the variable $t \in \mathbb{F}_q^{\tilde{k}}$ be chosen uniformly at random, and consider the random $(q-1) \times N$ matrix $A := \varphi(\mathsf{Lin}_{\{t\}})$ formed by picking the

t^{th} row of the $N \times N$ matrix Lin and replacing each entry by a column vector representing its simplex encoding. Let $\mathcal{A} := A^\top A - (q-1)I_N$, where I_N is the $N \times N$ identity matrix, and let $\|\mathcal{A}\|_\Upsilon$ denote the following norm

$$\|\mathcal{A}\|_\Upsilon := \sup_{x \in \mathcal{B}_2^{k,N}} \left| x^\top \mathcal{A} x \right|.$$

Denote the rows of A by $A_1, \ldots, A_{q-1}$, and observe that for every $x \in \mathcal{B}_2^{k,N}$ and $i \in \{1, \ldots, q-1\}$,

$$|\langle A_i, x\rangle| \leq \|A_i\|_\infty \|x\|_1 \leq \sqrt{k}, \tag{5.47}$$

where the second inequality follows from Cauchy-Schwarz. Therefore, since

$$\mathcal{A} = \sum_{i=1}^{q-1} (A_i^\top A_i - I_N),$$

for every $x \in \mathcal{B}_2^{k,N}$, we have

$$x^\top \mathcal{A} x = \sum_{i=1}^{q-1} \langle A_i, x\rangle^2 - (q-1) \overset{(5.47)}{\leq} (q-1)(k-1),$$

and thus,

$$\|\mathcal{A}\|_\Upsilon \leq qk. \tag{5.48}$$

Suppose the original random row of Lin is written as a vector over $\mathbb{F}_q^N$ with coordinates indexed by the elements of $\mathbb{F}_q^{\tilde{k}}$. That is, $\mathsf{Lin}_{\{t\}} =: (w(u))_{u \in \mathbb{F}_q^{\tilde{k}}} =: w$. In particular, $w(u) = \langle u, t\rangle$, where the inner product is over $\mathbb{F}_q$. Let $u, v \in \mathbb{F}_q^{\tilde{k}}$. By basic linear algebra,

$$\Pr_t[w(u) = w(v)] = \Pr[\langle (u-v), t\rangle = 0] = \begin{cases} 1/q & \text{if } u \neq v, \\ 1 & \text{if } u = v. \end{cases}$$

Note that the (u, v)th entry of the matrix $A^\top A$ can be written as

$$(A^\top A)(u, v) = \langle \varphi(w(u)), \varphi(w(v))\rangle \overset{(5.1)}{=} \begin{cases} -1 & \text{if } w(u) \neq w(v), \\ q-1 & \text{if } w(u) = w(v). \end{cases}$$

Therefore, from this we can deduce that $\mathbf{E}[A^\top A] = (q-1)I_N$, or in other words, all entries of $\mathcal{A}$ are centered random variables; i.e., $\mathbf{E}[\mathcal{A}] = 0$.

Let $X_1, \ldots, X_{|T|}$ be independent random matrices, each distributed identically to $\mathcal{A}$, and consider the independent matrix summation

$$X := X_1 + \cdots + X_{|T|}.$$

Since each summand is a centered random variable, X is centered as well. Recall the random variables Δ_x and Δ from (5.11) and (5.12), and observe that Δ_x can be written as

$$\Delta_x = x^\top X x,$$

which in turn implies

$$\Delta = \|X\|_\Upsilon.$$

Thus, the assumption of the lemma implies that

$$\mathbf{E}[\|X\|_\Upsilon] \le \delta' |T| (q-1),$$

and proving a tail bound on Δ is equivalent to proving a tail bound on the norm of X. This can be done using Corollary 52. However, the result cannot be directly applied to X since the X_i are centered but not symmetric for $q > 2$. As in [RV08], we use standard symmetrization techniques to overcome this issue. Namely, let $\mathcal{B}$ be the symmetrized version of $\mathcal{A}$ defined as

$$\mathcal{B} := \mathcal{A} - \mathcal{A}',$$

where $\mathcal{A}'$ is an independent matrix identically distributed to $\mathcal{A}$. Similar to X, define

$$Y := Y_1 + \cdots + Y_{|T|},$$

where the Y_i are independent and distributed identically to $\mathcal{B}$. As in the proof of Theorem 3.9 of [RV08], a simple application of Fubini and triangle inequalities implies that

$$\begin{aligned} \mathbf{E}[X] &\le \mathbf{E}[Y] \le 2\mathbf{E}[X], \\ \Pr[X > 2\mathbf{E}[X] + \tau] &\le 2\Pr[Y > \tau]. \end{aligned} \tag{5.49}$$

Let $E := 2\delta'|T|(q-1)$ so that by the above inequalities we know that $\mathbf{E}[Y] \le E$. We can now apply Corollary 52 to Y and deduce that, for some absolute constants $C_1, C_2 > 0$, and every $\lambda \ge 1$,

$$\Pr[Y \ge (C_1 + C_2\lambda)E] \le \exp(-\lambda^2), \tag{5.50}$$

provided that $E \ge \lambda R$, where we can take $R = qk$ by (5.48). Plugging in the choice of E, we get the requirement that

$$\frac{|T|}{k} \ge \frac{\lambda q}{2\delta'(q-1)},$$

which can be ensured by an appropriate choice of c_3 in (5.44). Finally, (5.49) and (5.50) can be combined to deduce that

$$\begin{aligned} \Pr[X > 2E + (C_1 + C_2\lambda)E] &\le \Pr[X > 2\mathbf{E}[X] + (C_1 + C_2\lambda)E] \\ &\le 2\Pr[Y > (C_1 + C_2\lambda)E] \\ &\le 6\exp(-\lambda^2). \end{aligned}$$

This completes the proof of Lemma 28. □

□

Now it suffices to instantiate the above lemma with $\lambda := \sqrt{\ln(6/\gamma)}$ and $\delta' := \delta/(c_1 + c_2\lambda) = \delta/\Theta(\sqrt{\ln(6/\gamma)})$, and use the resulting value of δ' in Lemma 27. Since Lemma 27 ensures that $|T|/k = \Omega(\log N)$, we can take N large enough (depending on δ, γ) so that (5.44) is satisfied. This completes the proof of Theorem 48. □

The proof of Theorem 48 does not use any property of the DFT-based matrix other than orthogonality and boundedness of the entries. However, for syntactical reasons, that is, the way the matrix is defined for $q > 2$, we have presented the theorem and its proof for the special case of the DFT-based matrices. The proof goes through with no technical changes for any orthogonal matrix with bounded entries (as is the case for the original proof of [RV08]). In particular, we remark that the following variation of Theorem 48 also holds:

Theorem 53. *Let $A \in \mathbb{C}^{N\times N}$ be any orthonormal matrix with entries bounded by $O(1/\sqrt{N})$. Let T be a random multiset of rows of A, where $|T|$ is fixed and each element of T is chosen uniformly at random, and independently with replacement. Then, for every $\delta, \gamma > 0$, and assuming $N \geq N_0(\delta, \gamma)$, with probability at least $1 - \gamma$ the matrix $(\sqrt{N/|T|})A_T$ satisfies RIP-2 of order k with constant δ for a choice of $|T|$ satisfying*

$$|T| \lesssim \frac{\log(1/\gamma)}{\delta^2} k(\log N)\log^3 k. \qquad \square$$

We also note that the sub-sampling procedure required by Theorem 48 is slightly different from the one originally used by [RV08]. In our setting, we appropriately fix the target number of row (i.e., $|T|$) first, and then draw as many uniform and independent samples of the rows of the original Fourier matrix as needed (with replacement). On the other hand, [RV08] samples the RIP matrix by starting from the original $N \times N$ Fourier matrix, and then removing each row independently with a certain probability. This probability is carefully chosen so that the expected number of remaining rows in the end of the process is sufficiently large. Our modified sampling is well suited for our coding-theoretic applications, and offers the additional advantage of always returning a matrix with the exact desired number of rows. However, we point out that since Theorem 48 is based on the original ideas of [RV08], it can be verified to hold with respect to either of the two sub-sampling procedures.

Chapter 6

Affine Invariance and Local Testability

The results of this chapter were published in [GSVW15].

6.1 Motivation

Another property of interest in the context of error-correcting codes is *local testability*. Locally testable codes (LTCs) have received much attention in recent years. They are error-correcting codes equipped with a *tester*, a randomized algorithm that queries the received word at a few judiciously chosen positions and decides whether the word is a valid codeword. The tester must accept valid codewords with probability 1 and reject words that are far from the code in Hamming distance with nontrivial probability. LTCs have garnered much interest due to their connections to probabilistically checkable proofs (PCPs) and property testing (see the surveys [Gol11, Tre04]). Many PCP constructions are based on or related to LTCs [BSGH$^+$06, GS06, Din07, BSS08]. The primary focus thus far has been on LTCs in which the number of queries is *constant*, and much progress has been made on constructions in this regime (see for example the line of work culminating in [Vid13]). There has also been work on LTCs with a sub-linear number of queries (i.e., N^ϵ queries where N is the block length and $\epsilon > 0$ is arbitrary) [BSS06, GKS13].

Recently, high-rate LTCs in which the tester is allowed to make a *linear* number of queries (i.e., ϵN queries) have been shown to have surprising connections to central questions in the theory of approximation algorithms. Specifically, in [BGH$^+$12] a beautiful connection between such LTCs and the construction of small set expander graphs is presented. Instantiating this connection with the binary Reed-Muller (RM) code, the authors of [BGH$^+$12] construct small set expanders whose Laplacian has many small eigenvalues.

They also derandomize the "long code" (hypercube) which underlies all optimal PCP constructions to give a shorter low-degree version (which they called the "short code"). The low-degree long code has since been used to construct more size-efficient PCPs, leading to improved hardness results for hypergraph coloring [DG13, GHH$^+$14, KS14].

The binary Reed-Muller code $\mathrm{RM}(r, n)$ of degree r in n variables encodes a (multilinear) polynomial $f \in \mathbb{F}_2[X_1, \ldots, X_n]$ of total degree at most r by the vector of its evaluations $\big(f(\alpha)\big)_{\alpha \in \mathbb{F}_2^n}$. The (minimum) distance of $\mathrm{RM}(r, n)$ equals 2^{n-r}. A central ingredient in the above exciting recent developments is a local testability result for binary RM codes due to [BKS$^+$10]. In the high-rate regime of relevance to the above connections, the result of [BKS$^+$10] shows the following (one should think of s as constant, and n as growing in the statement below):

Theorem 54 ([BKS$^+$10])**.** *There exists an absolute constant $\xi > 0$ such that the Reed-Muller code $\mathrm{RM}(n-s, n)$ (of distance 2^s) can be tested with 2^{n-s+1} queries, rejecting a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that is $2^s/3$-far from $\mathrm{RM}(n-s, n)$ with probability at least ξ.*

The n-variate binary RM code of constant distance d has dimension $\approx N-(\log N)^{\log d - 1}$, where $N = 2^n$, and is testable with $2N/d$ queries. For the connection to small set expansion in [BGH$^+$12], a binary linear code $\mathcal{C}$ of block length N that is testable with ϵN queries results in a graph with vertex set $\mathcal{C}^\perp$ (the dual code to $\mathcal{C}$) whose Laplacian has $\Omega(N)$ eigenvalues smaller than $O(\epsilon)$. To get many "bad" eigenvalues as a function of the graph size, we would like $\mathcal{C}^\perp$ to be small compared to N, i.e., we would like the dimension of $\mathcal{C}$ to be as large as possible. This leads to the following question, which motivates our work:

Question 55. *What is the largest dimension of a distance d binary linear code $\mathcal{C} \subset \mathbb{F}_2^N$ that is testable with $O(N/d)$ queries?*

Reed-Muller codes give a construction with dimension $\approx N - (\log N)^{\log d - 1}$. Achieving higher dimension would imply small set expanders (SSEs) whose Laplacians have even larger number of small eigenvalues, and in particular, a dimension of $N - O_d(\log N)$ would imply polynomially many small eigenvalues (the existence of such SSEs is necessary if the SSE intractability hypothesis of [RS10] holds). The only known upper bound on dimension is the Hamming bound $\approx N - \frac{d}{2} \log N$, based just on the distance (*without* using the testability condition). BCH codes achieve (up to lower order terms) the Hamming bound; however, as all codewords in the dual of the BCH code have Hamming weight close to $N/2$, the BCH code is not testable with $O(N/d)$ queries.[1]

[1] It is known that for linear codes, one can assume without loss of generality that the tester checks orthogonality to a set of dual codewords (see [BSHR05]).

In other words, there is a gap between the dimension of the testable distance d Reed-Muller code, which is $\approx N-(\log N)^{\log d-1}$, and the dimension of the BCH code of distance d, which is $\approx N - \frac{d}{2}\log N$ (and best possible for distance d). The natural question motivating the work in this chapter is to understand how significant a limitation the testability requirement poses on the dimension of the code, and whether the highest possible dimension of a *testable* code with distance d is closer to that of BCH or RM. Unfortunately, this seems to be a difficult problem in general.

As a first step toward the above challenging goal, in this chapter, we focus on proving limitations in the special case of *affine-invariant codes*. Affine invariance generalizes many popular families of algebraic codes and is a well-studied concept in coding theory. The investigation of the role of affine invariance, and invariance in general, in the context of testability were initiated by [KS07a] and there have been many further works in the area (see, for instance, the survey by Sudan [Sud11, Section 5] and references therein). Affine-invariant codes are subsets of functions from $\mathbb{F}_Q^n$ to $\mathbb{F}_q$ that are invariant under affine transformations of the domain, where $\mathbb{F}_Q$ and $\mathbb{F}_q$ are finite fields with $\mathbb{F}_Q$ extending $\mathbb{F}_q$ (see Section 6.2.1 for a more formal definition in the case of $Q = q$).

As it turns out, both Reed-Muller and BCH are affine-invariant codes. Furthermore, [GKS13] as well as [HRZS13] show constructions of additional classes of codes that are testable with $O(N/d)$ queries and provide slight improvements to the dimension of the Reed-Muller code. Interestingly, these improved codes, too, are affine-invariant. It seems worthwhile, therefore, to initially restrict our attention to affine-invariant codes and gain further insights into the problem. The rich structure of affine invariance gives us some handle for understanding the constraints imposed by local testability. For example, although we know virtually no lower bounds for LTCs in the constant-query regime, it was shown in [BSS11] that affine-invariant LTCs for a constant number of queries cannot have constant rate.

Affine invariance also offers many advantages for *constructing* locally testable codes. It turns out that their structure means that only fairly weak conditions have to be satisfied in order for a code to be testable. For example, it has been shown that *any* affine-invariant linear code which is characterized by constant-weight constraints is testable with constantly many queries [KS07a].[2]

In the constant distance (linear query) regime, affine-invariant codes have yielded LTCs with the highest dimension known thus far, and improving slightly upon binary Reed-Muller codes. By using a technique known as *lifting* of affine-invariant codes, the

[2] In fact, the testability also extends to non-linear codes [BFH$^+$13], but with an enormous price in the error analysis.

works [GKS13, HRZS13] give constructions of a class of affine-invariant linear-query LTCs that improve upon the dimension of the RM code. For some of these codes, with lower dimension, [HRZS13] shows the soundness guarantee that is necessary to allow them to replace the RM code in the application of [BGH$^+$12]. Without this stronger guarantee, [GKS13] gives a code $\mathcal{C} \subseteq \{0,1\}^N$ (where $N = 2^n$) of distance d and dimension

$$\dim(\mathcal{C}) \geq N - \left(1 + \frac{\log N}{\log d - 1}\right)^{\log d - 1}, \tag{6.1}$$

which is testable with $2N/d$ queries. This code contains the binary code $\mathrm{RM}(n - \log d, n)$, as do the corresponding codes of [HRZS13]. Hence, it is natural to ask for the optimal dimension of a code containing the RM code that still has the desired testability properties. Note that the (extended) BCH code of distance d (which does not satisfy the testability requirements) also contains $\mathrm{RM}(n - \log d, n)$.

In this chapter, we show that the code of [GKS13] is essentially optimal. That is, we show for constant d that any linear affine-invariant code $\mathcal{C} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ of distance d which is testable with $2N/d$ queries (the number of queries needed for testing the RM code of the same distance) and contains $\mathrm{RM}(n - \log d, n)$ has dimension at most

$$\dim(\mathcal{C}) \leq N - \left(\frac{\log N}{\log^2 d}\right)^{\log d - 1},$$

where $N = 2^n$ (see Theorem 43 for the formal statement of the result). We also show that any linear affine-invariant code $\mathcal{C}$ satisfying (6.1) must contain the RM code of degree $\log N - (\log d - 1)\log(n + \log d - 1) + \Omega_d(1)$, implying that our assumptions are not far from the truth.

Our results suggest that any LTC which improves noticeably on the Reed-Muller code in the linear query regime would need techniques beyond the known ones based on affine invariance.

Organization. In Section 6.2, we give definitions and preliminaries on affine-invariant LTCs. Section 6.3 then describes previous work that complements our results. In Section 6.4, we prove our lower bound on affine-invariant codes that contain high-order Reed-Muller codes. Finally, Section 6.5 provides some justification for why containment of a high-order Reed-Muller code is a reasonable assumption. Omitted proofs appear in the appendices.

6.2 Preliminaries

6.2.1 Our Setup

We begin with some basic terminology for locally testable codes. In the following, $\mathbb{F}$ is a finite field. Recall Definition 12 for the dual code of a linear code, which we restate for convenience below:

Definition 12 (Dual code)**.** *Given an* $[n, k]_q$ *linear code* $\mathcal{C}$*, we define its* dual code $\mathcal{C}^\perp$ *to be the code given by*

$$\mathcal{C}^\perp = \{\mathbf{c}' \in \mathbb{F}_q^n : \mathbf{c}^T \cdot \mathbf{c}' = 0 \textit{ for all } \mathbf{c} \in \mathcal{C}\}.$$

Definition 34 (δ-far)**.** *A word* $w \in \mathbb{F}^N$ *is said to be* δ-far *from a linear code* $\mathcal{C} \subseteq \mathbb{F}^N$ *if* $\min_{c \in \mathcal{C}} \Delta(w, c) \geq \delta N$*, where* $\Delta(x, y)$ *denotes the Hamming distance between two vectors.*

We now define the notions of (weak) locally testable codes (LTCs) and canonical testers.

Definition 35 (Canonical testers)**.** *Suppose* $\mathcal{C} \subseteq \mathbb{F}^N$ *is a linear code. A* k-query canonical tester *for* $\mathcal{C}$ *is a distribution* $\mathcal{D}$ *over subsets* $I \subseteq \{1, 2, \ldots, N\}$ *satisfying* $|I| \leq k$*; invoking the tester on a word* $w \in \mathbb{F}^N$ *consists of sampling* $I \sim \mathcal{D}$ *and accepting* w *if and only if* $w|_I \in \mathcal{C}|_I$*.*

Definition 36 (LTCs)**.** *A linear code* $\mathcal{C} \subseteq \mathbb{F}^N$ *is said to be a* (k, ϵ, ρ)-LTC *if there exists a* k*-query canonical tester that always accepts elements of* $\mathcal{C}$ *and rejects all* $w \notin \mathcal{C}$ *that are* ρ*-far from* $\mathcal{C}$ *with probability at least* ϵ*. The parameter* ϵ *is known as the* soundness *of the tester.*

Our definition for LTCs and testers is motivated by a result of Ben-Sasson, et al. [BSHR05], which shows that any general tester for an LTC can be reduced to the above canonical form. Together with the linearity, it follows that a necessary condition for a linear code to be testable is the existence of a dual codeword of low Hamming weight.

Fact 56 (Existence of a low weight dual codeword)**.** *Let* $\mathcal{C} \subseteq \mathbb{F}^N$ *be a linear LTC of distance at least 2 that is testable with* k *queries. Then, for any* $1 \leq j \leq N$*, there must exist a nonzero* $w \in \mathcal{C}^\perp$ *such that* $w_j \neq 0$ *and* $|\{i \in \{1, \ldots, N\} : w_i \neq 0\}| \leq k$*, i.e.,* w *has Hamming weight at most* k*.*

Proof. By Theorem 3.3 in [BSHR05], we know that if $\mathcal{C}$ is a (k, ϵ, ρ)-LTC, then $\mathcal{C}$ has a k-query *canonical* tester $\mathcal{T}$ that accepts all $v \in \mathcal{C}$ with probability 1 and rejects any v that is ρ-far from $\mathcal{C}$ with probability at least ϵ. Consider an arbitrary v that is ρ-far from $\mathcal{C}$. There exists some $I \subseteq \{1, 2, \ldots, N\}$ in the support of the underlying distribution of $\mathcal{T}$ such that $v|_I \notin \mathcal{C}|_I$. Thus, $\mathcal{C}|_I$ is a linear subspace of $\mathbb{F}^N|_I$ that is strictly contained in $\mathbb{F}^N|_I$. It follows that there exists a nonzero $w' \in \mathbb{F}^N|_I$ that is orthogonal to all of $\mathcal{C}|_I$. Hence, the word $w \in \mathbb{F}^N$ that is supported on I and satisfies $w|_I = w'$ is an element of $\mathcal{C}^\perp$ with Hamming weight at most k. □

In this work, we will write $N = 2^n$. All logarithms will be base 2 unless otherwise specified.

We next define affine-invariant codes, which are the focus of this work.

Definition 37. *Let $\mathbb{F}_Q$ be a field of size Q. We call a function $A : \mathbb{F}_Q^t \to \mathbb{F}_Q^t$ an* affine transformation *if $A(x) = Mx + b$ for some matrix $M \in \mathbb{F}_Q^{t \times t}$ and vector $b \in \mathbb{F}_Q^t$.*

Definition 38. *Let $\mathbb{F}_q$ be a field of size q, and let $\mathbb{F}_Q$ be its extension field of size $Q = q^m$. Then, we call a code $\mathcal{F} \subseteq \{\mathbb{F}_Q^t \to \mathbb{F}_q\}$* affine-invariant *if for every $f \in \mathcal{F}$ and affine transformation $A : \mathbb{F}_Q^t \to \mathbb{F}_Q^t$, the function $f \circ A$ is in $\mathcal{F}$.*

Throughout, we will make use of the following useful fact about affine-invariant codes.

Fact 57. *If $\mathcal{C} \subseteq \mathbb{F}^N$ is a linear affine-invariant code of dimension D, then its dual code $\mathcal{C}^\perp \subseteq \mathbb{F}^N$ is also a linear affine-invariant code, of dimension $N - D$.*

The task is to consider binary affine-invariant codes $\mathcal{C} \subseteq \{f : \mathbb{F}_2^n \to \mathbb{F}_2\}$ with fixed distance d such that $\mathcal{C}$ is an LTC with locality $O\left(\frac{N}{d}\right)$. We wish to find the optimal rate of such a code $\mathcal{C}$.

6.2.2 Affine-Invariant Codes

From now on, we will only consider *univariate* affine-invariant codes (i.e., subsets of $\{f \colon \mathbb{F}_{2^n} \to \mathbb{F}_2\}$), since $(\mathbb{F}_{2^s})^t$ is isomorphic to $\mathbb{F}_{2^{st}}$ for all t, and passing from a multivariate code to the corresponding univariate code preserves affine invariance and testability ([BSS11]). More precisely, there exists an isomorphism $\phi : \mathbb{F}_{2^{st}} \to (\mathbb{F}_{2^s})^t$ such that for any *multivariate* linear affine-invariant LTC $\mathcal{C} \subseteq \{f \colon (\mathbb{F}_{2^s})^t \to \mathbb{F}_2\}$, the corresponding univariate code $\{g \circ \phi \,|\, g \in \mathcal{C}\} \subseteq \{f : \mathbb{F}_{2^{st}} \to \mathbb{F}_2\}$ is also a linear affine-invariant LTC with the same testability parameters and dimension.[3]

Here, we present some basic facts that will allow us to study affine-invariant codes by analyzing their degree sets (see, for example, [KS07b]). The definitions are stated in their full generality for fields of size q, although we will primarily be concerned with the case $q = 2$.

Definition 39. *For a function $f : \mathbb{F}_{q^n} \to \mathbb{F}_q$, write it as the unique polynomial $f(x) = \sum_{e=0}^{q^n-1} c_e x^e$ of degree at most $q^n - 1$ which agrees with f on $\mathbb{F}_{q^n}$. Then, the* support *of f, denoted Supp(f), is the set of degrees with non-zero coefficients in f, that is, Supp$(f) = \{e : c_e \neq 0\}$.*

Definition 40. *Let $\mathcal{F} \subseteq \{\mathbb{F}_{q^n} \to \mathbb{F}_q\}$ be a code. We define $\mathrm{Deg}(\mathcal{F})$, the* degree set *of $\mathcal{F}$, to be the set $\mathrm{Deg}(\mathcal{F}) = \bigcup_{f \in \mathcal{F}} Supp(f)$.*

Definition 41. *Suppose $D \subseteq \{0, 1, \ldots, q^n - 1\}$. We define $\mathcal{T}(D) \subseteq \{\mathbb{F}_{q^n} \to \mathbb{F}_q\}$ to be the* trace code *on D defined by*

$$\mathcal{T}(D) = \left\{ \left(\sum_{e \in D} Tr(c_e x^e) \right) \in (\mathbb{F}_{q^n} \to \mathbb{F}_q) : c_e \in \mathbb{F}_{q^n} \right\},$$

where $Tr : \mathbb{F}_{q^n} \to \mathbb{F}_q$ is the field trace function given by $Tr(x) = x + x^q + x^{q^2} + \cdots + x^{q^{n-1}}$.

Let (mod* Q) refer to the operation that maps non-negative integers into $\{0, 1, \ldots, Q - 1\}$ such that a (mod* Q) $= 0$ if $a = 0$, while if $a \neq 0$, then a (mod* Q) $= b$, where $b \in \{1, 2, \ldots, Q - 1\}$ is the unique integer such that $a \equiv b \pmod{Q - 1}$.

Definition 42. *For any $e \in \{0, 1, \ldots, q^n - 1\}$, we say that $e' \in \{0, 1, \ldots, q^n - 1\}$ is a q-shift of e if there exists some nonnegative integer i such that $e' = q^i \cdot e$ (mod* q^n). Furthermore, we define the* shift closure *of e to be the set of all shifts of e:*

$$\overline{\mathrm{shift}}(e) = \{eq^i \ (\textit{mod}^* \ q^n) : i \in \{0, 1, \ldots, n - 1\}.$$

The shift closure *of a set $D \subseteq \{0, 1, \ldots, q^n - 1\}$ is then defined to be the union of the shift closures of its elements:*

$$\overline{\mathrm{shift}}(D) = \bigcup_{e \in D} \overline{\mathrm{shift}}(e).$$

[3] Note that multivariate functions admit a larger class of affine transformations than univariate functions over the corresponding domain. However, each affine transformation of $\mathbb{F}_{2^{st}}$ corresponds to an affine transformation of $(\mathbb{F}_{2^s})^t$ under the isomorphism ϕ. Thus, since we are proving *limitations* of affine-invariant codes, any upper bound on the dimension of univariate affine-invariant LTCs will also apply to multivariate affine-invariant LTCs over the corresponding domain.

Finally, D is said to be shift-closed *if $D = \overline{\mathrm{shift}}(D)$.*

An alternate view of shift-closed sets arises by viewing an element $e \in D$ as a vector in $\{0, 1, \ldots, q-1\}^n$ given by the base q representation of e. The q-shifts of e are precisely the integers whose corresponding vectors (obtained by taking the base q representation) are cyclic shifts of the vector associated with e. A set D is, therefore, shift-closed if the set is closed under taking "cyclic" shifts of the associated base q representations.

Definition 43. *Let $e, e' \in \{0, 1, \ldots, q^n - 1\}$. Let $e = \sum_{i=0}^{n-1} e_i q^i$ and $e' = \sum_{i=0}^{n-1} e'_i q^i$ be the base q representations of e and e', respectively. We say that e' lies in the q-*shadow *of e if $e'_i \leq e_i$ for all $0 \leq i \leq n-1$. We will denote this as $e' \leq_q e$.*

*A set $D \subseteq \{0, 1, \ldots, q^n - 1\}$ is said to be q-*shadow-closed *if*

$$\{e' : e' \leq_q e \textit{ for some } e \in D\} = D.$$

When q is understood, we will say D is shadow-closed*.*

It is known that linear affine-invariant codes can be characterized by their corresponding degree sets.

Theorem 58. *Let $\mathcal{F} \subseteq \{\mathbb{F}_{q^n} \to \mathbb{F}_q\}$ be a linear affine-invariant code. Then, $D = \mathrm{Deg}(\mathcal{F})$ is the unique set $D \subseteq \{0, 1, \ldots, q^n - 1\}$ that is shift-closed and shadow-closed such that $\mathcal{F}$ equals the trace code $\mathcal{T}(D)$. Conversely, if $D \subseteq \{0, 1, \ldots, q^n - 1\}$ is shift-closed and shadow-closed, then $\mathcal{T}(D)$ is a linear affine-invariant code with degree set D.*

Moreover, the dimension of a linear affine-invariant code is given by the size of its degree set.

Theorem 59. *If $\mathcal{F} \subseteq \{\mathbb{F}_{q^n} \to \mathbb{F}_q\}$ is a linear affine-invariant code, then $\dim(\mathcal{F}) = |\mathrm{Deg}(\mathcal{F})|$.*

6.3 Background and Previous Work

We now state some results on binary affine-invariant codes that motivate our work.

Definition 44. *The* 2-weight *of a degree $e \in \{0, 1, \ldots, 2^n - 1\}$, denoted $\mathrm{wt}_2(e)$, is the number of ones in the binary representation of e.*

Recall the definition of a trace code (see Definition 41, with $q = 2$). It is a folklore fact that the Reed-Muller code is equivalent to the univariate code

$$\mathrm{RM}(r, n) = \mathcal{T}(\{e \in \{0, 1, \dots, 2^n - 1\} : \mathrm{wt}_2(e) \leq r\}).$$

Furthermore, the dual of the extended BCH code of distance $d = 2t + 2$ can be expressed as

$$\text{dual-eBCH}(n, t) = \mathcal{T}(\{0, 1, \dots, t\}).$$

Similarly, the extended BCH code itself is expressible as

$$\text{eBCH}(n, t) = \mathcal{T}(D),$$

where $D \subseteq \{0, 1, \dots, 2^n - 1\}$ is the set of all degrees e such that the zeros in the n-bit binary representation of e do not all lie within a cyclic block of length $\log d - 1$. Note that we have

$$\mathrm{RM}(n - \log d, n) \subseteq \text{eBCH}(n, t),$$

and both are linear affine-invariant codes of distance d. Moreover, $\mathrm{RM}(n - \log d, n)$ has dimension

$$\sum_{i=0}^{n-\log d} \binom{n}{i} \approx N - \left(\frac{en}{\log d - 1}\right)^{\log d - 1},$$

while eBCH(n, t) has dimension roughly $N - \frac{dn}{2}$. However, $\mathrm{RM}(n - \log d, n)$ can be tested with $\frac{2N}{d}$ queries [BKS$^+$10, AKK$^+$05]. More specifically, we have the following result (one should think of s as constant, and n as growing in the statement below):

Theorem 60 ([BKS$^+$10])**.** *There exists an absolute constant $\xi > 0$ such that the Reed-Muller code $\mathrm{RM}(n - s, n)$ (of distance 2^s) can be tested with 2^{n-s+1} queries, rejecting a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that is $2^s/3$-far from $\mathrm{RM}(n - s, n)$ with probability at least ξ.*

On the other hand, we cannot hope to test eBCH(n, t) with the same number of queries (for $d > 4$), due to Fact 56 and the fact that dual-eBCH(n, t) has relative distance close to $1/2$ (see [MS81]).

6.3.1 Testable Codes Surpassing Reed-Muller

The authors of [GKS13] construct linear affine-invariant codes of linear locality that contain the generalized Reed-Muller code of appropriate order. More specifically, for $n = \ell m$, where $\ell = \log d - 1$ and m is any positive integer, they present a multivariate affine-invariant LTC $\mathcal{C} \subseteq \{\mathbb{F}_{2^\ell}^m \to \mathbb{F}_2\}$ of block length $N = 2^n$ which satisfies $\dim(\mathcal{C}) = N - (m+1)^\ell = N - \left(1 + \frac{n}{\log d - 1}\right)^{\log d - 1}$. This code contains the binary code $\mathrm{RM}(n - \log d, n)$, and is testable with $2N/d$ queries (where $N = 2^n$).

There is also a univariate analogue of the above codes with identical distance and dimension. See 6.6.1 of Appendix 6.6 for details.

6.3.2 Consequence of the Extended Weil Bound

In [KL11], the authors prove an extension of the Weil bound, which implies that sparse linear affine-invariant codes have relative distance close to $1/2$. The main theorem of [KL11], specialized to our setting (where we set $p = 2$, $\chi(x) \equiv \mathrm{Tr}(x)$ and $g(x) \equiv 0$), can be stated as follows.

Theorem 61 ([KL11])**.** *Let $f(x)$ be the sum of $k \geq 1$ monomials, each of 2-weight at most d. Then, either Tr$(f(x))$ is constant over all $x \in \mathbb{F}_{2^n}$, or*

$$\left|\mathbf{E}_{x \in \mathbb{F}_{2^n}}[\mathit{Tr}(f(x))]\right| \leq 2^{-\frac{n}{4d^2 2^d k}}.$$

This yields a lower bound on the dimension of any sparse linear affine-invariant code that has relative distance much less than $1/2$:

Theorem 62 (consequence of [KL11])**.** *Let $\mathcal{F} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ be a linear affine-invariant code of relative distance $\leq \frac{1}{2} - \delta$, for some $\delta > 0$. Then for any $\epsilon > 0$, $|\mathcal{F}| \geq 2^{\Omega(n^{\frac{3}{2}-\epsilon})}$, i.e.,* $\dim(\mathcal{F}) = \Omega(n^{\frac{3}{2}-\epsilon})$.

This theorem does not appear explicitly in [KL11], but it can be deduced from their techniques. In this section, we show how to prove Theorem 62. For the remainder of this section, assume $\mathcal{F}$ is a linear affine-invariant code, and let $D = \mathrm{Deg}(\mathcal{F})$. Let $R = \{1, 3, 5, \ldots, 2^n - 1\}$ be the set of odd degrees, and set $R' = D \cap R$.

Let us bound the maximum possible 2-weight of a degree in D in terms of $|D|$ and $|R'|$.

Lemma 29. *Let r be the maximum 2-weight of a degree in D. Then, $r \leq \log |D|$.*

Proof. Pick a degree $e \in D$ of 2-weight r. There are exactly 2^r degrees in the shadow of e. Since D is shadow-closed, $2^r \leq |D|$, as desired. □

Lemma 30. *Suppose $|D| > 1$. Let r be the maximum 2-weight of a degree in D. Then, $r \leq \log |R'| + 1$.*

Proof. Observe that we can pick a degree $e \in R'$ of weight $r \geq 1$ (since $|D| > 1$ and D is shift-closed). Note that there are 2^{r-1} odd degrees in the shadow of e. Thus, $2^{r-1} \leq |R'|$, which implies the claim. □

Next, we prove an upper bound on $|R'|$ in terms of $|D|$.

Lemma 31. *Suppose $|D| > 1$. Then, $|R'| \leq \frac{|D| \log^2 |D|}{n}$.*

Proof. Note that for any nonzero degree $e \in D$, there are at least $\frac{n}{\mathrm{wt}_2(e)} \geq \frac{n}{\log |D|}$ distinct shifts of e, by Lemma 29. Moreover, for any nonzero degree $e \in D$, there are at most $\mathrm{wt}_2(e) \leq \log |D|$ shifts of e that lie in R'. Since D contains $|D| - 1$ nonzero degrees, we see that

$$|R'| \leq \frac{|D| - 1}{n/ \log |D|} \cdot \log |D| \leq \frac{|D| \log^2 |D|}{n},$$

as desired. □

Now, we are ready to prove Theorem 62.

Proof of Theorem 62. Suppose the code $\mathcal{F} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ satisfies the hypothesis of Theorem 62. Let $D = \mathrm{Deg}(\mathcal{F})$. Since the code has relative distance $\frac{1}{2} - \delta$, $|D| > 1$.

For the sake of contradiction, assume that $\dim(\mathcal{F}) \leq O(n^{\frac{3}{2}-\epsilon})$ for some $\epsilon > 0$. Then,

$$|D| \leq O(n^{\frac{3}{2}-\epsilon}). \tag{6.2}$$

We have that $\mathcal{F} = \mathcal{T}(R' \cup \{0\})$, since each nonzero degree in D has some shift contained in R'. Therefore, any $h(x) \in \mathcal{F}$ can be written as $\mathrm{Tr}(f(x))$ for some $f(x)$ that is a sum of at most $k = |R'| + 1$ monomials. Moreover, by Lemma 30, we can guarantee that each of these monomials has 2-weight at most $d = \log |R'| + 1$. Then, Theorem 61 implies that either h is constant or

$$|\mathbf{E}_{x \in \mathbb{F}_{2^n}}[h(x)]| \leq 2^{-\frac{n}{8(\log |R'|+1)^2 \cdot |R'|(|R'|+1)}}.$$

Assume h is not constant. By Lemma 31, we have $|R'| \leq \frac{|D| \log^2 |D|}{n}$, and so,

$$|\mathbf{E}_{x \in \mathbb{F}_{2^n}}[h(x)]| \leq \exp\left(-\Omega\left(\frac{n^3}{|D|^2 \log^4 |D| \cdot (\log(|D| \log^2 |D|) - \log n + 1)^2}\right)\right)$$

It is now straightforward to observe that (6.2) implies that

$$|\mathbf{E}_{x\in\mathbb{F}_{2^n}}[h(x)]| \to 0$$

as $n \to \infty$. However, this implies that the relative Hamming weight of any nonconstant $h(x)$ approaches $\frac{1}{2}$ in the limit $n \to \infty$. Furthermore, any (nonzero) constant $h(x)$ has relative Hamming weight 1. Hence, the relative distance of $\mathcal{F}$ approaches $\frac{1}{2}$ in the limit $n \to \infty$, which contradicts the assumption that the relative distance is $\leq \frac{1}{2} - \delta$. This concludes the proof of Theorem 62. □

Because we are interested in very large codes $\mathcal{C}$ whose duals are sparse, we can apply Theorem 62 to $\mathcal{F} = \mathcal{C}^\perp$ to obtain an upper bound on the dimension of $\mathcal{C}$. The requirement $d \geq 5$ used below ensures that $2N/d < 1/2$.

Corollary 63. *If $\mathcal{C}$ is a linear affine-invariant code of distance $d \geq 5$ testable with $\frac{2N}{d}$ queries, then $\dim(\mathcal{C}^\perp) \geq n^{3/2-o(1)}$.*

Remark 64. *Although we are able to prove much stronger lower bounds in the following section, our results only hold when $\mathcal{C}^\perp$ contains (the indicator of) a low-dimensional subspace. The work of [KL11] does not require this assumption.*

6.4 Upper Bounds on the Dimension of $\mathcal{C}$

In this section, we prove the following bound on the co-dimension of certain families of affine-invariant LTCs:

Theorem 65. *Let $\mathcal{C} \supseteq \mathrm{RM}(n - \log d, n)$ be a linear affine-invariant code of block length $N = 2^n$ that has distance d and is testable with $\frac{2N}{d}$ queries. Then, $\dim(\mathcal{C}^\perp) \geq \left(\frac{n}{\log^2 d}\right)^{\log d - 1}$.*

Note that this bound is much stronger than that of Corollary 63, which followed from the results of [KL11]. However, our bound only holds in the special case when $\mathcal{C} \supseteq \mathrm{RM}(n - \log d, n)$, unlike the results of [KL11]. Strengthening the results of [KL11] in the more general setting is a challenging open problem.

By Theorem 59, to show that $\dim(\mathcal{C})$ is small, it suffices to show that $\mathrm{Deg}(\mathcal{C})$ is small. Thus, we will show that under our assumptions, there are *many* degrees which cannot be in $\mathrm{Deg}(\mathcal{C})$. At a high level, we start with a monomial which violates some dual constraint, and use affine invariance to translate it to many other monomials which also violate this dual constraint.

We first show an equivalent condition for a monomial to violate a dual constraint (Lemma 32), which can already be used to show that $\mathcal{C} \subseteq$ eBCH(n, t) (Theorem 66). The proof itself is given in Section 6.4.3, where we consider a degree which violates some dual constraint, and show that the bits of its binary expansion can be "moved around" to give new degrees which also violate this dual constraint (Lemmas 68 and 69). Bounding the number of such degrees gives us our main theorem, Theorem 65.

We will assume throughout that $\mathcal{C} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ is affine-invariant, contains the Reed-Muller code $\mathrm{RM}(n - \log d, n)$, and is testable with $2N/d$ queries. Note that the containment assumption implies that $\mathrm{Deg}(\mathcal{C})$ contains all degrees of 2-weight at most $n - \log d$ (see the discussion about degree sets of Reed-Muller codes at the beginning of Section 6.3). Furthermore, Fact 56 guarantees the existence of some $f \in \mathcal{C}^{\perp} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ of Hamming weight $2N/d$. Since $\mathcal{C}^{\perp}$ is affine-invariant, we have that $g = f \circ A \in \mathcal{C}^{\perp}$ for any affine transformation $A : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$. In particular, choose A to be an invertible transformation that maps 0 to some $x \in \mathbb{F}_{2^n}$ with $f(x) \neq 0$. Then, g has Hamming weight $2N/d$ and is supported on 0.

Since $\mathcal{C}^{\perp}$ is contained in the dual of $\mathrm{RM}(n - \log d, n)$, all dual codewords of Hamming weight $2N/d$ correspond to (indicators of) affine subspaces of dimension $n - \log d + 1$ (see [PHE98]). Therefore, g must be (the indicator of) an affine subspace S of dimension $n - \log d + 1$. Moreover, since g is supported on 0, S must in fact be a *linear* subspace.

6.4.1 Matrix Determinant Formulation

In this section, we give a necessary condition (Equation (6.4)) for a degree to be in the degree set of our code $\mathcal{C}$, whenever the indicator of the subspace S lies in $\mathcal{C}^{\perp}$.

Recall that an affine-invariant code is specified by its degree set. Thus, if $e \in \mathrm{Deg}(\mathcal{C})$ and the indicator vector of a subspace S is in the dual code $\mathcal{C}^{\perp}$, then we must have

$$\sum_{\alpha \in S} \alpha^e = 0. \tag{6.3}$$

We will often abuse notation and say that if (6.3) holds, then S is *orthogonal* to e, or e *passes* S.

We have assumed that any degree e of 2-weight at most $n - \log d$ is in $\mathrm{Deg}(\mathcal{C})$. Thus, let us consider which degrees e of 2-weight exactly $n - \log d + 1$ can be contained in $\mathrm{Deg}(\mathcal{C})$. The following lemma gives an equivalent condition for when a subspace of a certain dimension k is orthogonal to a degree of the 2-weight k. Note that we are interested in the special case $k = n - \log d + 1$.

Lemma 32. *Suppose $e = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_k}$ is a degree of 2-weight $k \geq 1$, for i_j distinct. Suppose S is a subspace of dimension k, and let $\alpha_1, \alpha_2, \ldots, \alpha_k$ be an $\mathbb{F}_2$-basis for S. Then S is orthogonal to e if and only if the following determinant is zero:*

$$M_e(\alpha_1, \alpha_2, \ldots, \alpha_k) := \begin{pmatrix} \alpha_1^{2^{i_1}} & \alpha_2^{2^{i_1}} & \cdots & \alpha_k^{2^{i_1}} \\ \alpha_1^{2^{i_2}} & \alpha_2^{2^{i_2}} & \cdots & \alpha_k^{2^{i_2}} \\ \vdots & \vdots & & \vdots \\ \alpha_1^{2^{i_k}} & \alpha_2^{2^{i_k}} & \cdots & \alpha_k^{2^{i_k}} \end{pmatrix} \tag{6.4}$$

Proof. S is orthogonal to e if and only if $\sum_{\alpha \in S} \alpha^e = 0$. Note that

$$\begin{aligned} \sum_{\alpha \in S} \alpha^e &= \sum_{\lambda_1, \ldots, \lambda_k \in \{0,1\}} \prod_{j=1}^{k} (\lambda_1 \alpha_1 + \cdots + \lambda_k \alpha_k)^{2^{i_j}} \\ &= \sum_{\pi \in S_n} \prod_{j=1}^{k} \alpha_j^{2^{i_{\pi(j)}}}, \end{aligned}$$

where the last sum ranges over all permutations of $\{1, 2, \ldots, n\}$. The final line follows because any term $\alpha_1^{t_1} \alpha_2^{t_2} \cdots \alpha_k^{t_k}$ that has some t_j of 2-weight at least 2 must also have some $t_j = 0$, hence implying that such a term must occur an even number of times in the sum. Since we are working over fields of characteristic 2, it follows that such a term cannot have a nonzero coefficient. Moreover, the above quantity is equal to the permanent of $M_e(\alpha_1, \alpha_2, \ldots, \alpha_k)$, which, over fields of characteristic 2, is equal to $\det M_e(\alpha_1, \alpha_2, \ldots, \alpha_k)$. This proves the claim. □

6.4.2 Warm-up: Containment in Extended BCH

To give some idea of our approach, let us first show how to use the determinant formulation of Lemma 32 to prove that if an LTC satisfies our desired conditions, then it must be contained inside an extended BCH code of the same distance.

Loosely, we find a nontrivial degree (e^*) which cannot be in $\mathrm{Deg}(\mathcal{C})$, and use the fact that $\mathrm{Deg}(\mathcal{C})$ is closed under cyclic "shifts" to obtain more degrees which are not in $\mathrm{Deg}(\mathcal{C})$.

Theorem 66. *Suppose $\mathcal{C}$ is a linear affine-invariant code of distance $d = 2t + 2$ that contains $\mathrm{RM}(n - \log d, n)$ and is locally testable with $\frac{2N}{d}$ queries. Then, $\mathcal{C} \subseteq$ eBCH(n, t).*

Proof. First, we consider the degree $e^* = 2^0 + 2^1 + 2^2 + \cdots + 2^{n - \log d}$ of 2-weight $n - \log d + 1$. We will show that $e^* \notin \mathrm{Deg}(\mathcal{C})$.

Let S be an arbitrary subspace of dimension $k = n - \log d + 1$. We will show that S cannot be orthogonal to e^*. Let $\alpha_1, \alpha_2, \ldots, \alpha_k$ be an $\mathbb{F}_2$-basis for S. Then,

$$M_{e^*}(\alpha_1, \ldots, \alpha_k) = \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_k \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_k^2 \\ \vdots & \vdots & & \vdots \\ \alpha_1^{2^{k-1}} & \alpha_2^{2^{k-1}} & \cdots & \alpha_k^{2^{k-1}} \end{pmatrix},$$

which has been studied as the (transpose) Moore matrix, whose (i, j) entry is $\alpha_j^{2^{i-1}}$ (see [Moo96]). The determinant of the matrix is known to be

$$\prod_{\lambda_1, \lambda_2, \ldots, \lambda_k \in \{0,1\} \text{ not all zero}} (\lambda_1\alpha_1 + \lambda_2\alpha_2 + \cdots + \lambda_k\alpha_k),$$

i.e., the product of all non-trivial $\mathbb{F}_2$-linear combinations of $\alpha_1, \alpha_2, \ldots, \alpha_k$. Since the α_i are $\mathbb{F}_2$-linearly independent by choice, it follows that the above determinant is nonzero. Thus, Lemma 32 implies that S cannot be orthogonal to e^*. Since S was arbitrary, any $\mathcal{C}$ whose degree set contains e^* cannot have dual distance $\frac{2N}{d}$ and would therefore not be locally testable with the desired locality.

Now recall that for $d = 2t + 2$, $\text{Deg}(\text{eBCH}(n, t)) = \{0, 1, \ldots, 2^n - 1\} \setminus T$, where T is the set of all degrees e for which the zeros in the n-bit base-2 representation of e are contained in a consecutive (cyclic) block of size $\log d - 1$ (see the discussion about degree sets of extended BCH codes at the beginning of Section 6.3). Note that for any $e \in T$, there is some cyclic shift of e^* in its shadow (see Definition 43, with $q = 2$). Since $\text{Deg}(\mathcal{C})$ does not contain e^*, and affine-invariant codes are closed under shifts and shadows (by Theorem 58), it follows that $\text{Deg}(\mathcal{C}) \cap T = \emptyset$. Hence, $\text{Deg}(\mathcal{C}) \subseteq \text{Deg}(\text{eBCH}(n, t))$, and so, $\mathcal{C} \subseteq \text{eBCH}(n, t)$. □

6.4.3 Dimension Bound via Local Transformations of Degree

Now, we show that for any degree e of 2-weight $n - \log d + 1$ that does not pass a fixed subspace S of dimension $n - \log d + 1$, we can perform a slight perturbation to e to obtain another degree e' of 2-weight $n - \log d + 1$ that does not pass S. In other words, for any subspace S, the existence of one degree that does not pass S implies many others.

First, let us state some facts which will be useful for the proof of the main result.

Fact 67. *Let $\lambda \in \mathbb{F}_{2^n}$ be nonzero. Then, a subspace S is orthogonal to a degree e if and only if the subspace $\lambda S = \{\lambda s : s \in S\}$ is orthogonal to e.*

Lemma 33. *Let $m < n$ and $\alpha_1, \alpha_2, \dots, \alpha_m \in \mathbb{F}_{2^n}$. There exists a nonzero $\lambda \in \mathbb{F}_{2^n}$ such that*

$$\mathit{Tr}(\lambda\alpha_1) = \mathit{Tr}(\lambda\alpha_2) = \cdots = \mathit{Tr}(\lambda\alpha_m) = 0. \tag{6.5}$$

Proof. As $(\mathrm{Tr}(\lambda\alpha_1), \dots, \mathrm{Tr}(\lambda\alpha_m)) \in \{0,1\}^m$ for all $\lambda \in \mathbb{F}_{2^n} \setminus \{0\}$, the pigeonhole principle implies that there exist two distinct $\lambda_1, \lambda_2 \in \mathbb{F}_{2^n} \setminus \{0\}$ for which $(\mathrm{Tr}(\lambda_i\alpha_1), \mathrm{Tr}(\lambda_i\alpha_2), \dots, \mathrm{Tr}(\lambda_i\alpha_m))$ is identical for $i = 1, 2$. Thus, by linearity of trace, we see that (6.5) holds for $\lambda = \lambda_1 - \lambda_2$. □

Now, we prove one of the main technical theorems.

Theorem 68. *Suppose S is a subspace of dimension $k = n - \log d + 1$. Let $e = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_k}$ be a degree of 2-weight k that does not pass S. Then, for any integer $1 \le r \le k$, there exists $u \in \{0, 1, \dots, n-1\} \setminus \{i_1, i_2, \dots, i_k\}$ such that $e' = e - 2^{i_r} + 2^u$ does not pass S.*

Proof. Let $\{j_1, j_2, \dots, j_\ell\} = \{0, 1, \dots, n-1\} \setminus \{i_1, i_2, \dots, i_k\}$. Let $\alpha_1, \alpha_2, \dots, \alpha_k$ be a basis for S. Then, by Lemma 33, there exists some nonzero $\lambda \in \mathbb{F}_{2^n}$ such that $\mathrm{Tr}(\lambda\alpha_i) = 0$ for each i. Scaling S by λ, we may assume that $\mathrm{Tr}(\alpha_1) = \mathrm{Tr}(\alpha_2) = \cdots = \mathrm{Tr}(\alpha_k) = 0$.

For ease of notation, we will write $\alpha^{[i]}$ for α^{2^i}. Consider the matrix

$$M = \begin{pmatrix} \alpha_1^{[i_1]} & \alpha_2^{[i_1]} & \cdots & \alpha_k^{[i_1]} \\ \vdots & \vdots & & \vdots \\ \alpha_1^{[i_{r-1}]} & \alpha_2^{[i_{r-1}]} & \cdots & \alpha_k^{[i_{r-1}]} \\ \sum_{t=1}^{\ell} \alpha_1^{[j_t]} & \sum_{t=1}^{\ell} \alpha_2^{[j_t]} & \cdots & \sum_{t=1}^{\ell} \alpha_k^{[j_t]} \\ \alpha_1^{[i_{r+1}]} & \alpha_2^{[i_{r+1}]} & \cdots & \alpha_k^{[i_{r+1}]} \\ \vdots & \vdots & & \vdots \\ \alpha_1^{[i_k]} & \alpha_2^{[i_k]} & \cdots & \alpha_k^{[i_k]} \end{pmatrix}$$

We observe that $\det M$ is equal to the determinant of the following matrix M' which is obtained by replacing the r^{th} row of M with the sum of all rows of M:

$$M' = \begin{pmatrix} \alpha_1^{[i_1]} & \cdots & \alpha_k^{[i_1]} \\ \vdots & & \vdots \\ \alpha_1^{[i_{r-1}]} & \cdots & \alpha_k^{[i_{r-1}]} \\ \sum_{0\le t<n, t\neq i_r} \alpha_1^{[t]} & \cdots & \sum_{0\le t<n, t\neq i_r} \alpha_k^{[t]} \\ \alpha_1^{[i_{r+1}]} & \cdots & \alpha_k^{[i_{r+1}]} \\ \vdots & & \vdots \\ \alpha_1^{[i_k]} & \cdots & \alpha_k^{[i_k]} \end{pmatrix}$$

However, note that $\sum_{0\le t<n, t\neq i_r} \alpha_s^{[t]} = \alpha_s^{[i_r]} + \text{Tr}(\alpha_s) = \alpha_s^{[i_r]}$ for $s = 1, 2, \ldots, k$. Hence, $M' = M_e(\alpha_1, \ldots, \alpha_k)$. By Lemma 32, since S is not orthogonal to e, we must have that $\det(M_e(\alpha_1, \ldots, \alpha_k)) \neq 0$. It follows that $\det M \neq 0$. Note that

$$\det M = \sum_{s=1}^{\ell} \det M_{e_s}(\alpha_1, \ldots, \alpha_k),$$

where $e_s = e - 2^{i_r} + 2^{j_s}$. Thus, there exists some s for which $\det M_{e_s}(\alpha_1, \ldots, \alpha_k) \neq 0$. Hence, we conclude that the desired statement holds for $u = j_s$. □

Theorem 68 shows that for a given degree e that does not pass a fixed subspace S, one can shift any 1 in the binary representation of e to some position that is currently occupied by a 0 and obtain another degree that does not pass S. Next, we try to prove an analogue (Theorem 69) which allows us to shift any desired 0 to a position occupied by a 1. First, we prove a lemma.

Lemma 34. *Suppose* $\alpha_1, \alpha_2, \ldots, \alpha_k \in F_{2^n}$ *are* $\mathbb{F}_2$*-linearly independent, and let* $v_0, \ldots, v_{n-1} \in \mathbb{F}_{2^n}^k$ *be defined as*

$$v_i = (\alpha_1^{2^i}, \alpha_2^{2^i}, \ldots, \alpha_k^{2^i}),$$

where $k = n - \log d + 1$. *Then any set of* $(n/\log d)$ *of the* v_i *is linearly independent over* F_{2^n}.

Proof. Let $h = \frac{n}{\log d}$ and suppose, for the sake of contradiction, that $\lambda_1 v_{i_1} + \lambda_2 v_{i_2} + \cdots + \lambda_h v_{i_t} = 0$, where $i_1, i_2, \ldots, i_h$ are distinct, and not all of the $\lambda_1, \lambda_2, \ldots, \lambda_h \in \mathbb{F}_{2^n}$ are zero. Without loss of generality, suppose $0 \le i_1 < i_2 < \cdots < i_h \le n-1$. Let $j_r = (i_{r+1} - i_r) \pmod{n}$, where $i_{h+1} = i_1$. Since $j_1 + j_2 + \cdots + j_h = n$, there exists some r such that

$j_r \geq \frac{n}{h} = \log d$. Then, note that $v_{i_{r+1}}, v_{i_{r+1}+1}, \cdots, v_{i_{r+1}+(k-1)}$ are linearly independent (where subscripts on v are modulo n): Letting $e^* = 2^0 + 2^1 + \cdots + 2^{k-1}$, we have

$$\det \begin{pmatrix} v_{i_{r+1}} \\ v_{i_{r+1}+1} \\ \vdots \\ v_{i_{r+1}+k-1} \end{pmatrix} = \left(\det M_{e^*}(\alpha_1, \dots, \alpha_k)\right)^{2^{i_{r+1}}} \neq 0,$$

where the last statement is shown in the proof of Theorem 66. However, $v_{i_1}, v_{i_2}, \dots, v_{i_h}$ appear among $v_{i_{r+1}}, v_{i_{r+1}+1}, \dots, v_{i_{r+1}+(k-1)}$. Thus, we obtain a contradiction. □

Theorem 69. *Suppose S is a subspace of dimension $k = n - \log d + 1$. Let $e = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_k}$ be a degree of 2-weight k that does not pass S. Then, for any integer $0 \leq u \leq n-1$ with $u \notin \{i_1, i_2, \dots, i_k\}$, there exist at least $\frac{n}{\log d} - 1$ values of $r \in [k]$ for which $e + 2^u - 2^{i_r}$ is a degree that does not pass S.*

Proof. Let $u \notin \{i_1, i_2, \dots, i_k\}$, and let $\alpha_1, \alpha_2, \dots, \alpha_k$ be a basis for S. Because e does not pass S, we know that the matrix $M = M_e(\alpha_1, \alpha_2, \dots, \alpha_k)$ has a nonzero determinant. Write $w_t = (\alpha_1^{2^{i_t}}, \alpha_2^{2^{i_t}}, \dots, \alpha_k^{2^{i_t}})$ for $t = 1, 2, \dots, k$, i.e., w_t is the t^{th} row of M. Also, let $v = (\alpha_1^{2^u}, \alpha_2^{2^u}, \dots, \alpha_k^{2^u})$. Since M has nonzero determinant, its row span is all of $\mathbb{F}_{2^n}^k$, and we can find $\lambda_1, \lambda_2, \dots, \lambda_k \in \mathbb{F}_{2^n}$ such that $v = \lambda_1 w_1 + \lambda_2 w_2 + \cdots + \lambda_k w_k$.

Suppose $\lambda_j \neq 0$. Then, the linear dependence

$$\sum_{\substack{1 \leq i \leq k \\ i \neq j}} \lambda_i w_i + \lambda_j (w_j + \lambda_j^{-1} v) = 0$$

implies that

$$0 = \det \begin{pmatrix} w_1 \\ \vdots \\ w_{j-1} \\ w_j + \lambda_j^{-1} v \\ w_{j+1} \\ \vdots \\ w_k \end{pmatrix} = \det M + \lambda_j^{-1} \det M_{e'}(\alpha_1, \dots, \alpha_k),$$

where $e' = e + 2^u - 2^{i_j}$. Since $\det M \neq 0$, we have $\det M_{e'}(\alpha_1, \alpha_2, \dots, \alpha_k) \neq 0$, implying that e' does not pass S. To conclude, simply note that Lemma 34 implies that there are at least $\frac{n}{\log d} - 1$ values of j for which $\lambda_j \neq 0$. Thus, the desired conclusion follows. □

Remark 70. *The bounds in Theorems 68 and 69 are tight, as they are achieved by the (univariate analogue) of the codes of [GKS13]. See 6.6.2 in Appendix 6.6 for details.*

Now, we are ready to prove the main theorem, which proves a lower bound on $\dim(\mathcal{C}^{\perp})$.

Theorem 65. *Let* $\mathcal{C} \supseteq \mathrm{RM}(n - \log d, n)$ *be a linear affine-invariant code of block length* $N = 2^n$ *that has distance* d *and is testable with* $\frac{2N}{d}$ *queries. Then,* $\dim(\mathcal{C}^{\perp}) \geq \left(\frac{n}{\log^2 d}\right)^{\log d - 1}$.

Proof. Fix a subspace S of dimension $n - \log d + 1$ whose indicator is in $\mathcal{C}^{\perp}$. Let $k = n - \log d + 1$. Recall that $e^* = 2^0 + 2^1 + \cdots + 2^{k-1}$ does not pass S.

Consider the following procedure. Let $e_k = e^*$. Then, for $j = k, k+1, \ldots, n-1$ (in succession), we perform either one of the following steps:

- Set $e_{j+1} = e_j$.
- Choose an $i_j \in \{0, 1, \ldots, n-1\}$ such that 2^{i_j} appears in the binary representation of e_j and so that $e_j + 2^j - 2^{i_j}$ does not pass S. Set $e_{j+1} = e_j + 2^j - 2^{i_j}$.

It is clear that at the end of the procedure, e_n will be a degree of 2-weight k that does not pass S. Moreover, for each j in the procedure, there will be at least $\frac{n}{\log d}$ choices for setting e_{j+1} (by Theorem 69). On the other hand, any final e_n could have been obtained in at most $(\log d)^{\log d - 1}$ ways. Thus, it follows that there are at least $\left(\frac{n}{\log d}\right)^{\log d - 1} \Big/ (\log d)^{\log d - 1} = \left(\frac{n}{\log^2 d}\right)^{\log d - 1}$ degrees that do not pass S. □

6.5 Reed-Muller Containment Assumption

In this work, we have analyzed affine-invariant codes $\mathcal{C} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ that contain $\mathrm{RM}(n - \log d, n)$. Let us provide some justification for this assumption by showing that any linear affine-invariant code with large dimension must contain a Reed-Muller code of large order.

Theorem 71. *Suppose* $\mathcal{C} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ *is a linear affine-invariant code such that* $\mathrm{RM}(s, n) \not\subseteq \mathcal{C}$*, for some* $s = n - (\log d - 1)\log(n + \log d - 1) + \Omega_d(1)$*. Then,* $\dim(\mathcal{C}) \leq 2^n - \left(1 + \frac{n}{\log d - 1}\right)^{\log d - 1}$.

Proof. Suppose $\mathcal{C}$ satisfies the conditions of the hypothesis. Recall that $\mathrm{RM}(s, n)$ is the trace code with degree set consisting of precisely those $0 \leq e < 2^n$ of 2-weight at most s. Thus, there exists some degree of 2-weight at most s that does not appear in $\mathrm{Deg}(\mathcal{C})$. Since the degree set of $\mathcal{C}$ is shadow-closed, it then follows that there exists e of 2-weight *exactly* s that does not appear in $\mathrm{Deg}(\mathcal{C})$. Note that there are n shifts of e (possibly repeated). For any shift e' of e, there are 2^{n-s} degrees that contain e' in their shadow, for a total of $n \cdot 2^{n-s}$. Moreover, any of these degrees may appear up to n times (since each degree contains at most n shifts of e in its shadow). Thus, there are at least $n \cdot 2^{n-s}/n = 2^{n-s}$ distinct degrees that cannot be in $\mathrm{Deg}(\mathcal{C})$. This shows that

$$\dim(\mathcal{C}) \leq 2^n - 2^{n-s}.$$

Thus when $s = n - (\log d - 1)\log(n + \log d - 1) + \Omega_d(1)$, we have

$$\dim(\mathcal{C}) \leq 2^n - \left(1 + \frac{n}{\log d - 1}\right)^{\log d - 1}.$$

□

Therefore, any affine-invariant code that is expected to improve on the testable codes of [GKS13] and [HRZS13] must contain a Reed-Muller code of order $n - O_d(1)\log n$. However, note that the above theorem holds for *any* linear affine-invariant code and does not use testability. It seems that using the testability assumption should yield a tighter bound, which is a promising direction for future work.

6.6 Univariate Constructions of Codes

Recall that [GKS13] gives a linear affine-invariant code $\mathcal{C} \subseteq \{\mathbb{F}_{2^\ell}^m \to \mathbb{F}_2\}$ with block length $N = 2^n$, where $n = \ell m$. For $\ell = \log d - 1$, the code has distance d and is testable with $2N/d$ queries. Moreover, $\mathcal{C}$ contains the multivariate Reed-Muller code $\mathrm{RM}(n - \log d, n)$.

The above code is obtained by "lifting" a parity check code of smaller block length and happens to be *multivariate*. In our work, we are concerned with dimension bounds on *univariate* codes. As it turns out, the code of [GKS13] has a univariate analogue, i.e., a subset of $\{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$. We provide a construction of this univariate code which does not involve lifting. For the sake of convenience, we state the important properties of the code below:

Theorem 72. *Suppose $N = 2^n$, $d \geq 2$, and $\ell = \log d - 1$ such that $\ell \mid n$. Then, there exists a linear affine-invariant LTC $\mathrm{RM}(n-\ell-1, n) \subseteq \mathcal{C} \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$ of distance d that is testable with $2N/d$ queries and has dimension*

$$\dim(\mathcal{C}) = N - \left(1 + \frac{n}{\ell}\right)^{\ell} = N - \left(1 + \frac{\log N}{\log d - 1}\right)^{\log d - 1}.$$

Note that the dimension of the code essentially matches the upper bound on $\dim(\mathcal{C})$ implied by Theorem 65 (up to some lower-order factors involving d). While we present a construction of the code, we do not prove here that the code is an LTC (the proof of testability can be found in [GKS13]).

6.6.1 Subspaces from Subfields

We now try to construct a code with the properties listed in Theorem 72. Let N, n, d, and ℓ be as defined in the theorem statement. Again, we consider $\mathcal{C} \supseteq \mathrm{RM}(n-\ell-1, n)$. Recall from Fact 56 that in order for $\mathcal{C}$ to be testable with the desired locality, there must be a codeword in $w \in \mathcal{C}^{\perp}$ of Hamming weight at most $2N/d$ such that $w_0 \neq 0$. It is known that $\mathrm{RM}(n-\ell-1, n)$ has dual distance $2N/d$, and the dual codewords of minimum weight are precisely the affine subspaces of dimension $n - \ell$. Hence, w must be (the indicator of) a *linear* subspace S of the aforementioned dimension.

Hence, we will consider a fixed subspace S of dimension $n - \ell$ and consider which degrees we can take in $\mathrm{Deg}(\mathcal{C})$. We will say that a degree e *passes* the subspace S if

$$\sum_{a \in S} a^e = 0.$$

The above condition is necessary for us to be able to take e in $\mathrm{Deg}(\mathcal{C})$.

Assume $\ell \mid n$, so that $\mathbb{F}_{2^\ell}$ is a subfield of $\mathbb{F}_{2^n}$. Write $n = \ell m$. We can then consider subspaces S of the form

$$S = \lambda_1 \mathbb{F}_{2^\ell} + \lambda_2 \mathbb{F}_{2^\ell} + \cdots + \lambda_{m-1} \mathbb{F}_{2^\ell}, \tag{6.6}$$

where $\lambda_1, \lambda_2, \ldots, \lambda_{m-1} \in \mathbb{F}_{2^n}$ and λA is used to mean $\{\lambda a : a \in A\}$.

Now, a degree $e = 2^{i_1} + 2^{i_2} + \cdots + 2^{i_u}$ passes S if and only if

$$\begin{aligned}
0 = \sum_{a \in S} a^e &= \sum_{a \in S} \prod_{j=1}^{u} a^{2^{i_j}} \\
&= \sum_{a_1,\ldots,a_{m-1} \in \mathbb{F}_{2^\ell}} \prod_{j=1}^{u} (\lambda_1 a_1 + \cdots + \lambda_{m-1} a_{m-1})^{2^{i_j}} \\
&= \sum_{a_1,\ldots,a_{m-1} \in \mathbb{F}_{2^\ell}} \prod_{j=1}^{u} \sum_{k=1}^{m-1} (\lambda_k a_k)^{2^{i_j}} \\
&= \sum_{a_1,\ldots,a_{m-1} \in \mathbb{F}_{2^\ell}} \sum_{e_1,\ldots,e_{m-1}} \prod_{j=1}^{m-1} (\lambda_j a_j)^{e_j} \\
&= \sum_{e_1,\ldots,e_{m-1}} \lambda_1^{e_1} \cdots \lambda_{m-1}^{e_{m-1}} \prod_{j=1}^{m-1} \left(\sum_{a \in \mathbb{F}_{2^\ell}} a^{e_j} \right), \qquad (6.7)
\end{aligned}$$

where in the last two equations, $e_1, \ldots, e_{m-1}$ range over all $e_1, \ldots, e_{m-1}$ with distinct supports in their binary expansion, such that $e_1 + \cdots + e_{m-1} = e$. Observe that $\sum_{a \in \mathbb{F}_{2^t}} a^{e_j} \neq 0$ if and only if e_j is a positive integral multiple of $2^\ell - 1$. Hence, the above condition would be guaranteed for e if there happens to be no way to write e as a sum $e = e_1 + e_2 + \cdots + e_{m-1}$ such that **(1.)** $e_1, \ldots, e_{m-1}$ have distinct supports in their binary expansion, and **(2.)** $e_1, e_2, \ldots, e_{m-1}$ are all positive multiples of $2^\ell - 1$.

Now, it will be convenient to reason about degrees in terms of a matrix form.

Definition 45. *Let $0 \leq e < 2^n$. Moreover, let $e = b_0 2^0 + b_1 2^1 + \cdots + b_{n-1} 2^{n-1}$ be the binary representation of e (where $b_0, b_1, \ldots, b_{n-1} \in \{0, 1\}$). Then, define the* block matrix representation *of e to be the following $m \times \ell$ matrix:*

$$\begin{pmatrix} b_{n-\ell} & b_{n-\ell+1} & \cdots & b_{n-1} \\ \vdots & \vdots & & \vdots \\ b_\ell & b_{\ell+1} & \cdots & b_{2\ell-1} \\ b_0 & b_1 & \cdots & b_{\ell-1} \end{pmatrix}.$$

Furthermore, for $j = 0, 1, \ldots, \ell - 1$, we define the j-shifted row projection *of e, denoted $proj_j(e)$, as*

$$proj_j(e) = \sum_{i=0}^{n-1} b_i 2^{((i+j) \, mod \, \ell)}.$$

In other words, proj$_j(e)$ is obtained by taking the block matrix representation of e, cyclically shifting its columns by j to the right, and then taking the inner product of the vector $(2^0, 2^1, \ldots, 2^{n-1})$ with the row sum of the resulting matrix.

Note the following easy property about row projections.

Lemma 35. *For any $j = 0, 1, \ldots, \ell - 1$, we have that proj$_j(e) \equiv 2^j e \pmod{2^\ell - 1}$. In particular, proj$_j(e) \equiv 0 \pmod{2^\ell - 1}$ if and only if $e \equiv 0 \pmod{2^\ell - 1}$.*

Proof. As usual, let $e = b_0 2^0 + \cdots + b_{n-1} 2^{n-1}$ be the binary representation of e. Note that

$$\begin{aligned} \text{proj}_j(e) &= \sum_{i=0}^{n-1} b_i 2^{((i+j) \bmod \ell)} \\ &\equiv 2^j e \pmod{2^\ell - 1}, \end{aligned}$$

which proves the first part of the claim. The second part of the claim is a simple consequence of the first part. □

Theorem 73. *Suppose e is a degree whose block matrix representation has at least two zeros in some column. Then, e passes any $(n - \log d + 1)$-dimensional subspace S of the form (6.6).*

Proof. Recall (6.7). Suppose e satisfies the hypothesis of the claim. As noted before, it suffices to show that there is no way to write e as a sum $e = e_1 + e_2 + \cdots + e_{m-1}$ such that **(1.)** $e_1, \ldots, e_{m-1}$ have distinct supports in their binary expansion, and **(2.)** $e_1, e_2, \ldots, e_{m-1}$ are all positive multiples of $2^\ell - 1$.

For the sake of contradiction, assume that there does exist a decomposition $e = e_1 + e_2 + \cdots + e_{m-1}$ satisfying **(1.)** and **(2.)**. Also, suppose the j^{th} column of the block matrix representation of e contains at least two zeros. Then, by Lemma 35, we have that for $i = 1, 2, \ldots, m-1$,

$$\text{proj}_{\ell-j}(e_i) \equiv 2^{\ell-j} e_i \equiv 0 \pmod{2^\ell - 1}.$$

Moreover, since e_i is positive, we must have that proj$_{\ell-j}(e_i) > 0$. Thus, proj$_{\ell-j}(e_i) \geq 2^\ell - 1$. It follows that

$$\text{proj}_{\ell-j}(e) = \sum_{i=1}^{m-1} \text{proj}_{\ell-j}(e_i) \geq (m-1)(2^\ell - 1). \tag{6.8}$$

On the other hand, since there are at least two zeros in the j^{th} column of the block matrix representation of e, we have

$$\begin{aligned}\text{proj}_{\ell-j}(e) &\leq m(2^0 + 2^1 + \cdots + 2^{\ell-1}) - 2 \cdot 2^{\ell-1} \\ &= (m-1)(2^\ell - 1) - 1,\end{aligned}$$

which contradicts (6.8). Hence, **(1.)** and **(2.)** cannot be satisfied, and the desired result follows. □

Thus, let us define $D \subseteq \{0, 1, \ldots, 2^n - 1\}$ by

$$\begin{aligned}D = \{0 \leq e \leq 2^n - 1 : &\text{ the block matrix rep. of } e \\ &\text{contains at least two zeros in some column}\}. \end{aligned} \tag{6.9}$$

It is easy to see that D is shift-closed and shadow-closed. Thus, $\mathcal{T}(D) \subseteq \{\mathbb{F}_{2^n} \to \mathbb{F}_2\}$. Moreover, for none of the degrees in D can the zeros in the n-bit binary representation lie in a cyclic block of length $\log d - 1$ (this is guaranteed by the condition that there are two zeros in some column of the block matrix representation). Thus, $\mathcal{T}(D) \subseteq$ eBCH$(n, (d-2)/2)$. Combining this with $\mathrm{RM}(n - \log d, n) \subseteq \mathcal{T}(D)$ shows that $\mathcal{T}(D)$ has distance exactly d. Moreover, by Theorem 73, all $e \in D$ simultaneously pass a common subspace S of dimension $n - \log d + 1$, which means that the distance of the dual code is $2N/d$.

Finally, recall from Theorem 59 that $\dim(\mathcal{T}(D)) = |\operatorname{Deg}(D)|$. The degrees that are *not* in D are precisely those that have at most one zero in each column of their block matrix representation. Hence, a simple counting argument shows that

$$\dim(\mathcal{T}(D)) = N - \left(1 + \frac{\log N}{\log d - 1}\right)^{(\log d - 1)}.$$

Remark 74. *The above code $\mathcal{T}(D)$ turns out to be the univariate analogue of the multivariate linear locality LTC presented in [GKS13]. The criterion for the degree set in the multivariate code is virtually the same "two zeros in some column" criterion here, except that the degrees for the multivariate code are m-tuples, and each component of the m-tuple corresponds to a row (viewed as a binary representation) of our block matrix representation. Testability of our univariate analogue follows from [GKS13], with the use of an isomorphism between $\mathbb{F}_{2^n}$ and $\mathbb{F}_{2^\ell}^m$.*

Remark 75. *The linear locality code of [HRZS13] is a code $\mathcal{C} \subseteq \{\mathbb{F}_{2^t}^{n/t} \to \mathbb{F}_2\}$ for general t dividing n. It is a generalization of the code in [GKS13] (the latter follows by setting $t = \ell$ for n that are multiples of ℓ). The procedure of this section can be applied in a* *similar fashion to obtain univariate analogues of the codes of [HRZS13], except that one uses subspaces constructed using the subfield $\mathbb{F}_{2^t}$ instead of $\mathbb{F}_{2^\ell}$, and the block matrix representation will have to be defined as an $(n/t) \times t$ matrix. We omit the details, since the technique is similar enough, and the specific construction of [GKS13] is the one that matches the lower bound on co-dimension given by Theorem 65.*

6.6.2 Optimality Results

Now, we show that the technical results of Theorems 68 and 69 are tight by showing that the univariate construction of the previous section matches those bounds.

Again, take $\ell = \log d - 1$ and $n = \ell m$, and let D be as in (6.9). Moreover, choose S to be a subspace whose indicator lies in the dual of $\mathcal{T}(D)$. Let $e^* = 2^0 + 2^1 + \cdots + 2^{n-\ell-1}$.

Lemma 36. *For any $0 \leq r \leq n - \ell - 1$, there exists at most one value of $u \in \{n - \ell, n - \ell + 1, \ldots, n - 1\}$ such that $e' = e^* - 2^r + 2^u$ does not pass S.*

Proof. Let $s = r \bmod \ell$. Note that for any $u \in \{n - \ell, \ldots, n - 1\}$ such that $u \neq n - \ell + s$, the block matrix representation of $e' = e^* - 2^r + 2^u$ contains two zeros in some column, and thus, e' would pass S. This implies that the only admissible value of u for which $e' = e^* - 2^r + 2^u$ does not pass S is $u = n - \ell + s$, as desired. □

From the proof of Theorem 66, we know that e^* does not pass S. Therefore, the result of Theorem 68 shows that there must exist *at least* one value of $e' = e - 2^r + 2^u$ that does not pass S. Thus, Lemma 36 matches this lower bound.

Next, we note the following lemma.

Lemma 37. *For any $n - \ell \leq u \leq n - 1$, there exist at most $m - 1 = \frac{n}{\log d - 1} - 1$ values of $r < n - \ell$ such that $e' = e^* + 2^u - 2^r$ does not pass S.*

Proof. Let $s = u \bmod \ell$. Then, note that for any $r < n - \ell$ such that $r \bmod \ell \neq s$, the block matrix representation of $e' = e^* + 2^u - 2^r$ contains two zeros in some column, and hence, e' would pass S. Thus, the only possible values of $r < n - \ell$ for which $e' = e^* + 2^u - 2^r$ may not pass S are those for which $r \bmod \ell = s$. There are precisely $m - 1$ such values of r, which proves the desired claim. □

Since the result of Theorem 69 shows that there must exist *at least* $\frac{n}{\log d - 1} - 1$ values of $e' = e + 2^u - 2^r$ that do not pass S, we see that Lemma 37 matches this lower bound.

Remark 76. *In a straighforward manner, one can show that Lemmas 36 and 37 still hold for any e^* whose block matrix representation has exactly one zero in each column. Such generalizations actually imply that $\mathcal{T}(D)$ is* maximal *among affine-invariant codes with the desired properties, i.e., that there is no non-trivial affine-invariant LTC whose degree set D' is a strict superset of D.*

Chapter 7

Conclusion

In this thesis, we have examined questions relating to capacity and limitations of error-correcting codes. The questions we tackle are fundamental in nature and, in general, relate to the central question of determining and achieving the optimal tradeoff between error tolerance and redundancy that has occupied the minds of coding theorists for decades. In answering such questions, we have considered a number of different settings for coding schemes, including one-way communication, interactive communication, list decoding, and local testing. Moreover, although error-correcting codes were initially developed for the purpose of reliable communication over noisy channels, we have not only focused on addressing this original goal but have also highlighted applications and connections to exciting new areas such as compressed sensing, approximation, communication complexity, etc.

We summarize the main contributions of this thesis and highlight some important open questions and future directions for research below.

7.1 Polar Codes

As discussed in Chapter 3, we have shown that polar codes are the first-known construction of explicit error-correcting codes that are efficiently encodable/decodable and provide a polynomial speed of convergence to capacity over *all* symmetric channels with input symbols from an alphabet of prime size. Moreover, for general (possibly non-symmetric) channels, we have shown that polar codes provide a polynomial speed of convergence to the symmetric capacity. Furthermore, we have shown how to extend the construction to alphabets of arbitrary size.

However, in the construction of polar codes, the dependence of the block length on N on the gap to capacity ϵ depends poorly on the arity q. In particular, a polynomial speed of convergence implies that there exists some constant $c > 0$ (possibly depending on q), often referred to as the *scaling exponent*, such that it suffices to take $N = \Omega((1/\epsilon)^c)$ in order to achieve ϵ gap to capacity. Our result has shown that $c = \mathrm{poly}(q)$ suffices. Furthermore, for $q = 2$ (the binary case), one can obtain $3.5 < c < 4$ [HAU13]. On the other hand, for a random code, it suffices to take $c = 2$. Thus, one possible objective for future work is to close this gap:

Objective 77. *Determine the optimal scaling exponent c as a function of q for polar codes over an alphabet of size q.*

As it turns out, c cannot be improved to 2 for the standard *binary* polar code construction discussed in Chapter 3, as the work of [HAU13] shows that one must necessarily take $c \geq 3.579$. However, we do not know the limit of the optimal c as the arity q of the polar code increases. One possible route to improving c beyond $\mathrm{poly}(q)$ is to improve the entropy sumset inequality that is used in the proof technique. It may be possible to reduce the exponent from $c = \mathrm{poly}(q)$ to $c = \mathrm{poly}(\log q)$ via such an approach by proving Theorem 7 with $\alpha(q) = 1/\mathrm{poly}(\log q)$ (recall that we obtain $\alpha(q) = 1/\mathrm{poly}(q)$; see Remark 16). This leads to the following concrete question:

Question 78. *Can the constant $\alpha(q)$ in the underlying entropy sumset inequality of Theorem 7 be improved?*

While the above question is of interest due to its connection to the convergence properties of polar codes, it is also of independent interest as a fundamental question in pseudorandomness, especially in light of similar sumset inequality counterparts in additive combinatorics.

Furthermore, it is possible that the scaling exponent c can be improved by changing the construction of polar codes. This can be done, for example, by concatenation with other codes or by using different polarizing kernels. Another possibility is to find a better decoder.

The approach to tighten the gap in the scaling exponent of polar codes and random codes by considering different polarizing *kernels* is especially intriguing. The *kernel* refers to the basic polarizing transform that is used in the construction of the codes. The standard kernel that has been used by [Arı09] and much of the polar codes literature, as well as Chapter 3, corresponds to the 2×2 matrix $K = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$. However, it is possible that we can use a different kernel, and, more specifically, it may be possible to achieve scaling exponents that approach 2 for kernels corresponding to $\ell \times \ell$ matrices for large ℓ. Moreover, it is known that in the limit $\ell \to \infty$, one can achieve a scaling exponent $c \approx 2$ because of the known behavior of random linear codes. An $\ell \times \ell$ kernel corresponds to using ℓ copies of a channel (instead of just two copies) in each recursive step of the polarization. Although using an $\ell \times \ell$ kernel results in an increased decoding complexity of $O(2^\ell N \log N)$ (instead of the $O(N \log N)$ decoding complexity under Arıkan's standard 2×2 kernel), it would nevertheless be interesting to investigate how increasing ℓ impacts the scaling exponent c:

Question 79. *Can polar codes based on larger $\ell \times \ell$ polarizing kernels (for $\ell > 2$) achieve a tighter polynomial dependence of block length on inverse gap to capacity with scaling exponent $c \approx 2$?*

Of course, the ultimate goal of the above questions is to construct explicit codes that exhibit a speed of convergence to capacity that matches that of random codes:

Objective 80. *Establish explicit capacity-achieving error-correcting codes whose speed of convergence to capacity matches the guarantees obtained by random codes, i.e., codes that exhibit polynomial convergence with scaling exponent $c \approx 2$.*

7.2 Interactive Communication

In Chapter 4, we have addressed the question of coding for interactive communication, in which two parties communicate back and forth with messages that can depend on the communication thus far. In particular, we have focused on finding interactive coding schemes for low error fractions $\epsilon > 0$ and have showed that under some modest assumptions about the protocol to be encoded, one can achieve a communication rate of $1 - \Theta(H(\epsilon))$ over random and oblivious adversarial channels, which matches (up to the constant factors on the $H(\epsilon)$) the capacity for one-way communication. Furthermore, our interactive coding scheme seems reasonably practical and have the added flexibility of being able to adapt to a rateless setting in which the error fraction is not known a priori. We also incorporate coding theoretic techniques (e.g., rateless codes) that were not used in prior interactive coding literature.

The most logical extension to our work would be to determine whether a similar result to Theorems 22 and 23 can be obtained for *fully* adversarial channels, in which the corruption patterns are allowed to depend adaptively on the communication transcript as the protocol proceeds:

Question 81. *Is there a reasonable set of assumptions on a two-party interactive protocol that would allow it to be encoded into a longer protocol that is resilient to an ϵ fraction of fully adversarial errors with a communication rate of* $1 - \Theta(H(\epsilon)) = 1 - \Theta(\epsilon \log(1/\epsilon))$*?*

Of course, we have still not entirely ruled out the possibility of achieving such communication rates for general interactive protocols. Recall that the work of [Hae14] showed that allowing *adaptivity* in the output protocol (i.e., a non-fixed speaking order) can support communication rates that surpass the bound of [KR13]. Specifically, Haeupler showed that for random errors or *oblivious* adversarial errors, there is a randomized coding scheme that allows one to achieve an error rate of $1 - O(\sqrt{\epsilon})$. Furthermore, in the case of full adversarial errors, he showed that a capacity of $1 - O(\sqrt{\epsilon \log\log(1/\epsilon)})$ is achievable. Although the results described in Chapter 4 allow us to surpass these error rates for the case of random and oblivious adversarial channels (see Theorems 22 and 23), they do have underlying assumptions about the average message length of the protocol that is being simulated. However, determining the optimality of the communication rates $1 - O(\sqrt{\epsilon})$ and $1 - O(\sqrt{\epsilon \log\log(1/\epsilon)})$ in their respective settings remains an important open question:

Question 82. *Is it possible to show the optimality of the best-known communication rates of* $1 - \sqrt{\epsilon}$ *and* $1 - \sqrt{\epsilon \log\log(1/\epsilon)}$ *for random errors and adversarial errors with low error fraction* $\epsilon > 0$*?*

Haeupler [Hae14] conjectures that this should be the case. One potential approach to resolving the question would be to adapt the lower-bound techniques from [KR13].

There also remain some open questions regarding the tolerable error fractions for interactive coding schemes. One question asked in [BR14] that remains open is the following fundamental question involving the tolerable error fraction for *binary* interactive coding schemes:

Question 83. *What is the maximum adversarial error fraction* ϵ *that can be tolerated by a* binary *coding scheme that encodes an arbitrary two-way protocol?*

While [BR14] showed that any adversarial error fraction $\epsilon < 1/4$ can be tolerated by encoding a given protocol into a longer protocol, approaching $1/4$ arbitrarily closely requires use of symbols from an alphabet that grows. If one restricts to coding schemes in which the output is a *binary* protocol, then the coding scheme of [BR14] only allows one to tolerate error rates up to $1/8$, and indeed, this is the best known bound so far.

On the other hand, it is known that a binary coding scheme cannot tolerate an error fraction of $1/6$ or more. This bound follows from an impossibility result for the problem of *communication with noiseless feedback* [Ber64, EGH15]. In this setup, the Alice and Bob communicate over a noisy channel, but the sender has access to an uncorrupted *feedback channel*. Essentially, one can show an upper limit of $1/3$ for the fraction of errors that one can tolerate in an interactive coding scheme over channels with feedback, and this translates to the $1/6$ bound for Question 83. It should be noted that the problem of communication with noiseless feedback is also related to the classic game of "Twenty Questions with a Liar" [SW92]. Closing the gap between $1/8$ and $1/6$ for Question 83 would be very interesting.

Finally, there have been several recent papers on interactive coding under various other models (e.g., multiparty communication, insertion/deletion errors), and one can ask questions about the tolerable error fractions and capacities for interactive coding under such models [BGMO16, ABE$^+$16, BEGH16, EGH16, GH15].

7.3 List Decoding and Compressed Sensing

In Chapter 5, we have considered the setting of list decoding, in which one relaxes the requirement that a decoder output a single message and allows the decoder to output a short list of possible messages that were intended by the sender. One of the important themes in coding theory is the investigation how *random* coding-theoretic objects behave, as they provide a reasonable target for what parameters and tradeoffs are achievable. As a concrete example, Shannon's noisy channel coding theorem provides an existential result and shows that random codes can achieve the channel capacity of a DMC with a block length that scales quadratically in the inverse gap to capacity. Although the codes achieved by the theorem are not explicit and, therefore, not practical, the result nevertheless provides a baseline for the search for explicit capacity-achieving codes (see the discussion in Section 7.1).

In the realm of list decoding, the known performance of random codes was largely incomplete prior to our work in Chapter 5. In particular, the optimal tradeoffs between the list decoding radius, rate, and list size were note known for random *linear* codes as the list decoding radius approaches $1 - 1/q$ (where the alphabet size is q). Since many well-known and widely-used error-correcting codes happen to be linear, determining the optimal tradeoff between various parameters for random linear codes is a fundamental question in coding theory. Further motivation for analyzing list decodability in this regime is provided by connections to a number of other topics in theoretical computer science, such as pseudorandom generators, randomness extractors, etc. In this thesis, we have essentially closed the gap between random linear codes and random codes. In particular, we have shown that a random q-ary linear code of rate $\Omega_q(\epsilon^2/\log^3(1/\epsilon))$ is list decodable up to a radius of $1-1/q-\epsilon$ with list size $O(1/\epsilon^2)$. As mentioned earlier, Wootters [Woo13] removed the suboptimal logarithmic factors in $1/\epsilon$ that appear in the rate. However, the dependence on q in the rate still appears. It is highly believed that the dependence on q should be removable, and this forms the basis of an interesting open question:

Question 84. *Is it possible to show that a random q-ary linear code of rate $\Omega(\epsilon^2)$ (independent of q) is $(1-1/q-\epsilon, O(1/\epsilon^2))$-list decodable?*

Answering this question in the affirmative would essentially resolve the question about the optimal tradeoffs for random linear codes in the aforementioned regime.

Another major contribution of this thesis is to establish a connection between list decoding and compressed sensing. Compressed sensing has been an emerging field that has attracted much attention in electrical engineering, computer science, and applied mathematics. We have established that the well-known restricted isometry property for subsampled Fourier matrices implies list decodability of random linear codes in our regime. In the process, we have improved important results of [CT06, RV08] on the number of Fourier samples needed to enable recovery of sparse signals. Although the connection of the compressed sensing result to list decoding is important, the question about the optimal number of samples for compressed sensing is also an interesting question in its own right, and the best known lower bound for the number of row samples needed in an $N \times N$ Fourier matrix in order to satisfy the restricted isometry property (RIP-2) of order k (with a fixed constant δ) is $\Omega(k \log N)$ [BLM15]. This suggests the following important question in compressed sensing:

Question 85. *Let $\delta > 0$ be a sufficiently small fixed constant. What is the minimum number of random row samples m needed for a normalized $N \times N$ Fourier matrix M with entries of absolute value $O(1/\sqrt{N})$ such that the resulting subsampled matrix (with m rows chosen uniformly and independently from the rows of M) satisfies RIP-2 of order k with constant δ? In particular, can one take $m = O(k \log N)$?*

7.4 Local Testability

Finally, in Chapter 6 of this thesis, we have explored the property of local testability in error-correcting codes. Recall that local testability gurarantees the property that it is possible to distinguish codewords from words that are far in Hamming distance from the code with nontrivial probability by querying a received word in just a few carefully chosen positions. As discussed, locally testable codes have been showed to have applications to probabilistically checkable proofs and property testing, especially in the case in which the query complexity is sub-linear or constant.

In this thesis, we have examined locally testable codes in a different regime, namely, those codes in which the query complexity is linear. The specific application in mind for this regime is the theory of approximation algorithms, where locally testable codes can be used to construct small set expander graphs. Under certain assumptions (e.g., affine invariance, containment of a sufficiently large Reed-Muller code), we have shown that among locally testable codes of block length N, distance d, and appropriate query complexity, the lifted codes of [GKS13] are essentially optimal. Another consequence of our work in Chapter 6 is the resolution of the limitation posed by the local testability requirement in the spectrum of codes from Reed-Muller to BCH (for fixed N and d).

However, the nature of the assumptions that we have made in order to prove the upper bound on code dimension in Theorem 65 leaves open the possibility for other locally testable codes with higher dimension that do not satisfy these assumptions. Thus, the main open question is to determine whether we can still establish upper bounds on the dimension with fewer assumptions about the code:

Question 86. *Can the assumptions on $\mathcal{C}$ be relaxed in the statement of Theorem 65? In particular, can we:*

1. *Remove the assumption about containing* $\mathrm{RM}(n - \log d, n)$*?*
2. *Show similar bounds for codes that are testable with* $O(N/d)$ *queries instead of specifically* $2N/d$ *queries?*
3. *Show similar bounds for codes that are not affine-invariant?*

If any of the individual questions in the bulleted list in Question 86 cannot be answered in the affirmative, it could mean the existence of locally testable codes that surpass the dimension of the lifted codes of Guo, et al. [GKS13]. This would imply interesting algorithmic results relating to the aforementioned small set expander problem. However, addressing Question 86 will require new techniques beyond the ones we use to prove Theorem 65.

Bibliography

[ABE$^+$16] Noga Alon, Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Reliable communication over highly connected noisy networks. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*, pages 165–173, 2016. 7.2

[AKK$^+$05] Noga Alon, Tali Kaufman, Michael Krivelevich, Simon Litsyn, and Dana Ron. Testing Reed-Muller codes. *IEEE Transactions on Information Theory*, 51(11):4032–4039, 2005. 6.3

[AL13] Nir Ailon and Edo Liberty. An almost optimal unrestricted fast johnson-lindenstrauss transform. *ACM Trans. Algorithms*, 9(3):21, 2013. 5.1.2

[ALM15] Emmanuel Abbe, Jiange Li, and Mokshay M. Madiman. Entropies of weighted sums in cyclic groups and applications to polar codes. *CoRR*, abs/1512.00135, 2015. 3.5

[Arı09] Erdal Arıkan. Channel polarization: a method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, 2009. 1, 1.2, 3.1, 7.1

[Arı10] Erdal Arıkan. Source polarization. In *Proceedings of 2010 IEEE International Symposium on Information Theory*, pages 899–903, 2010. 3.1, 3.2.4, 3.7

[AT09] Erdal Arıkan and Emre Telatar. On the rate of channel polarization. In *Proceedings of 2009 IEEE International Symposium on Information Theory*, pages 1493–1495, 2009. 3.7

[AT12] Emmanuel Abbe and Emre Telatar. Polar codes for the m–user multiple access channel. *IEEE Transactions on Information Theory*, 58(8):5437–5448, 2012. 3.1

[BDDW08] Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, Dec. 2008. 5.1.2

[BE14] Mark Braverman and Klim Efremenko. List and unique coding for interactive communication in the presence of adversarial noise. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 236–245, 2014. 4.1.1

[BEGH16] Mark Braverman, Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Constant-rate coding for multiparty interactive communication is impossible. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 999–1010, 2016. 7.2

[Ber64] Elwyn R. Berlekamp. *Block Coding with Noiseless Feedback*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1964. 7.2

[BFH$^+$13] Arnab Bhattacharyya, Eldar Fischer, Hamed Hatami, Pooya Hatami, and Shachar Lovett. Every locally characterized affine-invariant property is testable. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 429–436, 2013. 2

[BGH$^+$12] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 370–379, 2012. 6.1, 6.1, 6.1

[BGMO16] Mark Braverman, Ran Gelles, Jieming Mao, and Rafail Ostrovsky. Coding for Interactive Communication Correcting Insertions and Deletions. In Yuval Rabani Ioannis Chatzigiannakis, Michael Mitzenmacher and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 61:1–61:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 7.2

[BK12] Zvika Brakerski and Yael Tauman Kalai. Efficient interactive coding against adversarial noise. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 160–166, 2012. 4.1.1

[BKN14] Zvika Brakerski, Yael Tauman Kalai, and Moni Naor. Fast interactive coding against adversarial noise. *J. ACM*, 61(6):35, 2014. 4.1.1

[BKS$^+$10] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of Reed-Muller codes. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 488–497, 2010. 6.1, 54, 6.3, 60

[Bli86] Volodia M. Blinovsky. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22(1):7–19, 1986. 5.1, 5.1.1

[Bli08] Volodia M. Blinovsky. On the convexity of one coding-theory function. *Problems of Information Transmission*, 44(1):34–39, 2008. 5.1

[BLM15] Afonso S. Bandeira, Megan E. Lewis, and Dustin G. Mixon. Discrete uncertainty principles and sparse signal processing. *CoRR*, abs/1504.01014, 2015. 5.1.2, 7.3

[BN13] Zvika Brakerski and Moni Naor. Fast algorithms for interactive coding. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 443–456, 2013. 4.1.1

[Bou14] Jean Bourgain. *An Improved Estimate in the Restricted Isometry Problem*, pages 65–70. Springer International Publishing, Cham, 2014. 37

[BR14] Mark Braverman and Anup Rao. Toward coding for maximum errors in interactive communication. *IEEE Transactions on Information Theory*, 60(11):7248–7255, 2014. 4.1.1, 4.2, 4.6, 7.2, 7.2

[BSGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput.*, 36(4):889–974, 2006. 6.1

[BSHR05] Eli Ben-Sasson, Prahladh Harsha, and Sofya Raskhodnikova. Some 3CNF properties are hard to test. *SIAM Journal on Computing*, 35(1):1–21, September 2005. 1, 6.2.1, 6.2.1

[BSS06] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Structures and Algorithms*, 28(4):387–402, 2006. 6.1

[BSS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM J. Comput.*, 38(2):551–607, 2008. 6.1

[BSS11] Eli Ben-Sasson and Madhu Sudan. Limits on the rate of locally testable affine-invariant codes. In *APPROX-RANDOM*, pages 412–423, 2011. 6.1, 6.2.2

[Can08] Emmanuel Candès. The restricted isometry property and its implications for compresses sensing. *C. R. Math. Acad. Sci. Paris*, 346:589–592, 2008. 5.1.2

[Car85] Bernd Carl. Inequalities of Bernstein-Jackson-type and the degree of compactness of operators in Banach spaces. *Annales de l'institut Fourier*, 35(3):79–118, 1985. 5.4

[CGV13] Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of fourier matrices and list decodability of random linear codes. *SIAM J. Comput.*, 42(5):1888–1914, 2013. 1.2, 5

[Che10] Mahdi Cheraghchi. *Applications of Derandomization Theory in Coding*. PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, 2010. (available online at `http://eccc.hpi-web.de/static/books/Applications_of_Derandomization_Theory_in_Coding/`). 5.1

[Che11] Mahdi Cheraghchi. Coding-theoretic methods for sparse recovery. In *Proceedings of the Annual Allerton Conference on Communication, Control, and Computing*, 2011. 5.1.2

[CRT06a] Emmanuel Candès, Justin Romberg, and Terence Tao. Robust uncertainty principle: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. on Inf. Th.*, 52:489–509, 2006. 5.1.2

[CRT06b] Emmanuel Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59:1208–1223, 2006. 5.1.2

[Ş10] Eren Şaşoğlu. An entropy inequality for q-ary random variables and its application to channel polarization. In *ISIT*, pages 1360–1363. IEEE, 2010. 3.4

[Ş12] Eren Şaşoğlu. Polarization and polar codes. *Foundations and Trends in Communications and Information Theory*, 8(4):259–381, 2012. 3.1, 3.2.5, 3.2.6, 3.2.7, 3.2.7, 3.2.7, 3.3, 3.4, 3.4, 3.5.1, 3.8

[CT06] Emmanuel Candès and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52:5406–5425, 2006. 3, 5.1.2, 7.3

[cTA09] Eren Şaşoğlu, Emre Telatar, and Erdal Arıkan. Polarization for arbitrary discrete memoryless channels. *CoRR*, abs/0908.0302, 2009. 3.1, 3.2.5, 3.4, 3.8

[cTY13] Eren Şaşoğlu, Emre Telatar, and Edmund M. Yeh. Polar codes for the two-user multiple-access channel. *IEEE Transactions on Information Theory*, 59(10):6583–6592, 2013. 3.1

[DG13] Irit Dinur and Venkatesan Guruswami. PCPs via low-degree long code and hardness for constrained hypergraph coloring. In *Proceedings of the 54th Annual Symposium on Foundations of Computer Science*, FOCS, pages 340–349, 2013. 6.1

[Din07] Irit Dinur. The PCP theorem by gap amplification. *J. ACM*, 54(3):12, 2007. 6.1

[Don06] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52:1289–1306, 2006. 5.1.2

[EGH15] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science, ITCS 2015, Rehovot, Israel, January 11-13, 2015*, pages 11–20, 2015. 4.1.1, 7.2

[EGH16] Klim Efremenko, Ran Gelles, and Bernhard Haeupler. Maximal noise in interactive communication over erasure channels and channels with feedback. *IEEE Trans. Information Theory*, 62(8):4575–4588, 2016. 7.2

[Eli91] Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37:5–12, 1991. 5.1, 5.1.2

[FGOS15] Matthew K. Franklin, Ran Gelles, Rafail Ostrovsky, and Leonard J. Schulman. Optimal coding for streaming authentication and interactive communication. *IEEE Transactions on Information Theory*, 61(1):133–145, 2015. 4.1.1

[GAG13] Naveen Goela, Emmanuel Abbe, and Michael Gastpar. Polar codes for broadcast channels. In *Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, July 7-12, 2013*, pages 1127–1131, 2013. 3.1

[GGR11] Parikshit Gopalan, Venkatesan Guruswami, and Prasad Raghavendra. List decoding tensor products and interleaved codes. *SIAM J. Comput.*, 40(5):1432–1462, 2011. 5.1.2, 5.2.3

[GH14] Mohsen Ghaffari and Bernhard Haeupler. Optimal error rates for interactive coding II: efficiency and list decoding. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 394–403, 2014. 4.1.1, 4.2, 4.6

[GH15] Ran Gelles and Bernhard Haeupler. Capacity of interactive communication over erasure channels and channels with feedback. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1296–1311, 2015. 7.2

[GHH$^+$14] Venkatesan Guruswami, Johan Håstad, Prahladh Harsha, Srikanth Srinivasan, and Girish Varma. Super-polylogarithmic hypergraph coloring hardness via low-degree long codes. In *Proceedings of the 46th annual ACM Symposium on Theory of Computing*, STOC, 2014. 6.1

[GHK11] Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. *IEEE Transactions on Information Theory*, 57(2):718–725, 2011. Special issue dedicated to the scientific legacy of Ralf Koetter. 5.1.1, 5.1.2

[GHS14] Mohsen Ghaffari, Bernhard Haeupler, and Madhu Sudan. Optimal error rates for interactive coding I: adaptivity and other settings. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 794–803, 2014. 4.1.1, 4.2, 4.3.1, 4.6

[GHSZ02] Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1035, 2002. 5.1.1, 5.1.2

[GKS13] Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In *Proceedings of ITCS 2013*, pages 529–540, 2013. 6.1, 6.1, 6.1, 6.3.1, 70, 6.5, 6.6, 6.6, 74, 75, 7.4, 7.4

[GKZ08] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. List-decoding reed-muller codes over small fields. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 265–274, 2008. 5.1.2, 5.2.3

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989. 5.1

[GMS14] Ran Gelles, Ankur Moitra, and Amit Sahai. Efficient coding for interactive communication. *IEEE Transactions on Information Theory*, 60(3):1899–1913, 2014. 4.1.1

[GN12] Venkatesan Guruswami and S. Narayanan. Combinatorial limitations of a strong form of list decoding. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:17, 2012. 5.1.1

[Gol11] Oded Goldreich. Short locally testable codes and proofs: a survey in two parts. In *Property testing*, pages 65–104. Springer, 2011. 6.1

[GS01] Venkatesan Guruswami and Madhu Sudan. Extensions to the Johnson bound. *Unpublished manuscript*, 2001. Available at `http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.145.9405`. 5.1.1, 5.2

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM*, 53(4):558–655, 2006. 6.1

[GS10] Venkatesan Guruswami and Adam Smith. Codes for computationally simple channels: Explicit constructions with optimal rate. In *Proceedings of IEEE Symposium on the Foundations of Computer Science*, 2010. 5.1

[GSVW15] Venkatesan Guruswami, Madhu Sudan, Ameya Velingker, and Carol Wang. Limitations on testable affine-invariant codes in the high-rate regime. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete*

Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015, pages 1312–1325, 2015. 1.2, 6

[GV10] Venkatesan Guruswami and Salil Vadhan. A lower bound on list size for list decoding. *IEEE Transactions on Information Theory*, 56(11):5681–5688, 2010. 5.1

[GV15] Venkatesan Guruswami and Ameya Velingker. An entropy sumset inequality and polynomially fast convergence to shannon capacity over all alphabets. In *30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA*, pages 42–57, 2015. 1.2, 3

[GX13] Venkatesan Guruswami and Patrick Xia. Polar codes: Speed of polarization and polynomial gap to capacity. In *FOCS*, pages 310–319, 2013. Full version to appear in *IEEE Trans. on Info. Theory*, Jan. 2015. 3.1, 3.4, 3.4, 3.6, 3.7, 3.7

[Hae14] Bernhard Haeupler. Interactive channel capacity revisited. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 226–235, 2014. 4.1.1, 4.1.2, 4.2, 4.2, 4.2, 4.5, 24, 4.5.2, 4.6, 4.9, 7.2, 7.2

[Ham50] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 26(2):147–160, 1950. 1.1.2

[HAT14] Saeid Haghighatshoar, Emmanuel Abbe, and I. Emre Telatar. A new entropy power inequality for integer-valued random variables. *IEEE Trans. Information Theory*, 60(7):3787–3796, 2014. 3.5, 4, 16

[HAU13] Seyed Hamed Hassani, Kasra Alishahi, and Rüdiger L. Urbanke. Finite-length scaling of polar codes. *CoRR*, abs/1304.4778, 2013. 3.1, 1, 7.1, 7.1

[HR16] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled fourier matrices. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*, pages 288–297, 2016. 37

[HRZS13] Elad Haramaty, Noga Ron-Zewi, and Madhu Sudan. Absolutely sound testing of lifted codes. In *Proceedings of APPROX-RANDOM 2013*, pages 671–682, 2013. 6.1, 6.1, 6.5, 75

[Huf52] David Huffman. A method for the construction of minimum redundancy codes. 40(9):1098–1101, 1952. 2.4

[HV16] Bernhard Haeupler and Ameya Velingker. Bridging the capacity gap between interactive and one-way communication. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:90, 2016. 1.2, 4

[JA14] Varun Jog and Venkat Anantharam. The entropy power inequality and mrs. gerber's lemma for groups of order 2^n. *IEEE Transactions on Information Theory*, 60(7):3773–3786, 2014. 3.4

[KcU10] Satish Babu Korada, Eren Şaşoğlu, and Rüdiger L. Urbanke. Polar codes: Characterization of exponent, bounds, and constructions. *IEEE Transactions on Information Theory*, 56(12):6253–6264, 2010. 3

[KL11] Tali Kaufman and Shachar Lovett. New extension of the Weil bound for character sums with applications to coding. In *52nd Annual IEEE Symposium on Foundations of Computer Science*, FOCS, pages 788–796, 2011. 6.3.2, 61, 62, 6.3.2, 64, 6.4

[Kor10] Satish Babu Korada. Polar codes for Slepian-Wolf, Wyner-Ziv, and Gelfand-Pinsker. In *Proceedings of the 2010 IEEE Information Theory Workshop*, pages 1–5, 2010. 3.1

[KR13] Gillat Kol and Ran Raz. Interactive channel capacity. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 715–724, 2013. 4.1.1, 4.1.2, 4.6, 7.2, 7.2

[Kra49] L. G. Kraft. A device for quantizing, grouping, and coding amplitude modulated pulses. Master's thesis, Department of Electrical Engineering, MIT, Cambridge, MA, USA, 1949. 4

[KS99] S. R. Kumar and D. Sivakumar. Proofs, codes, and polynomial-time reducibilities. In *Proceedings of the 14th Annual IEEE Conference on Computation Complexity*, 1999. 5.1

[KS07a] Tali Kaufman and Madhu Sudan. Algebraic property testing: The role of invariance. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(111), 2007. 6.1

[KS07b] Tali Kaufman and Madhu Sudan. Algebraic property testing: The role of invariance. Technical Report TR07-111, Electronic Colloquium on Computational Complexity, 2 November 2007. Extended abstract in *Proc. 40th STOC*, 2008. 6.2.2

[KS14] Subhash Khot and Rishi Saket. Hardness of coloring 2-colorable 12-uniform hypergraphs with $2^{(\log n)^{\Omega(1)}}$ colors. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:51, 2014. 6.1

[KU10] Satish Babu Korada and Rüdiger L. Urbanke. Polar codes are optimal for lossy source coding. *IEEE Transactions on Information Theory*, 56(4):1751–1768, 2010. 3.1

[KW11] Felix Krahmer and Rachel Ward. New and improved johnson-lindenstrauss embeddings via the restricted isometry property. *SIAM J. Math. Analysis*, 43(3):1269–1281, 2011. 5.1.2

[LA14] Jingbo Liu and Emmanuel Abbe. Polynomial complexity of polar codes for non-binary alphabets, key agreement and slepian-wolf coding. In *48th Annual Conference on Information Sciences and Systems, CISS 2014, Princeton, NJ, USA, March 19-21, 2014*, pages 1–6, 2014. Available online at `http://arxiv.org/abs/1405.0776`. 3.8

[LT91] Michel Ledoux and Michel Talagrand. *Probability in Banach spaces*. Springer Verlag, 1991. 5.1.2, 5.4, 5.4

[McM56] Brockway McMillan. Two inequalities implied by unique decipherability. 2(4):115–116, December 1956. 4

[Moo96] Eliakim Hastings Moore. A two-fold generalization of Fermat's theorem. *Bull. Am. Math. Soc.*, 2(7):189–199, 1896. MR:1557441. JFM:27.0139.05. 6.4.2

[MS81] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Codes*. Elsevier/North-Holland, Amsterdam, 1981. 6.3

[MU01] Elchanan Mossel and Christopher Umans. On the complexity of approximating the VC dimension. *Journal of Computer and System Sciences*, 65(4):660–671, 2001. 5.1

[MV11] Hessam Mahdavifar and Alexander Vardy. Achieving the secrecy capacity of wiretap channels using polar codes. *IEEE Transactions on Information Theory*, 57(10):6428–6443, 2011. 3.1

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. 4.8, 6

[PHE98] Vera S. Pless and William C. Huffman (Eds.). *Handbook of Coding Theory (2 Volumes)*. Elsevier, 1998. 6.4

[RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC, pages 755–764, 2010. 6.1

[Rud11] Atri Rudra. Limits to list decoding of random codes. *IEEE Transactions on Information Theory*, 57(3):1398–1408, 2011. 5.1.1

[RV08] Mark Rudelson and Roman Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61:1025–1045, 2008. 3, 5.1.2, 3, 37, 5.4, 5.4, 5.4, 5.4, 5.4, 28, 5.4, 51, 5.4, 5.4, 5.4, 5.4, 7.3

[Sch92] Leonard J. Schulman. Communication on noisy channels: A coding theorem for computation. In *33rd Annual Symposium on Foundations of Computer Science, Pittsburgh, Pennsylvania, USA, 24-27 October 1992*, pages 724–733, 1992. 4.1, 4.1.1, 4.9

[Sch93] Leonard J. Schulman. Deterministic coding for interactive communication. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*, pages 747–756, 1993. 4.1, 4.1.1

[Sch96] Leonard J. Schulman. Coding for interactive communication. *IEEE Transactions on Information Theory*, 42(6):1745–1756, 1996. 4.1, 4.1.1, 4.2

[Sha48] C. E. Shannon. A mathematical theory of communication. *Bell system technical journal*, 27, 1948. 1.1.1, 2, 2.4, 5, 2

[Sha12] Shai Shalev-Shwartz. Online learning and online convex optimization. *Foundations and Trends in Machine Learning*, 4(2):107–194, 2012. 3.5.1

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and Systems Sciences*, 62(2):236–266, 2001. 5.1

[Sud11] Madhu Sudan. Guest column: Testing linear properties: Some general themes. *SIGACT News*, 42(1):59–80, March 2011. 6.1

[SW92] Joel Spencer and Peter Winkler. Three thresholds for a liar. *Combinatorics, Probability & Computing*, 1:81–93, 1992. 7.2

[Tao10] Terence Tao. Sumset and inverse sumset theory for Shannon entropy. *Combinatorics, Probability and Computing*, 19(4):603–639, 2010. 3.5, 16

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001. 5.1

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, pages 347–424, 2004. 6.1

[Ver12] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In *Compressed Sensing, Theory and Applications, ed. Y. Eldar and G. Kutyniok (Chapter 5), Cambridge University Press*, pages 210–268, 2012. 5.4

[Vid13] Michael Viderman. Strong LTCs with inverse poly-log rate and constant soundness. In *54th Annual Symposium on Foundations of Computer Science*, FOCS, pages 330–339, 2013. 6.1

[Wc14] Lele Wang and Eren Şaşoğlu. Polar coding for interference networks. *CoRR*, abs/1401.7293, 2014. 3.1

[Wit74] Hans S. Witsenhausen. Entropy inequalities for discrete channels. *IEEE Transactions on Information Theory*, 20(5):610–616, 1974. 3.4

[Woo13] Mary Wootters. On the list decodability of random linear codes with large error rate. *CoRR*, abs/1302.2261, 2013. 36, 37, 7.3

[WZ73] Aaron D. Wyner and Jacob Ziv. A theorem on the entropy of certain binary sequences and applications-I. *IEEE Transactions on Information Theory*, 19(6):769–772, 1973. 3.4

[ZP82] Victor V. Zyablov and Mark S. Pinsker. List cascade decoding. *Problems of Information Transmission*, 17(4):29–34, 1981 (in Russian); pp. 236-240 (in English), 1982. 5.1, 5.1.2