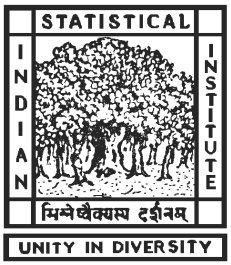
Quantum query complexity through the lens of communication complexity and exact learning



Manaswi Paraashar

Supervisor: Dr. Sourav Chakraborty

Advanced Computing and Microelectronics Unit
Indian Statistical Institute
203 B. T. Road, Kolkata-700108

A thesis submitted in partial fulfillment of the requirements for the degree of *Doctor of Philosophy* in *Computer Science* at *Indian Statistical Institute*

February 2022

Dedicated to my parents.

Acknowledgements

First and foremost, I would like to thank to my advisor, Sourav Chakraborty. Sourav introduced me to research and shaped my understanding of many aspects of the career I have chosen. He introduced me to the beautiful areas of Fourier analysis of Boolean functions and complexity theory. He introduced me to the right people at the right time which was crucial to my research in general and Ph.D. in particular. I had innumerable discussions with Sourav, which were often very lengthy, and he was always available for these discussions even in his most busy times. I am equally thankful to him for the great time we spent outside the research world, his efforts to instigate me with his love of traveling, and his patience towards me. I have been most fortunate to have Sourav as my advisor.

I am most grateful to Arijit Bishnu, Arijit Ghosh and Rajat Mittal. All three of them were always available to help me out. Arijit Bishnu, my masters' advisor, convinced me to pursue Ph.D. at ISI Kolkata, for which I am most thankful. I thank Arijit Ghosh and Rajat Mittal for answering my neverending questions and for always being available to discuss. Much of my understanding of my areas of research is due to them. I thank Arkadev Chattopadhyay for inviting me to TIFR, Mumbai, for many discussions that we had and for providing me with advice whenever I needed it during my Ph.D. I was very fortunate to interact with Nikhil Mande, Gopinath Mishra, and Nitin Saurabh during the initial years of my Ph.D. While Nikhil and Nitin were postdocs at that time, Gopinath was a senior Ph.D. student. I learned a lot from my interactions and discussions with them. I would like to thank Satya Lokam for inviting me to Microsoft Research, India. I would also like to thank Ronald de Wolf for several discussions and his advice during the later stages of my Ph.D. I thank all my collaborators, Srinivasan Arunachalam, Anup Bhattacharya, Arijit Bishnu, Sourav Chakraborty, Sameer Desai, Peter Høyer, Chandrima Kayal, Troy Lee, Nikhil Mande, Gopinath Mishra, Rajat Mittal, Tulasimohan Molli, Nikhil Shagrithaya, Swagato Sanyal, Sayantan Sen, and Ronald de Wolf, for many discussion that we had during the past few years.

I would like to thank my institute ISI Kolkata and my department (ACMU) for providing an excellent research environment. I would also like to thank the theoretical computer science community that is extremely generous in sharing knowledge and the online platforms that make this possible. I thank all my friends from 6th Floor ACMU, my institute, from Kolkata, and from my bachelors for the a great time during my Ph.D. days.

Finally, saving the best words for the last, I thank my parents and my wife, for their unending support, patience, and belief without which this thesis would not have been a possibility.

Manaswi Paraashar

Abstract

Query complexity, both classical and quantum, are important and well-established notions of computation. In this thesis, we will investigate query complexity in both classical and quantum worlds and study their relationship with other closely related models of computation.

Other than the query model, we will consider a few models of computation, namely model of computational learning, communications model and local query model for graphs. We will try to understand the power of these models for various classes of functions. We will be investigating how various complexity measures (defined as per different models of computation) relate to each other. In this endeavor we will be working both upper bound (i.e. designing efficient algorithms in the model of interest) and lower bound (i.e. proving hardness of computing interesting functions in the model of interest). As a part of our investigation, we use various mathematical tools like Fourier analysis, linear algebra, and geometry. Some questions that have motivated the work in this thesis are:

- How does structural simplicity affect its computational complexity in various models of computation? Does Fourier analytical simplicity lead to easier computation? Does geometric simplicity imply efficient communication?
- How does the behavior of quantum computing differ from its classical counterpart? What is the relation between classical and quantum query complexity of learning a function? Does the same relationship between a pair of classical complexity measures also hold in the quantum world?

Understanding the complexity measures that we study and the relationship between these measures has been an ongoing work for several decades. In this thesis, we push the boundary of our knowledge a bit further.

The results in this thesis have been divided into three parts.

Part I In this part, we study quantum query complexity from the view of quantum learning theory. First, we first study the model of exact learning using uniform quantum examples.

Each such quantum example can be generated by making one query to the function being learned. In this model we are interested in learning the class of Boolean functions whose Fourier sparsity is bounded, that is, the class of Fourier-sparse Boolean functions. For this class of functions we give a quantum learning algorithm which improves upon the tight classical algorithm by Haviv and Regev, 2016. Then, we consider the model of exact active learning. In this model, a learner is given access to the function via membership queries. We study the relationship between the number of classical and quantum membership queries needed to exactly learn a class of Boolean functions, where improve upon the previous best known relationship by Servedio and Gortler, 2004. Finally, we study Chang's lemma, a fundamental result in additive combinatorics. Our main result here is a refinement of Chang's lemma for Fourier-sparse Boolean functions. We also investigate how this lemma is connected to quantum learning theory.

Part II This part is dedicated to the study of the relation between quantum query and quantum communication complexity. It is well known, in the classical world, that a query algorithm for a function can be simulated to give a communication protocol of a closely related communication problem with only a constant overhead. The best known such simulation theorem in the quantum world, due to Buhrman, Cleve and Wigderson, 1998, has an overhead that is logarithmic in input size of the function. We construct the first total Boolean function that witnesses this logarithmic gap. This closes a long line of work. We also give a general recipe of constructing functions that witness separation between quantum query-to-communication simulation. Finally, we explore the role of symmetry on this simulation problem.

Part III This part of the thesis is devoted to classical query and communication complexity. In the first chapter of this part, we explore the role of geometric simplicity, quantified by bounded Vapnik–Chervonenkis Dimension (VC Dimension) of set systems, in communication complexity. Our work is motivated by the work of Håstad and Wigderson, 2007, who considered the Disjointness problem, a canonical problem in communication complexity, when inputs to the problem are promised to be sets of bounded cardinality. Indeed, set systems of bounded VC Dimension are a generalization of set systems with bounded cardinality, with a geometric motivation. Our main result here it to show that geometric simplicity does not imply efficient communication. In the second chapter of this part, we consider the problem of estimating the size of the global minimum cut in the local query model, a fundamental and well-studies model in graph property testing. We give an algorithm for this

problem which almost matches the known lower bound by Eden and Rosenbaum, 2018. This resolves the query complexity of estimating global minimum cut in the local query model.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Computational complexity theory deals with the following question: If $f : \mathsf{Domain} \to \mathsf{Range}$ is a function from Domain to Range, then

Given an $x \in \mathsf{Domain}$*, how "efficiently" can* $f(x)$ *be computed?*

While the description of the above question seems natural enough, there are many things that are not defined in its statement. For example:

- How is an algorithm (that would compute $f(x)$) allowed to access the input x?
- What are the resources available at the disposal of such an algorithm in order to evaluate the function?
- What does *efficient* mean?
- What is the correctness requirement? Do we allow our algorithm to err with a small probability?

A model of computation answers the above questions; it defines the rules of computation and the costs involved. But, what are some interesting and "natural" models of computation? How are different models of computations born? Which models of computation are worth studying?

A search for answers to these questions would possibly require one to explore the entire history of computing (the records of which date back to 19,000 BC [148]). It is apparent that at different points of time, different models of computation were used depending on the required application and available technology of that time. For example, the need to do basic arithmetic led to the development of *abacus* around 2500 BC ([148, 149]). While

landmark developments happened in the centuries that followed, the seemingly simple notion of computation was yet to be formally defined. We refer the interested readers to [148] for a timeline of the history of computation.

The second half of the 20th century saw extraordinary leaps in the theory and application of computer science that eventually changed the way we perceive computation today. One of the biggest milestone that was achieved during this period was the discovery of a fundamental model of computation: the Turing Machine, by Alan Turing in his landmark paper ([144]). With this model at hand, the notion of computation and computability became clear and, arguably, the field of modern theoretical computer science started. The research that followed this discovery led researchers to believe that Turing Machines capture the most general nature of *computability*. This is the Church-Turing Thesis which, at a high level, says that if a function can be computed by any reasonable model of computation, then it can also be computed in the Turing Machine model.

Further research led the researchers to further believe that not only Turing Machines capture computability but they also capture *efficient* computation. The belief that any function that can be efficiently computed in any reasonable model of computation can also be computed efficiently in the Turing Machine model is called the Extended Church-Turing Thesis. The study of “efficient" computing got a major boost in the early 1970s with the work of Stephen Cook [46] and Leonid Levin [107]. Computational Complexity Theory was now a major subject of research. The progress in technology was taking place hand-to-hand with theory in this period, perhaps at an even faster rate: this was truly an age of computer science. Large computers in labs during the mid 20th century gave way to personal computers (PCs) and small communication networks of the same time led to the birth of the internet.

By the dawn of the 21st century, computing devices became commonplace and many areas of computer science, like computer networking, cryptography, online algorithms, streaming algorithms to name a few, were born. Unprecedented capability to store and process data and interdisciplinary research between areas like computer science, optimization, statistics, and others, led to the birth of machine learning.

In the early 20th century, while computer science was undergoing a revolution, physics was also undergoing major breakthroughs that will later change the way we view nature. Of particular interest to us is the theory of quantum mechanics which eventually gave us *the model of quantum computation*, a model of computation that is central to this thesis. Yuri Manin [109, 110], Richard Feynman [59, 60] and Paul Benioff [22] were instrumental in formalizing the model of quantum computation. We refer to [85] for a history of quantum computing. It was observed that quantum computation does not pose any challenge to the

Church-Turing Thesis. But what was not known initially was whether quantum computation can help us increase the efficiency of our algorithms. This all changed in the 1990s. The foundations work of David Deutsch in 1984 was expanded in 1992 along with Richard Jozsa to give the Deutsch-Jozsa algorithm in 1992 [51]. Another remarkable achievement of this time was the Simon's algorithm from 1994 [140], by Daniel R. Simon, which would greatly influence the factoring algorithm by Peter Shor.

In 1994, Peter Shor [139] discovered a surprising algorithm for factoring integers, a problem that is believed to be difficult for classical computers (i.e. non-quantum computers). In fact, a large portion of cryptography is based on the assumption that factoring integers is hard. Also, in 1996, Lov Kumar Grover [77] designed a quantum algorithm for unstructured searching that had quadratic speed up over the classical algorithms. Both these discoveries led to even greater interest in quantum computing and Shor's algorithm posed a challenge to the Extended Church-Turing Thesis - a question that is open to this day.

Both in the classical and quantum world different models of computations have been proposed over the years. Some models were motivated by the hardware technology of the time (like PRAM models). Some models were motivated from the need to optimize certain resources over others (like space-bounded computation model, query model, etc.). Some models were proposed with the aim to understand various structural properties of functions (like Boolean circuit complexity, arithmetic circuit complexity, etc.). Some models are defined to understand the complexity of certain kind of functions (like graph query models, various learning models, etc.). And some models were proposed due to sheer mathematical curiosity, some of which later turned out to be central to understanding the complexity of functions and also deeply connected to other models (like the communication model).

Once a model of computation is defined two natural goals emerges:

- **Question (1):** How much cost is sufficient to compute a function?
- **Question (2):** What is the cost that is necessary in order to compute a function?

Question (1) is the question of proving an upper bound on the cost by designing an algorithm in the model of interest. This usually requires designing an efficient algorithm to compute the function. Question (2) is that of proving lower bound. This often involves constructing special functions that are "hard" in that model of computation. To have a more refined understanding of the power of a model of computation one often asks the above questions with the additional guarantee that the function, to be computed, is of a special kind.

An equally important area of research is to understand the *relation between various complexity measures*. Some questions that one might ask in this line of research are: Does quantum computing help when we want to learn a function exactly? Can quantum communication complexity be much higher than the query complexity of a related function? Does the relationship between two complexity measures in the classical world also translate to the quantum world?

In this thesis, we will consider a few models of computation, namely models of computational learning, query model, communications model, and local query model for graphs. We will try to understand the power of these models for various classes of functions. We will be investigating how various complexity measures (defined as per different models of computation) relate to each other. In this endeavor we will be working both Question (1) (upper bound) and Question (2) (lower bound). As a part of our investigation, we use various mathematical tools like Fourier analysis, linear algebra, and geometry. Some questions that have motivated the work in this thesis are:

- How does structural simplicity affect its computational complexity in various models of computation?
  - Does Fourier analytical simplicity lead to easier computation?
  - Does geometric simplicity imply efficient communication?
- How does the behavior of quantum computing differs from its classical counterpart?
  - What is the relationship between classical and quantum query complexity of learning a function?
  - Does the same relation between a pair of classical complexity measures also hold in the quantum world?

On our way, we discover beautiful connections between these questions and areas like Fourier analysis and additive combinatorics. Understanding the complexity measures that we study and the relationship between these measures has been an ongoing work for several decades. In this thesis, we push the boundary of our knowledge a bit further.

We now define the models of computation that are of interest to us. These are the are models of computational learning, query model, communications model, and local query model for graphs. We start by describing the query models and then move on to learning models and communication models. Finally, we describe the local query model for graphs.

1.1 Various models of computation

1.1.1 The query model

The query model is arguably the simplest yet extremely useful model for computation of Boolean functions. The domain in this model is the domain of a fixed function and the range is the range of that function. We start by describing the classical query models and then move on to the quantum query model.

Classical query model

We first look at the deterministic query model. A function $f : \{0,1\}^n \to \{-1,1\}$ is known to a query algorithm and the goal is to compute $f(x)$ on an unknown input $x \in \{0,1\}^n$. In order to do this, the algorithm is allowed to make queries to x: assume that there is an oracle that knows the entire input x. By making a single query, the algorithm asks this oracle to reveal the ith bit of x, $i \in [n]$, where the algorithm chooses i. Each such call to oracle costs one unit. If the algorithm computes f correctly on all inputs x then we say that the algorithm computes f.

The cost of a query algorithm D that computes f on an input x is the number of queries it makes on x in order to output $f(x)$, denote this by $\text{cost}(f,x)$. The cost of D is the maximum number of queries the algorithm makes, where the maximum is over all $x \in \{0,1\}^n$. Formally,

$$\text{cost}(D) = \max_{x \in \{0,1\}^n} \text{cost}(D,x).$$

The deterministic query complexity of a function f, denoted by $D(f)$, is the cost of the best algorithm that computes f, formally:

$$D(f) = \min_{D \text{ computes } f} \text{cost}(D).$$

Before moving to randomized query complexity let us analyze the deterministic query complexity of AND function on n-bits, $\mathsf{AND}_n : \{0,1\}^n \to \{-1,1\}$. In Figure 1.1 we give a query algorithm that computes AND_n. The cost of this algorithm is n. It is not hard to show, by an adversarial argument, that any algorithm that computes AND_n must make n queries. Thus the deterministic query complexity of AND_n is n.

A randomized query algorithm R, for a function $f : \{0,1\}^n \to \{-1,1\}$, is a probability distribution over deterministic query algorithms. We say that R computes f with bounded

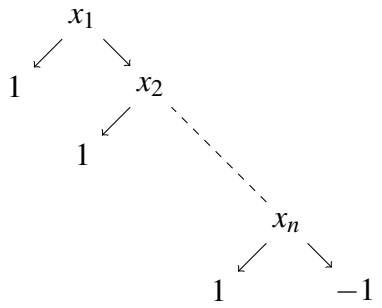


Fig. 1.1 A deterministic query algorithm for AND_n. If at any internal node x_i, the value of bit x_i is 0, the algorithm stops and outputs 1. Otherwise the algorithm outputs -1.

error if for all $x \in \{0,1\}^n$

$$\Pr[R(x) = f(x)] \geq 2/3. \tag{1.1}$$

By $R(x)$ in the above equation we mean the process of choosing a deterministic query algorithm according to the probability distribution R and returning the answer of that query algorithm on x.

The cost of R is the maximum over the cost of deterministic query algorithms in its support. The randomized query complexity of f, denoted by $R(f)$, is the cost of the best algorithm that computes f. Formally,

$$R(f) = \min_{R \text{ computes } f} \text{cost}(R).$$

Also note that the constant $2/3$ in Equation 1.1 can be replaced by any constant greater than $1/2$ without loss of generality.

Quantum query model

Let us now define the bounded-error quantum query model. Let $f : \{0,1\}^n \to \{-1,1\}$ be a Boolean function. A *quantum query* algorithm with T queries is a sequence:

$$U_0, Q_0, U_1, Q_1, \ldots, U_{T-1}, Q_{T-1}, U_T,$$

where U_i's, $i \in \{0,1\ldots,t\}$ are unitaries that do not depend on input while Q_i's, $i \in \{0,1\ldots,t\}$, are query transformations that depend on input. Each application of Q is counted as a single query and costs. The action of Q is illustrated in Figure 1.2

The query algorithm starts int the state $|\psi_0\rangle$ and ends with measuring the state $U_T \ldots Q_0 U_0 |\psi_0\rangle$. Similar to the classical setup, the query algorithm computes f if for all $x \in \{0,1\}^n$ the result

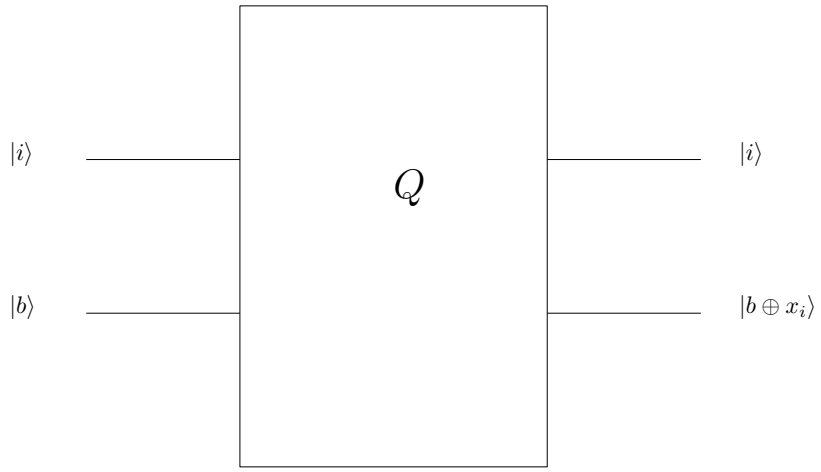


Fig. 1.2 An illustration of a single quantum query. The oracle Q has access to $x \in \{0,1\}^b$. On input $|i\rangle|b\rangle$, where $i \in \{0,1\}^{\log n}$ and $b \in \{0,1\}$, the oracle returns $|i\rangle|b \oplus x_i\rangle$.

of measurement is equal to $f(x)$ with probability at least $2/3$ and the query complexity of f, denoted by $Q(f)$, is the cost of the best protocol that computes f. In Table 1.1 we summarise the query models that we have discussed.

Variants	**Queries**	**Correctness** $\forall x \in \{0,1\}^n$
Deterministic	Bits	$\mathscr{A}(x) = f(x)$
Randomized	Bits	with probability $\geq \frac{2}{3}$, $\mathscr{A}(x) = f(x)$
Quantum	Qubits	with probability $\geq \frac{2}{3}$, $\mathscr{A}(x) = f(x)$

Table 1.1 Variants of query model. Here $\mathscr{A}$ is a query algorithm computing a function $f : \{0,1\}^n \to \{-1,1\}$.

1.1.2 The learning model

This model is defined primarily to study special classes of functions. The domain is the truth table of a class of functions and the range is the truth table of a function. We describe two learning models that are of interest to us.

Distribution Dependent Learning

Let us begin by explaining the setting of distribution-dependent learning from examples. Let $\mathscr{C}$ be a class of functions, that is, *concept class*. For concreteness assume they are ± 1-valued

functions on a domain of size N. If $N = 2^n$ then the domain may be identified with $\{0,1\}^n$. Suppose $c \in \mathscr{C}$ is an unknown function (the *target* function or concept) that we want to learn. A learning algorithm is given *examples* of the form $(x, c(x))$, where x is distributed according to some probability distribution D on $[N]$. An (ε, δ)-learner for $\mathscr{C}$ w.r.t. D is an algorithm that, for every possible target concept $c \in \mathscr{C}$, produces a hypothesis $h : [N] \to \{-1, 1\}$ such that with probability at least $1 - \delta$ (over the randomness of the learner and the examples for the target concept c), h's generalization error is at most ε:

$$\Pr_{x \sim D}[c(x) \neq h(x)] \leq \varepsilon.$$

In other words, from D-distributed examples the learner has to construct a hypothesis that mostly agrees with the target concept *under the same D*.

In the early days of quantum computing, Bshouty and Jackson [33] generalized this learning setting by allowing coherent *quantum* examples. A quantum example for target concept c w.r.t. distribution D, is the following $(\lceil \log N \rceil + 1)$-qubit state:

$$\sum_{x \in [N]} \sqrt{D(x)} |x, c(x)\rangle.$$

Clearly, such a quantum example is at least as useful as a classical example because measuring this state yields a pair $(x, c(x))$ where $x \sim D$. Bshouty and Jackson gave examples of concept classes that can be learned more efficiently from quantum examples than from classical random examples under specific D.

We focus on *exactly* learning the target concept from *uniform examples*, with high success probability. So $D(x) = 1/2^n$ for all x, $\varepsilon = 0$, and $\delta = 1/3$. A uniform quantum example for a concept $c \in \mathscr{C}$ is the quantum state

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, c(x)\rangle.$$

The goal, thus, is to exactly learn $c \in \mathscr{C}$ using the minimum number of uniform examples (each of which costs one unit) from c of the form $(x, c(x))$ where x is drawn from the uniform distribution on $\{0,1\}^n$.

Exact Active Learning

Another model that we are interested in is the model of exact active learning or *exact learning from membership queries* which was introduced in [9]. In this model the learner wants to

exactly learn an unknown target concept $c : [N] \to \{-1,1\}$ from a known concept class $\mathscr{C}$, but now the learner can choose which points of the truth-table of the target it sees, rather than those points being chosen randomly. More precisely, the learner can query $c(x)$ for any x of its choice. This is called a *membership query*: think of the set $\{x \mid c(x) = -1\}$ corresponding to the target concept: a membership query asks whether x is a member of this set or not.

In this model the quantum algorithms have the following query operation available:

$$O_c : |x,b\rangle \mapsto |x, b \cdot c(x)\rangle,$$

where $b \in \{-1,1\}$. The goal of a learning algorithm is to exactly learn a function using the minimum number of membership queries (each of which costs one unit) to the truth table of the function.

1.1.3 The communication model

Like the query model, the domain in this model is the domain of a fixed function and the range is the range of that function. We start by describing the classical communication models and then move on to the quantum communication model.

Classical communication model

Let $f : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$. In *communication complexity*, two players Alice and Bob get as inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ respectively, and the goal for the players is to devise a protocol to compute $f(x,y)$ by exchanging as few bits between themselves as possible. In all models of communication (including quantum, which we define later in this section) Alice and Bob are assumed to have unbounded power of computation and they are only charged for communication. We formalize this notion next.

The *deterministic communication complexity* $D^{cc}(f)$ of a function f is the minimum number of bits Alice and Bob will exchange in the worst case to deterministically compute the function f. See Figure 1.3 for an illustration of the setup of deterministic communication complexity and Figure 1.4 for an example of the EQUALTIY function which is defined as follows: Alice is given $x \in \{0,1\}^n$ and Bob is given $y \in \{0,1\}^n$. Their goal is to output -1 if $x = y$ and 1 otherwise. The protocol that we illustrate is simple. Alice sends x_1 to Bob, if $x_1 \neq y_1$ Bob outputs 1 and protocol terminates. Otherwise, Bob sends y_2 to Alice and the protocol continues.

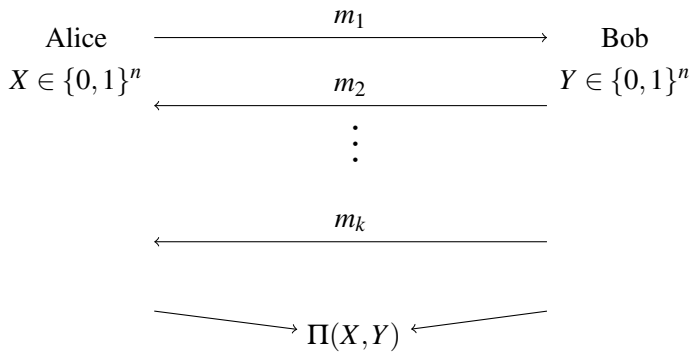


Fig. 1.3 A protocol Π computing $F : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$.

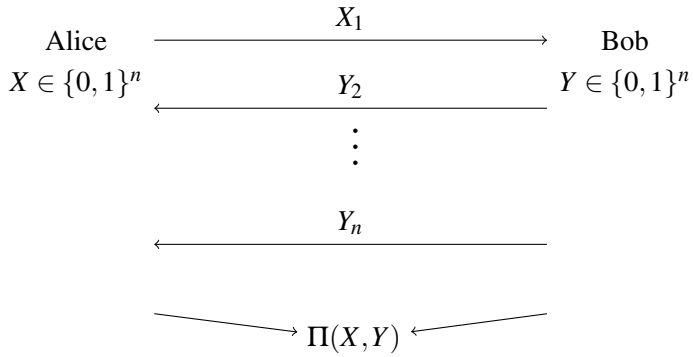


Fig. 1.4 A protocol Π computing EQUALITY: $\mathsf{EQUALITY}(X,Y) = -1$ iff $X = Y$.

In the randomized setting, both Alice and Bob share an infinite random source and the goal is to give the correct answer with probability at least $2/3$. The randomized communication complexity $R(f)$ of f denotes the minimum number of bits exchanged by the players in the worst case input by the best randomized protocol computing f. This is the setting of communication complexity with shared random coins. If no random string is shared, it is called the private random coins setting. By [118] we know that the communication complexity in both the setting differs by at most a logarithmic additive factor.

In both deterministic and randomized settings, Alice and Bob are allowed to make multiple rounds of interaction. Communication complexity when the number of rounds of interaction is bounded is also often studied. An important special case is when only one round of communication is allowed, that is, only Alice is allowed to send messages to Bob and Bob computes the output. We will denote by $D^{\rightarrow}(f)$ and $R^{\rightarrow}(f)$ the *one way deterministic communication complexity* and *one way randomized communication complexity* respectively, of f.

Quantum communication model

The setup of quantum communication is similar to classical communication except for now the messages and the channel of communication are quantum. There are several models in quantum communication complexity (see [49]). We refer to [49] for more details.

Let $f : \{0,1\}^n \times \{0,1\}^n \rightarrow \{-1,1\}$ be a communication problem and let Alice be given $x \in \{0,1\}^n$ and Bob be given $y \in \{0,1\}^n$. Alice and Bob have the extra resource of unbounded ERP-pairs (i.e. maximally entangled pairs of qubits) shared between them before communication starts (this is analogous to public randomness in classical communication complexity). Instead of bits, Alice and Bob are allowed to communicate qubits. Alice applies unitary to her input and the communication channel, this corresponds to Alice's computation and one round of communication. The cost of one round of communication is the number of qubits of channel affected. Bob does the same and the protocol continues. Measurement is performed at the end of the protocol.

The cost of a protocol is the maximum cost over all inputs and communication complexity of f, denoted by $Q^{cc}(f)$, is the cost of the best protocol computing f.

In Table 1.2 we summarize the communication models that we discussed in this section.

Variants	**Queries**	Resources	**Correctness** $\forall x,y \in \{0,1\}^n$
Deterministic	Bits		$\Pi(x,y) = f(x,y)$
Randomized	Bits	Shared Randomness	with probability $\geq \frac{2}{3}$, $\Pi(x,y) = f(x,y)$
Quantum	Qubits	Entanglement	with probability $\geq \frac{2}{3}$, $\Pi(x,y) = f(x,y)$

Table 1.2 Various Models of Communication complexity. Protocol Π for computing a function $f : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$.

1.1.4 Local query model

This model is defined to answer questions about the properties of graphs. The domain in this model are graphs over n vertices and the range is the set $\{1,\ldots,n\}$.

The Local Query Model is one of the most well-studied models in the area of graph property testing. Unlike the usual setting of algorithm design, in the local query model, we have restricted access to an unknown graph. We assume that there is an oracle that stores a set of local properties of the unknown graph and we can ask this oracle about those local properties: each such request to the oracle costs one unit and is called a query. We explain this formally next.

Let $G = (V,E)$, where $|V| = n$ and $|E| = m$, be an unknown graph. There are three types of queries, each costing one unit, that are allowed in the local query model:

- DEGREE query: given $u \in V$, the oracle reports the degree of u in V;
- NEIGHBOR query: given $u \in V$ and a positive integer i, the oracle reports the i-th neighbor of u, if it exists; otherwise, the oracle reports $\perp$;
- ADJACENCY query: given $u, v \in V$, the oracle reports whether $\{u,v\} \in E$.

Apart from the local queries mentioned, in the last few years, researchers have also used the RANDOM EDGE query [6, 14]. Every RANDOM EDGE query returns a uniformly random edge of G. Notice that the randomness will be over the probability space of all edges, and hence, a random edge query is not a local query.

We now describe the results of this thesis.

1.2 Overview of Results and Organization

This thesis is divided into three parts. The first is dedicated to quantum learning theory. In this part, we study the sample complexity of exactly learning classes of functions. We also explore the relation between additive combinatorics and quantum learning theory in this part. The second part of the thesis is dedicated to quantum query and communication complexity. In this part, we study a relationship between query and communication in the quantum world and how this relationship differs from its classical counterpart. The third part of the thesis studies classical query and communication complexity. We first investigate whether geometric simplicity implies efficient communication. Then, we resolve a fundamental problem, called the problem of approximating global minimum cut, in the local query model for graphs.

1.2.1 Part I

Both quantum computing and machine learning are hot topics at the moment, and their intersection has been receiving growing attention in recent years as well. On the one hand, there are particular approaches that use quantum algorithms like Grover search [77] and the Harrow-Hassidim-Lloyd linear-systems solver [80] to speed up learning algorithms for specific machine learning tasks (see [150, 135, 3, 25, 53] for recent surveys of this line of work). On the other hand there have been a number of more general results about the sample and/or time complexity of learning various concept classes using a quantum computer (see [11] for a survey). In the first two chapters of this part, we present two new results in the latter line of work. In the third chapter, we give refinements of an important result in additive combinatorics and show its connections to quantum learning theory.

Chapter 3

In order to define the concept class that we are interested in learning in this chapter, we first need to define some basic notions of Fourier analysis of Boolean functions referring to [121, 152] for more. The set of all functions with domain $\{0,1\}^n$ and range $\mathbb{R}$ forms a vector space of dimension 2^n. The Fourier basis for this vector space is the set of functions $\{\chi_S(X) := (-1)^{\langle x,S\rangle} x_i : S \subseteq [n]\}$. In the last expression and the rest of this thesis, we use the natural mapping between subsets of $[n]$ and $\{0,1\}^n$. Thus every $f : \{0,1\}^n \to \mathbb{R}$ has a unique representation of the form:

$$f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S(x) \text{ for all } x \in \{0,1\}^n,$$

where $\widehat{f}(S) \in \mathbb{R}$ for $S \subseteq [n]$, are called the Fourier coefficients of f. The above expression is defined to be the Fourier representation of f and the set $\{\widehat{f}(S), S \subseteq [n]\}$ is called the Fourier spectrum of f.

Define the inner product between functions $f, g : \{0,1\}^n \to \mathbb{R}$ as

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x \in \{0,1\}^n}[f(x) \cdot g(x)] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x).$$

Thus the expectation is uniform over all $x \in \{0,1\}^n$. Observe that the set of character functions, $\{\chi_S\}_{S \subseteq [n]}$, forms an orthonormal basis for the space of real-valued functions over the Boolean cube. Thus we have

$$\widehat{f}(S) = \langle f, \chi_S \rangle = \mathop{\mathbb{E}}_{x}[f(x)\chi_S(x)]$$

The Parseval's identity states that $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2]$. If f has range $\{-1, 1\}$, then Parseval gives $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = 1$, so $\{\widehat{f}(S)^2\}_{S \in \{0,1\}^n}$ forms a probability distribution.

In this chapter, we study exact learning of the concept class of k-Fourier-sparse Boolean functions, i.e. Boolean functions whose Fourier spectrum has at most k non-zero Fourier coefficients.

Variants of this class of Boolean functions have been studied in the area of sparse recovery ([81, 90]). Real valued Boolean functions (instead of Boolean valued) that have at Fourier sparsity at most k are extensively studied in [45, 27, 130, 37].

Quantum Fourier transform allows us to sample vectors from the Fourier support of the target function, with probability equal to the squared of the respective Fourier coefficients. This sampling procedure, called Fourier sampling, is used to give exact learning algorithm for the class of k-Fourier sparse Boolean functions. We show how to exactly learn a k-Fourier-sparse n-bit Boolean function from $O(k^{1.5}(\log k)^2)$ uniform quantum examples for that function. This improves over the bound of $\widetilde{\Theta}(kn)$ uniformly random *classical* examples by Haviv and Regev, 2015 ([84]). We state our main result formally below.

Theorem 1.1. *For every $n > 0$ and $k \leq 2^n$, the number of uniform quantum examples that suffice to learn the class of k-Fourier sparse Boolean functions with probability $\geq 2/3$ is $O(k^{1.5}(\log k)^2)$.*

We also give a non-matching lower bound of $\tilde{\Omega}(k)$ for the number of uniform quantum examples needed to learn the class of k-Fourier sparse functions in this chapter.

Theorem 1.2. *For every n, constant $c \in (0,1)$ and $k \leq 2^{cn}$, the number of uniform quantum examples necessary to learn the class of k-Fourier-sparse Boolean functions, with success probability $\geq 2/3$, is $\Omega(k \log k)$.*

Chapter 4

In this chapter, we consider the framework of active learning or learning through membership queries, and, like in the previous chapter, our focus is on exact learning. In this model, using the quantum adversary method ([7, 18, 142]) along with an entropic argument, we show that if a concept class $\mathscr{C}$ can be exactly learned using Q quantum membership queries, then it can also be learned using $O\left(\frac{Q^2}{\log Q} \log |\mathscr{C}|\right)$ *classical* membership queries. This improves the previous-best simulation result by Servedio and Gortler, 2004, by a $\log Q$-factor. We state our main theorem of this chapter below in the language of "spectral adversary" for the concept class $\mathscr{C}$, i.e. $ADV(\mathscr{C})$, which lower bounds Q. We refer the reader to Chapter 4 more details.

Theorem 1.3. *Let $\mathscr{C} \subseteq \{0,1\}^N$ be a concept class and*

$$ADV(\mathscr{C}) = \max_{\Gamma \geq 0} \| \Gamma \| / \max_{i \in [N]} \| \Gamma \circ D_i \|$$

be the nonnegative adversary bound for the exact learning problem corresponding to $\mathscr{C}$. Then there exists a classical learner for $\mathscr{C}$ using $O\left(\frac{ADV(\mathscr{C})^2}{\log ADV(\mathscr{C})} \log |\mathscr{C}|\right)$ membership queries that identifies the target concept with probability $\geq 2/3$.

Chapter 5

Chang's lemma [40] is a result in additive combinatorics with several applications to mathematics and computer science. This chapter is dedicated to a refinement of Chang's lemma and its application to quantum learning theory.

For Boolean functions, the lemma informally states that all the large Fourier coefficients of the indicator function of a large subset reside in a low dimensional subspace. We first give a refinement of Chang's lemma for the case of Fourier-sparse Boolean functions.

Theorem 1.4. *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1 - 2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$r \leq 2\alpha k \log k.$$

The above theorem improves Chang's lemma for k-Fourier sparse Boolean functions by a factor of $\sqrt{\alpha k}$.

Our results have seen a further refinement in [38]. We show that this new refinement is tight. We conclude this chapter by showing an application of our improvement of Chang's lemma in quantum learning, and by giving a concrete direction to obtain a better analysis of one of our learning algorithms from Chapter 3.

This part is based on:

- Srinivasan Arunachalam, Sourav Chakraborty, Troy Lee, Manaswi Paraashar and Ronald de Wolf. Two new results about quantum exact learning. *46th International Colloquium on Automata, Languages, and Programming (ICALP)*, 132 : 16:1–16:15, 2019, DOI: https://doi.org/10.4230/LIPIcs.ICALP.2019.16. QUANTUM, 2021, Volume 5, Page 587, DOI: https://doi.org/10.22331/q-2021-11-24-587.
- Parts of the following paper: Sourav Chakraborty, Nikhil S. Mande, Rajat Mittal, Tulasimohan Molli, Manaswi Paraashar and Swagato Sanyal. Tight Chang's-lemma-type bounds for Boolean functions. *Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2021, Volume 213, Pages 10:1–10:22, DOI: https://doi.org/10.4230/LIPIcs.FSTTCS.2021.10.

1.2.2 Part II

This part is dedicated to the study of the relation between quantum query and quantum communication complexity. We first define the well-known notion of composition of two functions which will allow us to define important communication problems.

Let f be an n-bit Boolean function and let G be a two-bit Boolean function like AND_2 or XOR_2. The function AND_2 evaluates to True is and only if both of its input bits are True. The function XOR_2 evaluates to True is and only if both of its input bits are different (i.e. XOR_2 evaluates to True is and only if its input is either $(\mathsf{True},\mathsf{False})$ or $(\mathsf{True},\mathsf{False})$). One way to to obtain a communication problem using f and G is to consider the composition of f and G which is defined to be the following function:

$$f \circ G(x_1, y_1, \ldots, x_n, y_n) = f(G(x_1, y_1), \ldots, G(x_1, y_1)).$$

Thus, $f \circ G$ is a function on $2n$ bits. $f \circ G$ is viewed as a two-party communication problem where the bits $x_1 \ldots, x_n$ are with Alice and the bits $y_1, \ldots, y_n$ are with Bob. We call f as the *outer function* and G as the *inner function*.

In this part, we study the relation between the query of a Boolean function f and the communication complexity of $f \circ G$.

Chapter 6

Buhrman, Cleve, and Wigderson [34] observed that for every n-bit Boolean function f the bounded-error quantum communication complexity of $(f \circ G)$ is $O(Q(f)\log n)$, where $Q(f)$ is the bounded-error quantum query complexity of f. Note that the randomized communication complexity of $(f \circ G)$ is bounded by $O(R(f))$, where $R(f)$ denotes the randomized query complexity of f. Thus, BCW simulation has an extra $O(\log n)$ factor appearing that is absent in classical simulation. A natural question is if this factor can be avoided. Perhaps somewhat surprisingly, in this chapter, we show that for $G = \mathsf{XOR}_2$, then the extra $\log n$ factor in the BCW simulation is unavoidable using techniques from approximation theory and Fourier analysis. To the best of our knowledge, it was not even known prior to this work whether there existed a total function F and 2-bit function G, such that $Q^{cc}(F \circ G) = \omega(Q(F))$. The main theorem of this chapter is formally stated below.

Theorem 1.5. *For any constant $0 < \delta < 1$, there exists a total function $F : \{-1,1\}^n \to \{-1,1\}$ for which $Q(F) = \Theta(n^\delta)$ and*

$$Q^{cc}(F \circ \mathsf{XOR}_2) = \Theta(Q(F)\log n).$$

Chapter 7

In the previous chapter, we showed that BCW simulation theorem is tight when the inner function, G, is XOR_2. This leaves the case when the inner function is AND_2. In this chapter, we construct outer functions such that BCW simulation theorem is tight when the inner function is AND_2. More generally, we give a recipe to construct outer functions such that there is a gap between query complexity of f and communication complexity of $(f \circ G)$, where $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. We also show that the function constructed in the previous chapter witnesses tightness of BCW Simulation when composed with AND_2. Our main theorem of this chapter is the following.

Theorem 1.6. *There exists total function $f : \{-1,1\}^n \to \{-1,1\}$, such that*

$$Q^{cc}(f \circ G) = \Omega(Q(f)\log n)$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Chapter 8

The role of symmetry has been extensively studied in complexity theory. A Boolean function f is said to be symmetric if f depends only on the Hamming weight of its inputs. This is equivalent to saying that on every input in the domain of f, the value of f remains unchanged if the bits of that input are arbitrarily permuted. A natural generalization of symmetric functions is the class of transitive functions.

In this chapter, we explore the role of symmetry in quantum query-to-communication simulation. One motivation for exploring the role of symmetry comes from the following fact. Aaronson and Ambainis, 2005, ([1]) showed that for the Disjointness function, a central problem in communication complexity, only a constant overhead is required in BCW Simulation Theorem. The outer function for the Disjointness function is a symmetric function. This leads us to the following question: Does symmetry of the outer function lead to no overhead in quantum query-to-communication simulation? We show that this is indeed the case.

Theorem 1.7. *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$, we have*

$$Q^{cc}(f \circ G) = O(Q(f)).$$

On the other hand, we construct a transitive function for which the logarithmic overhead can not be avoided in BCW Simulation Theorem.

Theorem 1.8. *There exists a transitive and total function $f : \{-1,1\}^n \to \{-1,1\}$, such that*

$$Q^{cc}(f \circ G) = \Omega(Q(f)\log n)$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

This part if based on the following papers:

- Sourav Chakraborty, Arkadev Chattopadhyay and Nikhil Mande. Quantum Query-To-Communication Simulation Needs a Logarithmic Overhead. *Computational Complexity Conference (CCC)*, 169 : 32:1–32:15, 2020, DOI: https://doi.org/10.4230/LIPIcs.CCC.2020.32. A preliminary version was presented in *Quantum Information Processing (QIP)*, 2020.
- Sourav Chakraborty, Arkadev Chattopadhyay, Peter Høyer, Nikhil S. Mande and Ronald de Wolf. Symmetry and Quantum Query-to-Communication Simulation. To

appear in To appear in *International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2022. *arxiv:2012.05233*.

1.2.3 Part III

This part of the thesis is devoted to classical query and communication complexity. In the first chapter of this part, we explore the role of the geometric simplicity, quantified by bounded Vapnik–Chervonenkis Dimension of set systems, in communication complexity. In the second chapter of this part, we give tight query complexity of approximating global minimum cut in the local query model. We start by defining Vapnik-Chervonenkis dimension and global minimum cut in graphs.

Vapnik and Chervonenkis [146] introduced the measure *Vapnik-Chervonenkis dimension* or *VC dimension* for set systems in the context of statistical learning theory. This is a natural measure of geometric simplicity of set systems. Let $\mathscr{S}$ be a collection of subsets of a $[n]$. For a subset y of $[n]$, define

$$\mathscr{S}|_y := \{y \cap x : x \in \mathscr{S}\}.$$

We say y is *shattered* by $\mathscr{S}$ if $\mathscr{S}|_y = 2^y$. VC-dimension of $\mathscr{S}$ is the size of the largest subset y of $[n]$ shattered by $\mathscr{S}$. VC dimension has found numerous connections and applications in many different areas like approximation algorithms, discrete and combinatorial geometry, computational geometry, discrepancy theory and many other areas [111, 44, 122, 112].

The global minimum cut of a connected, unweighted, undirected and simple graph $G = (V,E)$, $|V| = n$ and $|E| = m$, is a partition of the vertex set V into two sets S and $V \setminus S$ such that the number of edges between S and $V \setminus S$ is minimized. The problem of the global minimum cut is so fundamental that researchers keep coming back to it again and again across different models [96, 95, 113, 98, 117, 5, 129, 66, 64, 65, 94].

Chapter 9

The Disjointness problem is a central problem in communication complexity: Alice and Bob are given two subsets of $\{1,\ldots,n\}$ and they have to check if their sets intersect. We use the notation DISJ_n for this problem. Using the standard *rank argument* [102, 125] one can show that $D(\mathsf{DISJ}_n) = \Theta(n)$. In a breakthrough paper, Kalyanasundaram and Schnitger [93] proved that $R(\mathsf{DISJ}_n) = \Omega(n)$. Razborov [127] and Bar-Yossef et al. [17] gave alternate proofs for the above result.

It is also known that if the sets are assumed to be drawn from some restricted set systems then the communication complexity can be much lower. For example, Håstad and Wigderson, 2007, [82] considered the Disjointness problem when Alice and Bob are promised to be given sets of size at most k, $k \in [n]$,from a fixed set system that is known to both players. They showed that the randomized communication complexity of this problem is $\tilde{O}(k)$.

In this chapter, we explore how communication complexity changes with respect to the complexity of the underlying set system. The complexity measure for the set system that we use in this chapter is the VC dimension. More precisely, on any set system with VC dimension bounded by d, we analyze how large can the deterministic and randomized communication complexities be, as a function of d and n. The k-sparse set disjointness problem which was considered by Håstad and Wigderson, 2007, is one such set system with VC dimension k. We address the question of whether the randomized communication complexity is always upper bounded by a function of the VC dimension of the set system, and does there always exist a gap between the deterministic and randomized communication complexity for set systems with small VC dimension.

We construct two natural set systems of VC dimension d, motivated from geometry. Using these set systems we show that the deterministic and randomized communication complexity can be $\widetilde{\Theta}(d \log(n/d))$ for set systems of VC dimension d and this almost matches the deterministic upper bound for all set systems of VC dimension d.

We also study the deterministic and randomized communication complexities of the Set-Intersection problem, a problem related to the Disjoitnenss Problem, when sets belong to a set system of bounded VC dimension. We denote this problem by INT_n. Finally, we show that there exists set systems of VC dimension d such that both deterministic and randomized complexities for the set intersection problem can be as high as $\Theta(d \log(n/d))$, and this is almost tight among all set systems of VC dimension d.

The main theorem of this chapter is stated below.

Theorem 1.9. *Let* $1 \leq d \leq n$.

1. *There exists* $\mathscr{S} \subseteq 2^{[n]}$ *with VC-dim*$(\mathscr{S}) \leq d$ *such that randomized communication complexity of the Disjointness problem, when the inputs are promised to come from* $\mathscr{S}$*, is* $\Omega\left(d \frac{\log(n/d)}{\log\log(n/d)}\right)$.
2. *There exists* $\mathscr{S} \subseteq 2^{[n]}$ *with VC-dim*$(\mathscr{S}) \leq d$ *such that randomized communication complexity of the Set-Intersection problem, when the inputs are promised to come from* $\mathscr{S}$*, is* $\Omega\left(d \log \frac{n}{d}\right)$.

Chapter 10

Let CUT(G) denote this edge set corresponding to a minimum cut in G, and t denote $|\text{CUT}(G)|$. The algorithmic landscape for the minimum cut problem has been heavily influenced by Karger and Stein's work [95, 96] and algorithmic solutions for minimum cut across different models [113, 98, 117, 5, 129, 66, 64, 65, 94] [1] have revisited their approach [95, 96]. Fundamental graph parameter estimation problems, like estimation of the number of edges [58, 68], triangles [56], cliques [57], stars [70], etc. have been solved in the local and bounded query models [69, 68, 97]. Estimation of the size of MINCUT is also in the league of such fundamental problems to be solved in the model of local queries.

In this chapter, we resolve the query complexity of the global minimum cut problem for a graph by designing a randomized algorithm for approximating the size of minimum cut in a graph, where the graph can be accessed through local queries like DEGREE, NEIGHBOR, and ADJACENCY queries (see Section 1.1.4).

Given $\varepsilon \in (0,1)$, the algorithm with high probability outputs an estimate $\hat{t}$ satisfying the following $(1-\varepsilon)t \leq \hat{t} \leq (1+\varepsilon)t$, where t is the size of minimum cut in the graph. The expected number of local queries used by our algorithm is $\min\left\{m+n, \frac{m}{t}\right\} \text{poly}\left(\log n, \frac{1}{\varepsilon}\right)$ where n and m are the number of vertices and edges in the graph, respectively. Eden and Rosenbaum showed that $\Omega(m/t)$ local queries are required for approximating the size of minimum cut in graphs, but no local query based algorithm was known. Our algorithmic result coupled with the lower bound of Eden and Rosenbaum, 2018 ([54]), resolved the query complexity of the problem of estimating the size of minimum cut in graphs using local queries.

Building on the lower bound of Eden and Rosenbaum, we show that, for all $t \in \mathbb{N}$, $\Omega(m)$ local queries are required to decide if the size of the minimum cut in the graph is t or $t-2$. Also, we show that, for any $t \in \mathbb{N}$, $\Omega(m)$ local queries are required to find all the minimum cut edges even if it is promised that the input graph has a minimum cut of size t. Both of our lower bound results are randomized, and hold even if we can make RANDOM EDGE queries in addition to local queries.

We state the main theorem of this chapter below.

Theorem 1.10. *There exists an algorithm, with* DEGREE *and* NEIGHBOR *query access to an unknown graph $G = (V,E)$, that solves the minimum cut estimation problem with probability at least $2/3$. The expected number of queries used by the algorithm is* $\min\left\{m+n, \frac{m}{t}\right\} \text{poly}\left(\log n, \frac{1}{\varepsilon}\right)$.

[1] The list is to name a few and it is not exhaustive.

This part is based of the following papers:

- Anup Bhattacharya, Sourav Chakraborty, Arijit Ghosh, Gopinath Mishra and Manaswi Paraashar. Disjointness Through the Lens of Vapnik-Chervonenkis Dimension: Sparsity and Beyond. *International Conference on Randomization and Computation* (RANDOM), 176 : 23:1–23:15, 2020, DOI: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2020.23.
- Arijit Bishnu, Arijit Ghosh, Gopinath Mishra and Manaswi Paraashar. Query Complexity of Global Minimum Cut. International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), 2021, DOI: https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2021.6.

Chapter 2

Preliminaries

We start by mentioning some notations that will be used throughout the text. Then we go into an introduction to the Fourier analysis of Boolean functions, a main tool for several chapters of this thesis.

Notation

Let $[n] = \{1, \ldots, n\}$. We use the natural correspondence between subsets of $[n]$ and elements of $\{0,1\}^n$ interchangeably.

2.1 Fourier Analysis of Boolean Functions

The general representation of an n-bit Boolean function is as a function with domain $\{\mathsf{True}, \mathsf{False}\}^n$ to $\{\mathsf{True}, \mathsf{False}\}$, i.e. $f : \{\mathsf{True}, \mathsf{False}\}^n \to \{\mathsf{True}, \mathsf{False}\}$. Depending on the context we might view the functions to have domain $\{0,1\}^n$, $\{-1,1\}^n$ or $\mathbb{F}_2^n$ and the range as $\{0,1\}$, $\{-1,1\}$ or $\mathbb{F}_2$. In this introduction, let f be a function from $\{0,1\}^n$ to $\{-1,1\}$.

One of the main advantages of the above representation is that we think of Boolean functions as real valued functions. This allows us to use tools from Fourier analysis to study Boolean functions. This perspective has proven to be extremely useful, see, for example, the comprehensive book [121].

We now introduce the basics of Fourier analysis here, referring to [121, 152] for more. The set of all functions with domain $\{0,1\}^n$ and range $\mathbb{R}$ forms a vector space of dimension 2^n. The Fourier basis for this vector space is the set of functions $\{\chi_S(X) := (-1)^{\langle x,S\rangle} x_i : S \subseteq [n]\}$. In the previous expression, and the rest of this thesis, we use the natural mapping between

subsets of $[n]$ and $\{0,1\}^n$. The function $\chi_S : \{0,1\}^n \to \{-1,1\}$ is called the character function corresponding to the set $S \subseteq [n]$. Thus every $f : \{0,1\}^n \to \mathbb{R}$ has a unique representation of the form:

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x) \text{ for all } x \in \{0,1\}^n,$$

where $\widehat{f}(S) \in \mathbb{R}$ for $S \subseteq [n]$, are called the Fourier coefficients of f. The above expression is defined to be the Fourier representation of f and the set $\{\widehat{f}(S), S \subseteq [n]\}$ is called the Fourier spectrum of f.

Define the inner product between functions $f, g : \{0,1\}^n \to \mathbb{R}$ as

$$\langle f, g \rangle = \mathop{\mathbb{E}}_{x \in \{0,1\}^n}[f(x) \cdot g(x)] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x)g(x).$$

Thus the expectation is uniform over all $x \in \{0,1\}^n$. Observe that the set of character functions, $\{\chi_S\}_{S \subseteq [n]}$, forms an orthonormal basis for the space of real-valued functions over the Boolean cube. Thus we have

$$\widehat{f}(S) = \langle f, \chi_S \rangle = \mathop{\mathbb{E}}_{x}[f(x)\chi_S(x)]$$

The Parseval's identity states that $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2]$. If f has range $\{-1,1\}$, then Parseval gives $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = 1$, so $\{\widehat{f}(S)^2\}_{S \in \{0,1\}^n}$ forms a probability distribution.

As an illustration, we consider the AND function on two bits. The function $\mathsf{AND}_2 : \{0,1\}^2 \to \{-1,1\}$ is defined by $\mathsf{AND}_2(x) = -1$ if and only if $(x_1, x_2) = (1,1)$.

The Fourier representation of AND_2 is as follows:

$$\mathsf{AND}_2(x_1, x_2) = \frac{\chi_\emptyset(x_1, x_2)}{2} + \frac{\chi_{\{1\}}(x_1, x_2)}{2} + \frac{\chi_{\{2\}}(x_1, x_2)}{2} - \frac{\chi_{\{1,2\}}(x_1, x_2)}{2}.$$

Thus, the Fourier coefficients of AND_2 are

$$\widehat{\mathsf{AND}_2}(\emptyset) = \frac{1}{2}, \qquad \widehat{\mathsf{AND}_2}(\{1\}) = \frac{1}{2}, \widehat{\mathsf{AND}_2}(\{2\}) = \frac{1}{2}, \qquad \widehat{\mathsf{AND}_2}(\{1,2\}) = -\frac{1}{2}.$$

Next, we define some important Boolean functions.

2.2 Some important Boolean functions

We begin by recalling some common Boolean functions. As we have already seen, the AND function on two bits (AND_2) outputs true if and only if both its inputs are true. The truth table of AND_2 is given in Table 2.1. Similarly, AND function on n-bits, denoted by AND_n,

x_1	x_2	$\mathsf{AND}(x_1,x_2)$
1	1	-1
1	0	1
0	1	1
0	0	1

Table 2.1 Truth table of AND function on two bits.

outputs -1 if and only if all its n inputs are 1. The OR function on n-bits, denoted by OR_n, outputs 1 if and only if all its n inputs are 0 and the NOR function, denoted by NOR_2, is the negation of OR_n. When the domain size of functions are clear, we drop the subscript indicating the domain size.

The character functions that we defined previously are also called linear functions.

Definition 2.1 (Linear Functions)**.** *A function $f : \{0,1\}^n \to \{-1,1\}$ is called a linear function if there exists $S \in \{0,1\}^n$ such that for $x \in \{0,1\}^n$, $f(x) = (-1)^{\langle S,x\rangle}$, where $\langle S,x\rangle = \sum_{i=1}^{n} S_i x_i \ mod\ 2$.*

A special linear function is parity of n-bits, defined below:

$$\mathsf{PARITY}_n(x_1,\ldots,x_n) = (-1)^{\sum_{i\in[n]} x_i \bmod 2}. \tag{2.1}$$

A function may not depend of all n-bits. We capture this by the following definition.

Definition 2.2 (ℓ-Junta)**.** *We say $f : \{0,1\}^n \to \{-1,1\}$ is an* ℓ-junta *if there exists a set $S \subseteq [n]$ of size $|S| \leq \ell$ such that f depends only on the variables whose indices are in S.*

Now that we have studied several models of computation, we can motivate the definition of a few more important Boolean functions.

2.2.1 Partial Functions

Till now we have been considering functions defined on the every element of $\{0,1\}^n$. However it is natural to consider functions that are defined only on a subset of $\{0,1\}^n$. Let us

consider an example. In Section 1.1.3 we looked at the Disjointness problem: Alice and Bob are given subsets of $[n]$ and they want to decide whether their sets intersect. A variant of this problem is when we further assume that the sets given to Alice and Bob have sparsity at most k (where $k \in [n]$ is a parameter). With this promise, we get the Sparse-Set Disjointness problem. This function is defined only on the elements of $\{0,1\}^n$ who Hamming weight is at most k.

Formally, we define partial functions as follows.

Definition 2.3 (Partial Function)**.** *A partial function is a function of the form $f : \{0,1\}^n \to \{-1,1,\star\}$. If $f^{-1}(\star) = \emptyset$, then f is a total function.*

2.2.2 Composed Functions

Composed Boolean functions are an extremely important class of Boolean functions, especially for this thesis. We start by defining composition of two partial functions.

Definition 2.4 (Composition with partial functions)**.** *Let $f : \{-1,1\}^n \to \{-1,1,\star\}$ and $g : \{-1,1\}^m \to \{-1,1,\star\}$ be a partial functions. Let $f \circ g : \{-1,1\}^{nm} \to \{-1,1\}$ denote the composition of f and g that is defined as follows. On input $(X_1,\ldots,X_n) \in \{-1,1\}^{nm}$, where $X_i \in \{-1,1\}^m$ for all $i \in [n]$, $f(X_1,\ldots,X_n) = \star$ if there exists $i \in [n]$ such that $g(X_i) = \star$; otherwise $f \circ g(X_1,\ldots,X_n) = f(g(X_1),\ldots,g(X_n))$. We call f the outer function and g the inner function of the composed function $f \circ g$.*

The composition of total functions follows naturally from the above definition. A word on the notation: when talking about composition of Boolean functions, say f and g it is natural to have the range of g to be in the domain of f to make the definitions and notations simple. To this end we think of Boolean functions having domain $\{-1,1\}^n$ and range $\{-1,1\}$ when talking about composition. The second part of this thesis heavily studies composition of Boolean functions where we use this notation. At other places, we mention explicitly the notation of choice for the domain of Boolean functions.

Obtaining communication problems using composition

A very important use of composing Boolean functions is to obtain communication problems. Let us illustrate this by an example. Recall the Equality function from Section 1.1.3 (also see Figure 1.4). In this communication problem, Alice is given bits $X_1,\ldots,X_n$ and Bob is given bits $Y_1,\ldots,Y_n$. There goal is to decide whether $X_i = Y_i$ for all $i \in [n]$. The Equality function is in fact composition of NOR_n and XOR_2, as shown in Figure 2.1.

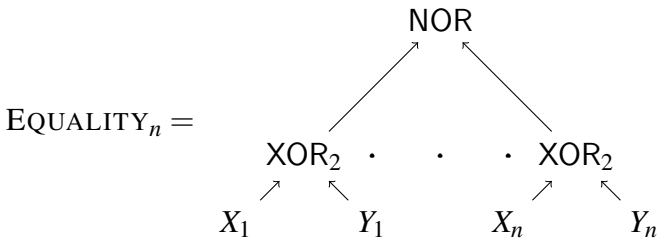


Fig. 2.1 The EQUALITY function on n-bits. The bits $X_1,\ldots,X_n$ are with Alice and bits $Y_1,\ldots,Y_n$ are with Bob.

Another extremely important function in communication complexity is the Disjointness function. We already discusses the importance of this function in the last chapter. In this problem Alice and Bob view their inputs, $(X_1,\ldots,X_n)$ and $(Y_1,\ldots,Y_n)$ respectively, as subsets of $[n]$, and they wish to output -1 if their sets are disjoint and 1 otherwise. The Disjointness problem is also a composed Boolean function as shown in Figure 2.2.

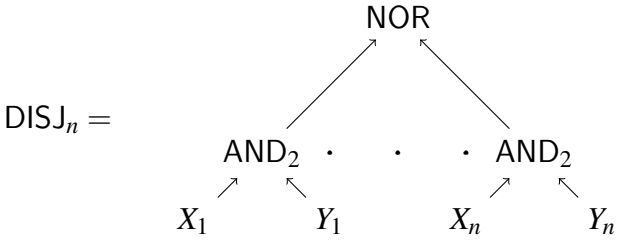


Fig. 2.2 The Disjointness function on n-bits. The bits $X_1,\ldots,X_n$ are with Alice and bits $Y_1,\ldots,Y_n$ are with Bob.

Observe that AND_2 or XOR_2 are the only non-constant functions on two bits, upto negations. Composed Boolean functions that represent communication problems and have their inner function as AND_2 (XOR_2) are are called AND functions (XOR functions).

Sometimes it becomes important to distinguish a Boolean function from a communication problem. For instance, AND_2 can be viewed both as a Boolean function and a communication problem. In order to avoid confusion we use the terminology of "two-party functions" when it is not clear from context whether we are thinking of a particular function as a communication problem or not.

Definition 2.5 (Two-party function)**.** *We call a function $G:\{-1,1\}^j\times\{-1,1\}^k\to\{-1,1\}$ a* two-party function *to indicate that it corresponds to a communication problem in which Alice is given input $x\in\{-1,1\}^j$, Bob is given input $y\in\{-1,1\}^k$, and their task is to compute $G(x,y)$.*

2.2.3 Classes of Boolean functions based on symmetry

We define two special classes of Boolean functions that are very well studies: Symmetric and Transitive functions.

Definition 2.6 (Symmetric functions)**.** *A function $f : \{0,1\}^n \to \{-1,1\}$ is symmetric if for all $\sigma \in S_n$ and for all $x \in \{0,1\}^n$ we have $f(x) = f(\sigma(x))$.*

It is easy to verify that a function is Symmetric if and only if the function only depends on the Hamming weight of its input. Important examples of Symmetric function are AND_n, OR_n, and Parity function on n-bits (see Equation 2.1).

Transitive functions can be seen as a natural generalization of Symmetric functions.

Definition 2.7 (Transitive functions)**.** *A function $f : \{0,1\}^n \to \{-1,1\}$ is transitive if for all $i, j \in [n]$ there exists a permutation $\sigma \in S_n$ such that*

- *$\sigma(i) = j$, and*
- *$f(x) = f(\sigma(x))$ for all $x \in \{0,1\}^n$.*

A prominent example of a transitive function is the Inner Product function.

Definition 2.8 (Inner Product function)**.** *For every positive integer n, define the function $\mathsf{IP}_n : \{0,1\}^n \times \{0,1\}^n \to \{-1,1\}$ by*

$$\mathsf{IP}_n(x,y) = \langle x,y \rangle.$$

In other words, $\mathsf{IP}_n = \mathsf{PARITY}_n \circ \mathsf{AND}_2$.

2.3 VC Dimension

Vapnik and Chervonenkis [146] introduced the measure *Vapnik-Chervonenkis dimension* or *VC dimension* for set systems in the context of statistical learning theory. VC dimension has been one of the fundamental measures for quantifying complexity of a collection of subsets and has found numerous connections and applications in many different areas like approximation algorithms, discrete and combinatorial geometry, computational geometry, discrepancy theory and many other areas [111, 44, 122, 112].

Definition 2.9. *Let $\mathscr{S}$ be a collection of subsets of a* universe *$\mathscr{U}$. For a subset y of $\mathscr{U}$, we define*

$$\mathscr{S}|_y := \{y \cap x : x \in \mathscr{S}\}.$$

We say a subset y of $\mathscr{U}$ is shattered *by $\mathscr{S}$ if $\mathscr{S}|_y = 2^y$, where 2^y denotes the power set of y.* Vapnik–Chervonenkis (VC) dimension *of $\mathscr{S}$, denoted as VC-dim($\mathscr{S}$), is the size of the largest subset y of $\mathscr{U}$ shattered by $\mathscr{S}$.*

Part I

Quantum Learning Theory

Chapter 3

Exact learning of Fourier sparse functions

3.1 Introduction

Recall the setting of distribution dependent learning from Section 1.1.2 for learning a concept $\mathscr{C}$ of $\{-1,1\}$-valued functions defined on domain size of N. Bshouty and Jackson [33] generalized the classical setup of learning through uniform examples to quantum setting setting by allowing coherent quantum examples. A learner is given access to quantum example for concept $c \in \mathscr{C}$ w.r.t. distribution D. This is the following $(\lceil \log N \rceil + 1)$-qubit state:

$$\sum_{x \in [N]} \sqrt{D(x)} |x, c(x)\rangle.$$

An (ε, δ)-learner for the concept class $\mathscr{C}$ w.r.t. D is an algorithm that, for every possible target concept $c \in \mathscr{C}$, produces a hypothesis $h : [N] \to \{-1,1\}$ such that with probability at least $1-\delta$ (over the randomness of the learner and the examples for the target concept c), h's generalization error is at most ε:

$$\Pr_{x \sim D}[c(x) \neq h(x)] \leq \varepsilon.$$

In particular, Bshouty and Jackson showed that the concept class of DNF-formulas can be learned in polynomial time from quantum examples under the *uniform* distribution (i.e. $D(x) = 1/2^n$ for all $x \in [N]$), something we do not know how to do classically (the best classical upper bound is quasi-polynomial time [147]). The key to this improvement

is the ability to obtain, from a uniform quantum example, a sample $S \sim \widehat{c}(S)^2$ distributed according to the squared *Fourier coefficients* of c [1]. This *Fourier sampling*, originally due to Bernstein and Vazirani [23], is very powerful. For example, if $\mathscr{C}$ is the class of linear functions on $\{0,1\}^n$ (see Definition 2.1), then the unknown target concept c is a character function $\chi_S(x) = (-1)^{\langle x,y \rangle}$; its only non-zero Fourier coefficient is $\widehat{c}(S)$ hence one Fourier sample gives us the unknown S with certainty. In contrast, learning linear functions from classical uniform examples requires $\Theta(n)$ examples. Another example where Fourier sampling is proven powerful is in learning the class of ℓ-juntas on n bits. Atıcı and Servedio [15] showed that $(\log n)$-juntas can be exactly learned under the uniform distribution in time polynomial in n. Classically it is a long-standing open question if a similar result holds when the learner is given uniform classical examples (the best known algorithm runs in quasi-polynomial time [116]). These cases (and others surveyed in [11]) show that uniform quantum examples (and in particular Fourier sampling) can be more useful than classical examples. However, this is not the case in Valiant's *PAC-learning* model [145] of distribution-independent learning. There we require the same learner to be an (ε, δ)-learner for $\mathscr{C}$ w.r.t. *every* possible distribution D. One can show in this model (and also in the broader model of *agnostic* learning) that the quantum and classical sample complexities are equal up to a constant factor [12].

We now define the concept class that we want to learn. In this chapter we consider learning the concept class $\mathscr{C}$ of k-Fourier-sparse Boolean functions:

$$\mathscr{C} = \{c : \{0,1\}^n \to \{-1,1\} : |\mathrm{supp}(\widehat{c})| \leq k\},$$

where $\mathrm{supp}(\widehat{c})$ is the number of non-zero Fourier coefficients in the Fourier spectrum of c. This is a natural generalization of the case of learning linear functions (Definition 2.1), which corresponds to $k = 1$. It also generalizes the case of learning ℓ-juntas (Definition 2.2) on n bits, which are functions of sparsity $k = 2^\ell$. Variants of the class of k-Fourier-sparse functions have been well-studied in the area of *sparse recovery*, where the goal is to recover a k-sparse vector $x \in \mathbb{R}^N$ given a low-dimensional linear sketch Ax for a so-called "measurement matrix" matrix $A \in \mathbb{R}^{m \times N}$. See [81, 90] for some upper bounds on the size of the measurement matrix that suffice for sparse recovery. Closer to what we discuss in this chapter, there has been extensive work on learning the concept class of n-bit *real-valued* functions that are k-sparse in the Fourier domain. In this direction Cheraghchi et al. [45] showed that $O(nk(\log k)^3)$ uniform examples suffice to learn this concept class, improving upon the works of Bourgain [27], Rudelson and Vershynin [130] and Candés and Tao [37].

[1] See Section 2.1 for an introduction to Fourier analysis of Boolean functions

Furthermore, we focus on *exactly* learning the target concept from *uniform examples*, with high success probability. So $D(x) = 1/2^n$ for all x, $\varepsilon = 0$, and $\delta = 1/3$. A *uniform quantum example* for a concept $c \in \mathscr{C}$ is the quantum state

$$\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, c(x)\rangle.$$

The goal, thus, is to exactly learn $c \in \mathscr{C}$ given uniform qauntum examples from c of the form $(x, c(x))$ where x is drawn from the uniform distribution on $\{0,1\}^n$.

Haviv and Regev [84] considered learning this concept class and showed the following results.

Theorem 3.1 (Corollary 3.6 of [84])**.** *For every $n > 0$ and $k \leq 2^n$, the number of uniform examples that suffice to learn $\mathscr{C}$ with probability $1 - 2^{-\Omega(n \log k)}$ is $O(nk \log k)$.*

Theorem 3.2 (Theorem 3.7 of [84])**.** *For every $n > 0$ and $k \leq 2^n$, the number of uniform examples necessary to learn $\mathscr{C}$ with constant success probability is $\Omega(k(n - \log k))$.*

3.1.1 Results and Organization

Our main results in this section are about the number of uniform *quantum* examples that are necessary and sufficient to exactly learn the class $\mathscr{C}$ of k-Fourier-sparse functions.

As we already saw, Haviv and Regev [84] showed that for classical learners $O(nk \log k)$ uniform examples suffice to learn k-Fourier-sparse functions, and $\Omega(nk)$ uniform examples are necessary. In Section 3.3 we study the number of uniform *quantum* examples needed to learn k-Fourier-sparse Boolean functions, and show that it is upper bounded by $O(k^{1.5}(\log k)^2)$. For $k \ll n^2$ this quantum bound is much better than the number of uniform examples used in the classical case.

Proving the upper bound is done in two phases. In the first phase we use the fact that a uniform quantum example allows us to Fourier-sample the target concept and, with some Fourier analysis of k-Fourier-sparse functions, we learn the Fourier span using $O(rk)$ examples, where r is the Fourier dimension of the target concept (see Section 3.2 for the definition of Fourier dimension). The main upper bound theorem of this chapter is stated below.

Theorem 3.3. *For every $n > 0$ and $k \leq 2^n$, the number of uniform quantum examples that suffice to learn $\mathscr{C}$ with probability $\geq 2/3$ is $O(k^{1.5}(\log k)^2)$.*

In the theorem below (Section 3.3.2) we prove the following (non-matching) lower bound on the number of uniform quantum examples necessary to learn $\mathscr{C}$ using techniques from quantum information theory.

Theorem 3.4. *For every n, constant $c \in (0,1)$ and $k \leq 2^{cn}$, the number of uniform quantum examples necessary to learn the class of k-Fourier-sparse Boolean functions, with success probability $\geq 2/3$, is $\Omega(k \log k)$.*

3.2 Preliminaries

Notation. For an n-dimensional vector space, the standard basis vectors are $\{e_i \in \{0,1\}^n \mid i \in [n]\}$, where e_i is the vector with a 1 in the ith coordinate and 0s elsewhere. For $x \in \{0,1\}^n$ and $i \in [n]$, let x^i be the input obtained by flipping the ith bit in x.

For a Boolean function $f : \{0,1\}^n \to \{-1,1\}$ and $B \in \mathbb{F}_2^{n \times n}$, define $f \circ B : \{0,1\}^n \to \{-1,1\}$ as $(f \circ B)(x) := f(Bx)$, where the matrix-vector product Bx is over $\mathbb{F}_2$. Throughout this paper, the rank of a matrix $B \in \mathbb{F}_2^{n \times n}$ will be taken over $\mathbb{F}_2$. Let $B_1, \ldots, B_n$ be the columns of B.

Recall from Section 2.1 the basic setup of Fourier analysis of Boolean functions. Every $f : \{0,1\}^n \to \mathbb{R}$ can be written uniquely as

$$f(x) = \sum_{S \in \{0,1\}^n} \widehat{f}(S)\chi_S(x) \qquad \text{for all } x \in \{0,1\}^n,$$

where $\chi_S(x) = (-1)^{\sum_{i=1}^n S_i x_i \bmod 2}$ and $\widehat{f}(S) = \langle f, \chi_S \rangle = \mathbb{E}_x[f(x)\chi_S(x)]$ is called a *Fourier coefficient* of f. For $i \in [n]$, we write $\widehat{f}(e_i)$ as $\widehat{f}(i)$ for notational convenience. Parseval's identity states that $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = \mathbb{E}_x[f(x)^2]$. If f has range $\{-1,1\}$, then Parseval gives $\sum_{S \in \{0,1\}^n} \widehat{f}(S)^2 = 1$, so $\{\widehat{f}(S)^2\}_{S \in \{0,1\}^n}$ forms a probability distribution. The *Fourier weight* of function f on $\mathscr{S} \subseteq \{0,1\}^n$ is defined as $\sum_{S \in \mathscr{S}} \widehat{f}(S)^2$.

For $f : \{0,1\}^n \to \mathbb{R}$, the *Fourier support* of f is $\mathrm{supp}(\widehat{f}) = \{S : \widehat{f}(S) \neq 0\}$. The *Fourier sparsity* of f is $|\mathrm{supp}(\widehat{f})|$. The *Fourier span* of f, denoted $\mathrm{Fspan}(f)$, is the span of $\mathrm{supp}(\widehat{f})$. The *Fourier dimension* of f, denoted $\mathrm{Fdim}(f)$, is the dimension of the Fourier span. We say f is *k-Fourier-sparse* if $|\mathrm{supp}(\widehat{f})| \leq k$.

3.2.1 Some structural results

We now state a number of known structural results about Fourier coefficients and dimension.

Theorem 3.5 ([133])**.** *The Fourier dimension of a k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ is $O(\sqrt{k}\log k)$.*

Lemma 3.6 ([71, Theorem 12])**.** *Let $k \geq 2$. The Fourier coefficients of a k-Fourier-sparse Boolean function $f : \{0,1\}^n \to \{-1,1\}$ are integer multiples of $2^{1-\lfloor \log k \rfloor}$.*

Definition 3.7. *Let $f : \{0,1\}^n \to \{-1,1\}$ and suppose $B \in \mathbb{F}_2^{n\times n}$ is invertible. Define f_B as*

$$f_B(x) = f((B^{-1})^{\mathsf{T}}x).$$

Lemma 3.8. *Let $f : \{0,1\}^n \to \mathbb{R}$ and suppose $B \in \mathbb{F}_2^{n\times n}$ is invertible. Then the Fourier coefficients of f_B are $\widehat{f_B}(Q) = \widehat{f}(BQ)$ for all $Q \in \{0,1\}^n$.*

Proof. Write out the Fourier expansion of f_B:

$$\begin{aligned} f_B(x) &= f((B^{-1})^{\mathsf{T}}x) \\ &= \sum_{S\in\{0,1\}^n} \widehat{f}(S)(-1)^{S\cdot((B^{-1})^{\mathsf{T}}x)} \\ &= \sum_{S\in\{0,1\}^n} \widehat{f}(S)(-1)^{(B^{-1}S)\cdot x} \\ &= \sum_{Q\in\{0,1\}^n} \widehat{f}(BQ)(-1)^{Q\cdot x}, \end{aligned}$$

where the third equality used $\langle S, (B^{-1})^{\mathsf{T}}x\rangle = \langle B^{-1}S, x\rangle$ and the last used the substitution $S = BQ$. □

An easy consequence is the next lemma:

Lemma 3.9. *Let $f : \{0,1\}^n \to \{-1,1\}$, and $B \in \mathbb{F}_2^{n\times n}$ be an invertible matrix such that the first r columns of B are a basis of the Fourier span of f, and $\widehat{f}(B_1), \ldots, \widehat{f}(B_r)$ are non-zero. Then*

1. *The Fourier span of $\widehat{f_B}$ is spanned by $\{e_1, \ldots, e_r\}$, i.e., f_B has only r influential variables.*
2. *For every $i \in [r]$, $\widehat{f_B}(i) \neq 0$.*

Here is the well-known fact, already mentioned in the introduction, that one can Fourier-sample from uniform quantum examples:

Lemma 3.10. *Let $f : \{0,1\}^n \to \{-1,1\}$. There exists a procedure that uses one uniform quantum example and satisfies the following: with probability $1/2$ it outputs an S drawn from the distribution $\{\widehat{f}(S)^2\}_{S\in\{0,1\}^n}$, otherwise it rejects.*

Proof. Using a uniform quantum example $\frac{1}{\sqrt{2^n}}\sum_x |x, f(x)\rangle$, one can obtain $\frac{1}{\sqrt{2^n}}\sum_x f(x)|x\rangle$ with probability $1/2$: unitarily replace $f(x) \in \{-1,1\}$ by $(1-f(x))/2 \in \{0,1\}$, apply the Hadamard transform to the last qubit and measure it. With probability $1/2$ we obtain the outcome 0, in which case our procedure rejects. Otherwise the remaining state is $\frac{1}{\sqrt{2^n}}\sum_x f(x)|x\rangle$. Apply Hadamard transforms to all n qubits to obtain $\sum_S \widehat{f}(S)|S\rangle$. Measuring this quantum state gives an S with probability $\widehat{f}(S)^2$. □

Information theory. We refer to [47] for a comprehensive introduction to classical information theory, and here just remind the reader of the basic definitions. A random variable $\mathbf{A}$ with probabilities $\Pr[\mathbf{A} = a] = p_a$ has *entropy* $H(\mathbf{A}) := -\sum_a p_a \log(p_a)$. For a pair of (possibly correlated) random variables $\mathbf{A}, \mathbf{B}$, the *conditional entropy* of $\mathbf{A}$ given $\mathbf{B}$, is $H(\mathbf{A} \mid \mathbf{B}) := H(\mathbf{A}, \mathbf{B}) - H(\mathbf{B})$. This equals $\mathbb{E}_{b\sim\mathbf{B}}[H(\mathbf{A} \mid \mathbf{B} = b)]$. The *mutual information* between $\mathbf{A}$ and $\mathbf{B}$ is $I(\mathbf{A} : \mathbf{B}) := H(\mathbf{A}) + H(\mathbf{B}) - H(\mathbf{A}, \mathbf{B}) = H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B})$. The *binary entropy* $H(p)$ is the entropy of a bit with distribution $(p, 1-p)$. If ρ is a density matrix (i.e., a trace-1 positive semi-definite matrix), then its singular values form a probability distribution P, and the *von Neumann entropy* of ρ is $S(\rho) := H(P)$. We refer to [120, Part III] for a more extensive introduction to quantum information theory.

3.3 Exact learning of k-Fourier-sparse functions

Our first theorem of this section (Section 3.3.1) gives an upper bound on the number of uniform quantum examples that are sufficient to learn $\mathscr{C}$ by giving a learning algorithm.

Theorem 3.11 (Restatement of Theorem 3.3)**.** *For every $n > 0$ and $k \leq 2^n$, the number of uniform quantum examples that suffice to learn $\mathscr{C}$ with probability $\geq 2/3$ is $O(k^{1.5}(\log k)^2)$.*

The learning algorithm has two phases: Phase 1 is described in Section 3.3.1 and Phase 2 is discussed in Section 3.3.1.

3.3.1 Upper bound on learning k-Fourier-sparse Boolean functions

We split our quantum learning algorithm into two phases. Suppose $c \in \mathscr{C}$ is the unknown concept, with Fourier dimension r. In the first phase the learner uses samples from the distribution $\{\widehat{c}(S)^2\}_{S\in\{0,1\}^n}$ to learn the Fourier span of c. In the second phase the learner

uses uniform *classical* examples to learn c exactly, knowing its Fourier span. Phase 1 uses $O(k(\log k)^2)$ uniform quantum examples (for Fourier-sampling) and Phase 2 uses $O(rk\log k)$ uniform *classical* examples. Note that since $r \geq \log k$, Phase 2 of our learner is always at least as expensive as Phase 1.

Theorem 3.12. *Let $k, r > 0$. There exists a quantum learner that exactly learns (with high probability) an unknown k-Fourier-sparse $c : \{0,1\}^n \to \{-1,1\}$ with Fourier dimension upper bounded by some known r, from $O(rk\log k)$ uniform quantum examples.*

The learner may not know the exact Fourier dimension r in advance, but Theorem 3.5 gives an upper bound $r = O(\sqrt{k}\log k)$, so our Theorem 3.11 follows immediately from Theorem 3.12.

Before we prove this Theorem 3.12, we first give a "trivial" algorithm for learning the Fourier support of Fourier-sparse functions quantumly. Gopalan et al. [71] showed that every k-Fourier-sparse Boolean function is "$2^{-\lceil \log k \rceil}$-granular", i.e., every Fourier coefficient of a k-Fourier-sparse Boolean function c is either 0 or an integer multiple of $2^{-\lceil \log k \rceil}$. Using this observation, if one is allowed to Fourier-sample from c, then each S with non-zero $\widehat{c}(S)$ will be observed with probability $\Omega(1/k^2)$, and using a coupon collector argument, we obtain the entire Fourier support using $O(k^2 \log k)$ many Fourier-samples. Our main contribution in Theorem 3.12 is to use the Fourier *dimension* in order to improve this trivial quantum algorithm. In particular observe that for functions with Fourier dimension $\log k$ (such as $(\log k)$-juntas), the theorem above scales as $O(k\log^2 k)$ which is better than the trivial algorithm by a factor of nearly k.

Phase 1: Learning the Fourier span

In this phase of the algorithm our goal is to learn the r-dimensional Fourier span of the k-Fourier-sparse target concept c, using $O(rk)$ Fourier-samples. The algorithm is very simple: Fourier-sample more and more S's and keep track of their span; stop when we reach dimension r. The key is the following technical lemma, which says that if our current span V' does not yet equal the full Fourier span V, then there is significant Fourier weight outside of V'. This implies that a small expected number of additional Fourier-samples will give us an $S \in V \setminus V'$, which will grow our current span. After r such grow-steps we have learned the full Fourier span.

Lemma 3.13. *Let $V \subseteq \{0,1\}^n$ be the r-dimensional Fourier span of k-Fourier-sparse function $c : \{0,1\}^n \to \{-1,1\}$, and $V' \subseteq V$ be a proper subspace. Then $\sum_{S \in V \setminus V'} \widehat{c}(S)^2 \geq 1/k$.*

Proof. Let us assume the worst case, which is that $\dim(V') = r-1$. Because we can do an invertible linear transformation on c as in Lemma 3.8, we may assume without loss of generality that the one "missing" dimension corresponds to the variable x_r (i.e., $V = \text{span}\{(\}V' \cup \{e_r\})$). Let g be the (not necessarily Boolean-valued) part of c with Fourier coefficients in V':

$$g(x) := \sum_{S \in V'} \widehat{c}(S)\chi_S(x).$$

Suppose, towards a contradiction, that the Fourier weight $W := \sum_{S \in V \setminus V'} \widehat{c}(S)^2$ is $< 1/k$. This implies that c and g have the same sign on every $x \in \{0,1\}^n$, as follows (using Cauchy-Schwarz):

$$|c(x) - g(x)| = \left| \sum_{S \in V \setminus V'} \widehat{c}(S)\chi_S(x) \right| \leq \sqrt{kW} < 1.$$

Since c depends on the variable x_r, there exists an $x \in \{0,1\}^n$ where x_r is influential, i.e., $c(x) \neq c(x^r)$. But g is independent of x_r, which implies $c(x) = \text{sgn}(g(x)) = \text{sgn}(g(x^r)) = c(x^r)$, a contradiction. Hence $W \geq 1/k$. □

We now conclude Phase 1 by presenting a quantum learning algorithm that learns the Fourier span of an unknown r-dimensional $c \in \mathscr{C}$, given uniform quantum examples for c.

Theorem 3.14. *Let $k, r > 0$. There exists a quantum learner that uses uniform quantum examples for an unknown k-Fourier-sparse $c : \{0,1\}^n \to \{-1,1\}$ with Fourier dimension r. After processing each new quantum example it outputs a subspace of the Fourier span of c. This sequence of subspaces is non-decreasing, and after an expected number of at most $2rk$ quantum examples, the output equals the Fourier span of c.*

This quantum learner can actually run forever, but if we know the Fourier dimension r of c, or an upper bound r on the actual Fourier dimension (e.g., by Theorem 3.5), then we can stop the learner after processing $6rk$ examples; now, by Markov's inequality, with probability $\geq 2/3$ the last subspace will be the Fourier span of c.

Proof. In order to learn the Fourier span of c, the quantum learner simply takes Fourier-samples until they span an r-dimensional space. Since we can generate a Fourier-sample from an expected number of 2 uniform quantum examples (by Lemma 3.10), the expected number of uniform quantum examples needed is at most twice the expected number of Fourier-samples. If our current sequence of Fourier-samples spans an r'-dimensional space V', with $r' < r$, then Lemma 3.13 implies that the next Fourier-sample has probability at least $1/k$ of yielding an $S \notin V'$. Hence an expected number of at most k Fourier-samples suffices

to grow the dimension of V' by at least 1. Since we stop at dimension r, the overall expected number of Fourier-samples is at most $2rk$. □

Phase 2: Learning the function completely

In the above Phase 1, the quantum learner obtains the Fourier span of c, which we will denote by $\mathscr{T}$. Using this, the learner can restrict to the following concept class

$$\mathscr{C}' = \{c : \{0,1\}^n \to \{-1,1\} \mid c \text{ is } k\text{-Fourier-sparse with Fourier span } \mathscr{T}\}$$

Let $\dim(\mathscr{T}) = r$. Let $B \in \mathbb{F}_2^{n\times n}$ be an invertible matrix whose first r columns form a basis for $\mathscr{T}$. Consider $c_B = c \circ (B^{-1})^{\mathsf{T}}$ for $c \in \mathscr{C}'$. By Lemma 3.9 it follows that c_B depends on only its first r bits, and we can write $c_B : \{0,1\}^r \to \{-1,1\}$. Hence the learner can apply the transformation $c \mapsto c \circ (B^{-1})^{\mathsf{T}}$ for every $c \in \mathscr{C}'$ and restrict to the concept class

$$\mathscr{C}'_r = \{c' : \{0,1\}^r \to \{-1,1\} \mid c' = c \circ (B^{-1})^{\mathsf{T}} \text{ for some } c \in \mathscr{C}' \text{ and invertible } B\}.$$

We now conclude Phase 2 of the algorithm by invoking the classical upper bound of Haviv-Regev (Theorem 3.1) which says that $O(rk\log k)$ uniform classical examples of the form $(z, c'(z)) \in \{0,1\}^{r+1}$ suffice to learn $\mathscr{C}'_r$. Although we assume our learning algorithm has access to uniform examples of the form $(x, c(x))$ for $x \in \{0,1\}^n$, the quantum learner knows B and hence can obtain a uniform example $(z, c'(z))$ for c' by letting z be the first r bits of $B^{\mathsf{T}}x$ and $c'(z) = c(x)$.

3.3.2 Lower bound on learning k-Fourier-sparse Boolean functions

In this section we show that $\Omega(k\log k)$ uniform quantum examples are necessary to learn the concept class of k-Fourier-sparse Boolean functions.

Theorem 3.15. *[Restatemet of Theorem 3.4] For every n, constant $c \in (0,1)$ and $k \leq 2^{cn}$, the number of uniform quantum examples necessary to learn the class of k-Fourier-sparse Boolean functions, with success probability $\geq 2/3$, is $\Omega(k\log k)$.*

Proof. Assume for simplicity that k is a power of 2, so $\log k$ is an integer. We prove the lower bound for the following concept class, which was also used for the classical lower bound of Haviv and Regev [84]: let $\mathscr{V}$ be the set of distinct subspaces in $\{0,1\}^n$ with dimension $n - \log k$ and

$$\mathscr{C} = \{c_V : \{0,1\}^n \to \{-1,1\} \mid c_V(x) = -1 \text{ iff } x \in V, \text{ where } V \in \mathscr{V}\}.$$

Note that every function in $\mathscr{C}$ has Fourier sparsity at most k, $|\mathscr{C}| = |\mathscr{V}|$, and each $c_V \in \mathscr{C}$ evaluates to 1 on a $(1-1/k)$-fraction of its domain.

We prove the lower bound for $\mathscr{C}$ using a three-step information-theoretic technique. A similar approach was used in proving classical and quantum PAC learning lower bounds in [12]. Let $\mathbf{A}$ be a random variable that is uniformly distributed over $\mathscr{C}$. Suppose $\mathbf{A} = c_V$, and let $\mathbf{B} = \mathbf{B}_1 \dots \mathbf{B}_T$ be T copies of the quantum example

$$|\psi_V\rangle = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, c_V(x)\rangle$$

for c_V. The random variable $\mathbf{B}$ is a function of the random variable $\mathbf{A}$. The following upper and lower bounds on $I(\mathbf{A} : \mathbf{B})$ are similar to [12, proof of Theorem 12] and we omit the details of the first two steps here.

1. $I(\mathbf{A} : \mathbf{B}) \geq \Omega(\log |\mathscr{V}|)$ because $\mathbf{B}$ allows one to recover $\mathbf{A}$ with high probability.
2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$ using a chain rule for mutual information.
3. $I(\mathbf{A} : \mathbf{B}_1) \leq O(n/k)$.

   Proof (of 3). Since $\mathbf{AB}$ is a classical-quantum state, we have

   $$I(\mathbf{A} : \mathbf{B}_1) = S(\mathbf{A}) + S(\mathbf{B}_1) - S(\mathbf{AB}_1) = S(\mathbf{B}_1),$$

   where the first equality is by definition and the second equality uses $S(\mathbf{A}) = \log |\mathscr{V}|$ since $\mathbf{A}$ is uniformly distributed over $\mathscr{C}$, and $S(\mathbf{AB}_1) = \log |\mathscr{V}|$ since the matrix

   $$\sigma = \frac{1}{|\mathscr{V}|} \sum_{V \in \mathscr{V}} |V\rangle\langle V| \otimes |\psi_V\rangle\langle\psi_V|$$

   is block-diagonal with $|\mathscr{V}|$ rank-1 blocks on the diagonal. It thus suffices to bound the entropy of the (vector of singular values of the) reduced state of $\mathbf{B}_1$, which is

   $$\rho = \frac{1}{|\mathscr{V}|} \sum_{V \in \mathscr{V}} |\psi_V\rangle\langle\psi_V|.$$

   Let $\sigma_0 \geq \sigma_1 \geq \dots \geq \sigma_{2^{n+1}-1} \geq 0$ be the singular values of ρ. Since ρ is a density matrix, these form a probability distribution. Now observe that $\sigma_0 \geq 1 - 1/k$ since the inner product between $\frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x, 1\rangle$ and every $|\psi_V\rangle$ is $1 - 1/k$. Let $\mathbf{N} \in \{0, 1, \dots, 2^{n+1} - 1\}$ be a random variable with probabilities $\sigma_0, \sigma_1, \dots, \sigma_{2^{n+1}-1}$, and $\mathbf{Z}$ an indicator for the event "$\mathbf{N} \neq 0$." Note that $\mathbf{Z} = 0$ with probability $\sigma_0 \geq 1 - 1/k$, and

$H(\mathbf{N} \mid \mathbf{Z} = 0) = 0$. By a similar argument as in [12, Theorem 15], we have

$$\begin{aligned} S(\rho) &= H(\mathbf{N}) = H(\mathbf{N}, \mathbf{Z}) = H(\mathbf{Z}) + H(\mathbf{N} \mid \mathbf{Z}) \\ &= H(\sigma_0) + \sigma_0 \cdot H(\mathbf{N} \mid \mathbf{Z} = 0) + (1 - \sigma_0) \cdot H(\mathbf{N} \mid \mathbf{Z} = 1) \\ &\leq H\Big(\frac{1}{k}\Big) + \frac{n+1}{k} \leq O\Big(\frac{n + \log k}{k}\Big) \end{aligned}$$

using $H(\alpha) \leq O(\alpha \log(1/\alpha))$.

Combining these three steps implies $T = \Omega(k(\log|\mathscr{V}|)/n)$. It remains to lower bound $|\mathscr{V}|$.

Claim 3.16. *The number of distinct d-dimensional subspaces of $\mathbb{F}_2^n$ is at least $2^{\Omega((n-d)d)}$.*

Proof. We can specify a d-dimensional subspace by giving d linearly independent vectors in it. The number of distinct sequences of d linearly independent vectors is exactly $(2^n - 1)(2^n - 2)(2^n - 4) \cdots (2^n - 2^{d-1})$, because once we have the first t linearly independent vectors, with span $\mathscr{S}_t$, then there are $2^n - 2^t$ vectors that do not lie in $\mathscr{S}_t$.

However, we are double-counting certain subspaces in the argument above, since there will be multiple sequences of vectors yielding the same subspace. The number of sequences yielding a fixed d-dimensional subspace can be counted in a similar manner as above and we get $(2^d - 1)(2^d - 2)(2^d - 4) \cdots (2^d - 2^{d-1})$. So the total number of subspaces is

$$\frac{(2^n - 1)(2^n - 2) \cdots (2^n - 2^{d-1})}{(2^d - 1)(2^d - 2) \cdots (2^d - 2^{d-1})} \geq \frac{(2^n - 2^{d-1})^d}{(2^d - 1)^d} \geq 2^{\Omega((n-d)d)}.$$

□

Combining this claim (with $d = n - \log k$) and $T = \Omega(k(\log|\mathscr{V}|)/n)$ gives $T = \Omega(k \log k)$.
□

3.4 Summary

In this chapter we studied quantum exact learning of the class of Boolean functions with Fourier sparsity at most k. Using Fourier sampling we gave a learning algorithm for this problem that uses $O(k^{1.5} \log k)$ samples. This algorithm has two phases. The first phase involved learning of Fourier span of the concept class and the second phase invoked the classical algorithm using the result of the first phase. We also gave a non-matching lower

bound on the number of uniform quantum examples to learn the class of k-Fourier sparse functions.

Chapter 4

Exact learning from membership queries

4.1 Introduction

We now consider the model of active learning or learning through membership queries which we introduced in Section 1.1.2. We start by recalling the model. Note that, similar to the last chapter, we are interested in exact learning. In this model the learner wants to exactly learn an unknown target concept $c : [N] \to \{-1,1\}$ from a known concept class $\mathscr{C}$, but now the learner can choose which points of the truth-table of the target it sees, rather than those points being chosen randomly. More precisely, the learner can query $c(x)$ for any x of its choice. This is called a *membership query*: think of the set $\{x \mid c(x) = 1\}$ corresponding to the target concept: a membership query asks whether x is a member of this set or not. Quantum algorithms have the following query operation available:

$$O_c : |x,b\rangle \mapsto |x,b \cdot c(x)\rangle,$$

where $b \in \{-1,1\}$.

For some concept classes, quantum membership queries can be much more useful than classical. Consider again the class linear functions on $\{0,1\}^n$ (see Definition 2.1). Using one query to a uniform superposition over all x and doing a Hadamard transform, we can Fourier-sample and hence learn the target concept exactly. In contrast, $\Theta(n)$ classical membership queries are necessary and sufficient for classical learners. As another example, consider the concept class $\mathscr{C} = \{\delta_i \mid i \in [N]\}$ of the N point functions, where $\delta_i(x) = -1$ iff $i = x$. Elements from this class can be learned using $O(\sqrt{N})$ quantum membership queries

by Grover's algorithm, while every classical algorithm needs to make $\Omega(N)$ membership queries.

For a given concept class $\mathscr{C}$ of ± 1-valued function on $[N]$, let $D(\mathscr{C})$ denote the minimal number of classical membership queries needed for learners that can exactly identify every $c \in \mathscr{C}$ with success probability 1 (such learners are deterministic without loss of generality). Let $R(\mathscr{C})$ and $Q(\mathscr{C})$ denote the minimal number of classical and quantum membership queries, respectively, needed for learners that can exactly identify every $c \in \mathscr{C}$ with error probability $\leq 1/3$.[1] Servedio and Gortler [136] showed that these quantum and classical measures cannot be too far apart. First, using an information-theoretic argument they showed

$$Q(\mathscr{C}) \geq \Omega\left(\frac{\log|\mathscr{C}|}{\log N}\right).$$

Intuitively, this holds because a learner recovers roughly $\log|\mathscr{C}|$ bits of information, while every quantum membership query can give at most $O(\log N)$ bits of information. Note that this is tight for the class of linear functions, where the left- and right-hand sides are both constant. Second, using the so-called hybrid method they showed

$$Q(\mathscr{C}) \geq \Omega(1/\sqrt{\gamma(\mathscr{C})}),$$

for some combinatorial parameter $\gamma(\mathscr{C})$ that we will not define here (but which is $1/N$ for the class $\mathscr{C}$ of point functions, hence this inequality is tight for that $\mathscr{C}$). They also noted the following upper bound:

$$D(\mathscr{C}) = O\left(\frac{\log|\mathscr{C}|}{\gamma(\mathscr{C})}\right).$$

Combining these three inequalities yields the following relation between $D(\mathscr{C})$ and $Q(\mathscr{C})$

$$D(\mathscr{C}) \leq O(Q(\mathscr{C})^2 \log|\mathscr{C}|) \leq O(Q(\mathscr{C})^3 \log N). \tag{4.1}$$

This shows that, up to a $\log N$-factor, quantum and classical membership query complexities of exact learning are polynomially close. While each of the three inequalities that together imply (4.1) can be individually tight (for different $\mathscr{C}$), this does not imply (4.1) itself is tight.

[1] We can identify each concept with a string $c \in \{-1,1\}^N$, and hence $\mathscr{C} \subseteq \{-1,1\}^N$. The goal is to learn the unknown $c \in \mathscr{C}$ with high probability using few queries to the corresponding N-bit string. This setting is also sometimes called "oracle identification" in the literature; see [11, Section 4.1] for more references.

4.1.1 Results and Organization

Note that Eq. (4.1) upper bounds the membership query complexity of *deterministic* classical learners. We are not aware of a stronger upper bound on *bounded-error* classical learners. However, in Section 4.3 we tighten that bound further by a $\log Q(\mathscr{C})$-factor:

$$R(\mathscr{C}) \leq O\left(\frac{Q(\mathscr{C})^2}{\log Q(\mathscr{C})} \log |\mathscr{C}|)\right) \leq O\left(\frac{Q(\mathscr{C})^3}{\log Q(\mathscr{C})} \log N\right).$$

Note that this inequality is tight both for the class of linear functions and for the class of point functions.

Our proof combines the quantum adversary method [7, 18, 142] with an entropic argument to show that we can always find a query whose outcome (no matter whether it's 0 or 1) will shrink the concept class by a factor $\leq 1 - \frac{\log Q(\mathscr{C})}{Q(\mathscr{C})^2}$. While our improvement over the earlier bounds is not very large, we feel our usage of entropy to save a log-factor is new and may have applications elsewhere. We formally state the main theorem of this chapter below (see Section 4.3).

Theorem 4.1. *Let $\mathscr{C} \subseteq \{0,1\}^N$ be a concept class and*

$$ADV(\mathscr{C}) = \max_{\Gamma \geq 0} \| \Gamma \| / \max_{i \in [N]} \| \Gamma \circ D_i \|$$

be the nonnegative adversary bound for the exact learning problem corresponding to $\mathscr{C}$. Then there exists a classical learner for $\mathscr{C}$ using $O\left(\frac{ADV(\mathscr{C})^2}{\log ADV(\mathscr{C})} \log |\mathscr{C}|\right)$ membership queries that identifies the target concept with probability $\geq 2/3$.

We start by describing some preliminaries needed for this chapter in the next section.

4.2 Preliminaries

We refer to the preliminaries of the last chapter for the relevant tools from information theory. The operator norm of a matrix A is the largest singular value of A and is denoted by $\| A \|$. For two matrices A and B of same dimensions, $A \circ B$ denotes the matrix obtained by taking the entry-wise product of A and B.

A key tool here will be the ("nonnegative" or "positive-weights") adversary method. This was introduced by Ambainis [7]; here we will use the formulation of Barnum et al. [18], which is called the "spectral adversary" in the survey [142].

Let $\mathscr{C} \subseteq \{0,1\}^N$ be a set of strings. If $N = 2^n$ then we may view such a string $c \in \mathscr{C}$ as (the truth-table of) an n-bit Boolean function, but in this chapter we do not need the additional structure of functions on the Boolean cube and may consider any positive integer N. Suppose we want to identify an unknown $c \in \mathscr{C}$ with success probability at least $2/3$ (i.e., we want to compute the identity function on $\mathscr{C}$). The required number of quantum queries to c can be lower bounded as follows. Let Γ be a $|\mathscr{C}| \times |\mathscr{C}|$ matrix with real, nonnegative entries and 0s on the diagonal (called an "adversary matrix"). Let D_i denote the $|\mathscr{C}| \times |\mathscr{C}|$ 0/1-matrix whose (c,c')-entry is $[c_i \neq c'_i]$.[2] Then it is known that at least (a constant factor times) $\| \Gamma \| / \max_{i \in [N]} \| \Gamma \circ D_i \|$ quantum queries are needed. Let

$$\mathrm{ADV}(\mathscr{C}) = \max_{\Gamma \geq 0} \frac{\| \Gamma \|}{\max_{i \in [N]} \| \Gamma \circ D_i \|}$$

denote the best-possible lower bound on $Q(\mathscr{C})$ that can be achieved this way.

4.3 Proof of Theorem 4.1

Our goal is to simulate quantum exact learners for a concept class $\mathscr{C}$ by classical exact learners, without using many more membership queries. The key to our classical simulation is the next lemma. It shows that if $Q(\mathscr{C})$ (and hence $\mathrm{ADV}(\mathscr{C})$) is small, then there is a query that splits the concept class in a "mildly balanced" way.

Lemma 4.2. *Let $\mathscr{C} \subseteq \{0,1\}^N$ be a concept class and*

$$ADV(\mathscr{C}) = \max_{\Gamma \geq 0} \| \Gamma \| / \max_{i \in [N]} \| \Gamma \circ D_i \|$$

be the nonnegative adversary bound for the exact learning problem corresponding to $\mathscr{C}$. Let μ be a distribution on $\mathscr{C}$ such that $\max_{c \in \mathscr{C}} \mu(c) \leq 5/6$. Then there exists an $i \in [N]$ such that

$$\min(\mu(C_i = 0), \mu(C_i = 1)) \geq \frac{1}{36 ADV(\mathscr{C})^2}.$$

Proof. Define unit vector $v \in \mathbb{R}_+^{|\mathscr{C}|}$ by $v_c = \sqrt{\mu(c)}$, and adversary matrix

$$\Gamma = vv^* - \mathrm{diag}(\mu),$$

[2] The bracket-notation $[P]$ denotes the truth-value of proposition P.

where $\mathrm{diag}(\mu)$ is the diagonal matrix that has the entries of μ on its diagonal. This Γ is a nonnegative matrix with 0 diagonal (and hence a valid adversary matrix for the exact learning problem), and $\| \Gamma \| \geq \| vv^* \| - \| \mathrm{diag}(\mu) \| \geq 1 - 5/6 = 1/6$. Abbreviate $A = \mathrm{ADV}(\mathscr{C})$. By definition of A, we have for this particular Γ

$$A \geq \frac{\| \Gamma \|}{\max_i \| \Gamma \circ D_i \|} \geq \frac{1}{6 \max_i \| \Gamma \circ D_i \|},$$

hence there exists an $i \in [N]$ such that $\| \Gamma \circ D_i \| \geq \frac{1}{6A}$. We can write $v = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$ where the entries of v_0 are the ones corresponding to Cs where $C_i = 0$, and the entries of v_1 are the ones where $C_i = 1$. Then

$$\Gamma = \begin{pmatrix} v_0 v_0^* & v_0 v_1^* \\ v_1 v_0^* & v_1 v_1^* \end{pmatrix} - \mathrm{diag}(\mu) \quad \text{and} \quad \Gamma \circ D_i = \begin{pmatrix} 0 & v_0 v_1^* \\ v_1 v_0^* & 0 \end{pmatrix}.$$

It is easy to see that $\| \Gamma \circ D_i \| = \| v_0 \| \cdot \| v_1 \|$. Hence

$$\frac{1}{36A^2} \leq \| \Gamma \circ D_i \|^2 = \| v_0 \|^2 \| v_1 \|^2 = \mu(C_i = 0)\mu(C_i = 1) \leq \min(\mu(C_i = 0), \mu(C_i = 1)),$$

where the last inequality used $\max(\mu(C_i = 0), \mu(C_i = 1)) \leq 1$. □

Note that if we query the index i given by this lemma and remove from $\mathscr{C}$ the strings that are inconsistent with the query outcome, then we reduce the size of $\mathscr{C}$ by a factor $\leq 1 - \Omega(1/\mathrm{ADV}(\mathscr{C})^2)$. Repeating this $O(\mathrm{ADV}(\mathscr{C})^2 \log |\mathscr{C}|)$ times would reduce the size of $\mathscr{C}$ to 1, completing the learning task. However, we will see below that analyzing the same approach in terms of entropy gives a somewhat better upper bound on the number of queries.

Theorem 4.3. *[Restatement of Theorem 4.1] Let $\mathscr{C} \subseteq \{0,1\}^N$ be a concept class and*

$$ADV(\mathscr{C}) = \max_{\Gamma \geq 0} \| \Gamma \| / \max_{i \in [N]} \| \Gamma \circ D_i \|$$

be the nonnegative adversary bound for the exact learning problem corresponding to $\mathscr{C}$. Then there exists a classical learner for $\mathscr{C}$ using $O\left(\frac{ADV(\mathscr{C})^2}{\log ADV(\mathscr{C})} \log |\mathscr{C}|\right)$ membership queries that identifies the target concept with probability $\geq 2/3$.

Proof. Fix an arbitrary distribution μ on $\mathscr{C}$. We will construct a deterministic classical learner for $\mathscr{C}$ with success probability $\geq 2/3$ under μ. Since we can do this for every μ, the

"Yao principle" [153] then implies the existence of a randomized learner that has success probability $\geq 2/3$ for every $c \in \mathscr{C}$.

Consider the following algorithm, whose input is an N-bit random variable $C \sim \mu$:

1. Choose an i that maximizes $H(C_i)$ and query that i.[3]
2. Update $\mathscr{C}$ and μ by restricting to the concepts that are consistent with the query outcome.
3. Goto 1.

The queried indices are themselves random variables, and we denote them by $I_1, I_2, \ldots$. We can think of t steps of this algorithm as generating a binary tree of depth t, where the different paths correspond to the different queries made and their binary outcomes.

Let P_t be the probability that, after t queries, our algorithm has reduced μ to a distribution that has weight $\geq 5/6$ on one particular c:

$$P_t = \sum_{i_1,\ldots,i_t \in [N],\ b \in \{0,1\}^t} \Pr[I_1 = i_1, \ldots, I_t = i_t, C_{i_1} \ldots C_{i_t} = b] \cdot [\exists c \in \mathscr{C} \ s.t.\ \mu(c \mid C_{i_1} \ldots C_{i_t} = b) \geq 5/6].$$

Because restricting μ to a subset $\mathscr{C}' \subseteq \mathscr{C}$ cannot decrease probabilities of individual $c \in \mathscr{C}'$, this probability P_t is non-decreasing in t. Because N queries give us the target concept completely, we have $P_N = 1$. Let T be the smallest integer t for which $P_t \geq 5/6$. We will run our algorithm for T queries, and then output the c with highest probability under the restricted version of μ we now have. With μ-probability at least $5/6$, that c will have probability at least $5/6$ (under μ conditioned on the query-results). The overall error probability under μ is therefore $\leq 1/6 + 1/6 = 1/3$.

It remains to upper bound T. To this end, define the following "energy function" in terms of conditional entropy:

$$\begin{aligned} E_t &= H(C \mid C_{I_1}, \ldots, C_{I_t}) \\ &= \sum_{i_1,\ldots,i_t \in [N],\ b \in \{0,1\}^t} \Pr[I_1 = i_1, \ldots, I_t = i_t, C_{i_1} \ldots C_{i_t} = b] \cdot H(C \mid C_{i_1} \ldots C_{i_t} = b). \end{aligned}$$

[3] Querying this i will give a fairly "balanced" reduction of the size of $\mathscr{C}$ irrespective of the outcome of the query. If there are several maximizing is, then choose the smallest i to make the algorithm deterministic.

Because conditioning on a random variable cannot increase entropy, E_t is non-increasing in t. We will show below that as long as $P_t < 5/6$, the energy shrinks significantly with each new query.

Let $C_{i_1} \dots C_{i_t} = b$ be such that there is no $c \in \mathscr{C}$ s.t. $\mu(c \mid C_{i_1} \dots C_{i_t} = b) \geq 5/6$ (note that this event happens in our algorithm with μ-probability $1 - P_t$). Let μ' be μ restricted to the class $\mathscr{C}'$ of concepts c where $c_{i_1} \dots c_{i_t} = b$. The nonnegative adversary bound for this restricted concept class is $A' = \mathrm{ADV}(\mathscr{C}') \leq \mathrm{ADV}(\mathscr{C}) = A$. Applying Lemma 4.2 to μ', there is an $i_{t+1} \in [N]$ with $p := \min(\mu'(C_{i_{t+1}} = 0), \mu'(C_{i_{t+1}} = 1)) \geq \frac{1}{36A'^2} \geq \frac{1}{36A^2}$. Note that $H(p) \geq \Omega(\log(A)/A^2)$. Hence

$$H(C \mid C_{i_1} \dots C_{i_t} = b) - H(C \mid C_{i_1} \dots C_{i_t} = b, C_{i_{t+1}}) = H(C_{i_{t+1}} \mid C_{i_1} \dots C_{i_t} = b) \geq \Omega(\log(A)/A^2).$$

This implies $E_t - E_{t+1} \geq (1 - P_t) \cdot \Omega(\log(A)/A^2)$. In particular, as long as $P_t < 5/6$, the $(t+1)$st query shrinks E_t by at least $\frac{1}{6}\Omega(\log(A)/A^2) = \Omega(\log(A)/A^2)$. Since $E_0 = H(C) \leq \log|\mathscr{C}|$ and E_t cannot shrink below 0, there can be at most $O\left(\frac{A^2}{\log A} \log|\mathscr{C}|\right)$ queries before P_t grows to $\geq 5/6$. □

Since $\mathrm{ADV}(\mathscr{C})$ lower bounds $Q(\mathscr{C})$, Theorem 4.3 implies the bound on $R(\mathscr{C})$ of $O\left(\frac{Q(\mathscr{C})^2}{\log Q(\mathscr{C})} \log|\mathscr{C}|)\right)$ claimed in our introduction. Note that this bound is tight up to a constant factor for the class of N-bit point functions, where $A = \Theta(\sqrt{N})$, $|\mathscr{C}| = N$, and $R(\mathscr{C}) = \Theta(N)$ classical queries are necessary and sufficient.

4.4 Summary

In this chapter we used quantum adversary method combined with entropic method to show that the number of queries a deterministic learner needs to learn a class is at most (roughly) quadratic of that of quantum learners for the same task. Our proofs went via adversary bound and an entropic argument.

Chapter 5

Chang's lemma and applications in quantum learning

5.1 Introduction

Chang's lemma [40] is a central result in additive combinatorics. This lemma has found several applications in mathematics [40, 73, 132, 74, 75] and theoretical computer science including complexity theory and algorithms [21, 39], analysis of Boolean functions [75, 143], communication complexity [143, 86] and extremal combinatorics [63]. See [89] for a proof of Chang's lemma.

Chang's lemma upper bounds the dimension of the span of the "large" Fourier coefficients. We state the lemma below for the special case of Boolean functions.

Lemma 5.1 (Chang's lemma)**.** *Let $\alpha \in (0,1)$ and $\rho > 0$. For every $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1 - 2\alpha$, we have*

$$\dim(span\{S : |\widehat{f}(S)| \geq \rho\alpha\}) \leq \frac{2\log(1/\alpha)}{\rho^2}. \tag{5.1}$$

Let us consider Chang's lemma for a k-Fourier-sparse Boolean function $f : \{0,1\}^n \to \{-1,1\}$ of Fourier dimension r and let $\rho \in (0,1]$. In particular, consider the case $\rho\alpha = 1/k$. In this case, since all elements of the Fourier support satisfy $|\widehat{f}(S)| \geq 1/k$ by Lemma 3.6, the left-hand side of Eq. (5.1) equals the Fourier dimension r of f. Thus Chang's lemma gives

$$r \leq 2\alpha^2 k^2 \log \rho k \leq 2\alpha^2 k^2 \log k. \tag{5.2}$$

Our main result of this chapter gives an almost quadratic improvement of the above upper bound on Fourier dimension, see Theorem 5.2.

Chang's lemma is also connected to quantum learning which we now explain. Recall Phase 1 of the learning algorithm from the Chapter 3, Section 3.3.1. Given a k-Fourier-sparse function c, Phase 1 starts by finding an $S \in \mathrm{supp}(\widehat{c})$ such that $S \neq 0^n$. Lemma 3.13 implies that an expected number of $O(k)$ many Fourier-samples are sufficient to sample such an S. Chang's lemma gives tighter bound on the expected number of samples for this step. Observe that Eqution 5.2 implies

$$\sum_{S \neq 0^n} \widehat{c}(S)^2 = \Omega\left(\frac{\sqrt{r}}{k\sqrt{\log k}}\right). \tag{5.3}$$

Thus an expected number of $O((k\sqrt{\log k})/\sqrt{r})$ many Fourier-samples are sufficient to obtain an $S \in \mathrm{supp}(\widehat{c})$ such that $S \neq 0^n$ in Phase 1. This is already an improvement from what Lemma 3.13 guaranteed. In Section 5.5, we improve this step of Phase 1. We also believe that entire Phase 1 should have $\tilde{O}(k)$ sample complexity. We give a concrete to achieve this in the same section.

5.1.1 Results and Organization

In Section 5.3 of this chapter we give an improvement of Chang's lemma for k-Fourier-sparse Boolean functions:

Theorem 5.2. *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1 - 2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$r \leq 2\alpha k \log k.$$

In a follow-up work [38], a further refinement of the above theorem was shown.

Theorem 5.3 ([38])**.** *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1 - 2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$r = O(\sqrt{\alpha k} \log k).$$

If $\alpha \leq 1/k$, then Theorem 5.3 gives a better bound on Fourier dimension than Theorem 5.2. In Section 5.4, we show that Theorem 5.3 is essentially tight:

Theorem 5.4. *For all $t \geq 2, t' \geq 4$, there exists a Boolean function f of Fourier sparsity $1+t^2(t'-1)$, Fourier dimension $(t\log t' + \log t')$ and satisfies $\widehat{f}(0^n) = 1 - 1/2t'$.*

In Section 5.5 we describe a potential way to improve the Phase 1 of our learning algorithm from Chapter 3

For convenience, we start by recalling some preliminaries from Chapter 3 along with some other facts in the following section.

5.2 Preliminaries

Recall the basic notions of Fourier analysis from Section 2.1 and Section 3.2. Every $f : \{0,1\}^n \to \mathbb{R}$ can be written uniquely as

$$f(x) = \sum_{S \in \{0,1\}^n} \widehat{f}(S)\chi_S(x) \qquad \text{for all } x \in \{0,1\}^n,$$

where $\widehat{f}(S)$ is called a *Fourier coefficient* of f. If f has range $\{-1,1\}$, then Parseval identity states that $\sum_{S\in\{0,1\}^n} \widehat{f}(S)^2 = 1$, so $\{\widehat{f}(S)^2\}_{S\in\{0,1\}^n}$ forms a probability distribution.

For $f : \{0,1\}^n \to \mathbb{R}$, the *Fourier support* of f is $\mathrm{supp}(\widehat{f}) = \{S : \widehat{f}(S) \neq 0\}$. The *Fourier sparsity* of f is $|\mathrm{supp}(\widehat{f})|$. The *Fourier span* of f, denoted $\mathrm{Fspan}(f)$, is the span of $\mathrm{supp}(\widehat{f})$. The *Fourier dimension* of f, denoted $\mathrm{Fdim}(f)$, is the dimension of the Fourier span. We say f is *k-Fourier-sparse* if $|\mathrm{supp}(\widehat{f})| \leq k$.

We need the notion of the *Fourier weight* of Boolean functions in this chapter. The Fourier weight of function f on $\mathscr{S} \subseteq \{0,1\}^n$ is defined as $\sum_{S\in\mathscr{S}} \widehat{f}(S)^2$. Define weight of a function f, denoted by $\delta(f)$, to be the probability that the function is -1. The weight of f has a natural connection with $\widehat{f}(\phi)$:

Observation 5.5. *Let $f : \{0,1\}^n \to \{-1,1\}$ be any function. Then,*

$$\widehat{f}(\emptyset) = 1 - 2\delta(f).$$

We require some known structural results about Fourier coefficients and dimension, for which we refer the reader to Section 3.2.1. We reproduce some of these results here for convenience.

Lemma 5.6 ([71, Theorem 12])**.** *Let $k \geq 2$. The Fourier coefficients of a k-Fourier-sparse Boolean function $f : \{0,1\}^n \to \{-1,1\}$ are integer multiples of $2^{1-\lfloor \log k \rfloor}$.*

Lemma 5.7. *Let $f : \{0,1\}^n \to \{-1,1\}$, and $B \in \mathbb{F}_2^{n\times n}$ be an invertible matrix such that the first r columns of B are a basis of the Fourier span of f, and $\widehat{f}(B_1), \ldots, \widehat{f}(B_r)$ are non-zero. Then*

1. *The Fourier span of $\widehat{f_B}$ is spanned by $\{e_1, \ldots, e_r\}$, i.e., f_B has only r influential variables.*
2. *For every $i \in [r]$, $\widehat{f_B}(i) \neq 0$.*

For any function $f : \{0,1\}^n \to \{-1,1\}$ and any real $t > 0$, define $\mathscr{S}_t := \{S \subseteq [n] : |\widehat{f}(S)| \geq 1/t\}$. Since Lemma 5.6 implies that $\mathscr{S}_k = \mathrm{supp}(\widehat{f})$, Theorem 5.2 and Theorem 5.3 can be seen as Chang's lemma type bounds for a special choice of threshold. These theorems give an upper bound on the dimension of $\mathscr{S}_k$, where k is the Fourier sparsity of the Boolean function under consideration. In the original spirit of Chang's lemma, it is therefore natural to consider thresholds other than $1/k$. We define two such natural choice of thresholds next that, to the best of our knowledge, were introduced in [38].

Definition 5.8. *Let $f : \{0,1\}^n \to \{-1,1\}$ be any function. Define the Fourier max-support-entropy of f, denoted $k'(f)$, by*

$$k'(f) := \operatorname*{argmin}_t \{\mathscr{S}_t = \mathrm{supp}(f)\}.$$

Equivalently,

$$k'(f) := \max_{S\in\mathrm{supp}(f)} \left\{ \frac{1}{|\widehat{f}(S)|} \right\}.$$

Define the Fourier max-rank-entropy of f, denoted $k''(f)$, by

$$k''(f) := \operatorname*{argmin}_t \{\dim(\mathscr{S}_t) = r(f)\}.$$

We do not study the above quantities in much in this thesis and refer the reader to [38] for a detailed analysis of these quantities and relationships between them. However, for purpose of completeness, we derive these quantities for the class of functions in Theorem 5.4.

5.3 Chang's Lemma for Fourier-sparse functions

We start by restating Theorem 5.2 in the following convenient form.

Theorem 5.9 (Restatement of Theorem 5.2)**.** *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1-2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$\widehat{f}(0^n) \leq 1 - \frac{r}{k \log k}.$$

Proof of Theorem 5.9. We first define the following notation. For $U \subseteq [r]$, let $f^{(U)}$ be the function obtained by fixing the variables $\{x_i\}_{i \in U}$ in f to $x_i = (1+\mathrm{sign}(\widehat{f}(i)))/2$ for all $i \in U$. Note that fixing variables cannot increase Fourier sparsity. For $i, j \in [r]$, define $f^{(i)} = f^{(\{i\})}$ and $f^{(ij)} = f^{(\{i,j\})}$. In this proof, for an invertible matrix $B \in \mathbb{F}_2^{n \times n}$, we will often treat its columns as a basis for the space $\mathbb{F}_2^n$. Recall $f_B(x) = f((B^{-1})^T x)$ from Definition 3.7. We let $f_B^{(i)}$ be the function obtained by fixing $x_i = (1+\mathrm{sign}(\widehat{f}(i)))/2$ in the function f_B.

The core idea in the proof of the theorem is the following structural lemma, which says that there is a particular x_i that we can fix in the function f_B without decreasing the Fourier dimension very much.

Lemma 5.10. *For every k-Fourier-sparse Boolean function $f : \{0,1\}^n \to \{-1,1\}$ with $\mathrm{Fdim}(f) = r$, there exists an invertible matrix $B \in \mathbb{F}_2^{n \times n}$ and an index $i \in [r]$ such that $\mathrm{Fdim}(f_B^{(i)}) \geq r - \log k$ and $\widehat{f_B}(j) \neq 0$ for all $j \in [r]$.*

We defer the proof of the lemma to later and first conclude the proof of the theorem assuming the lemma. Consider the matrix B defined in Lemma 5.10. Using Lemma 5.7 it follows that f_B has only r influential variables, so we can write $f_B : \{0,1\}^r \to \{-1,1\}$, where $\widehat{f_B}(j) \neq 0$ for every $j \in [r]$. Also, $\widehat{f_B}(0^r) = \widehat{f}(0^n) = 1-2\alpha$. For convenience, we abuse notation and abbreviate $f = f_B$. It remains to show that for every $f : \{0,1\}^r \to \{-1,1\}$ with $\widehat{f}(j) \neq 0$ for all $j \in [r]$, we have $2\alpha = 1 - \widehat{f}(0^r) \geq r/(k \log k)$. We prove this by induction on r.

Base case. Let $r = 1$. Then $k = 2$ (since $r \geq \log k$ and $k \geq 2$ by assumption). Note that the only Boolean functions with Fourier dimension 1 and $|\mathrm{supp}(\widehat{f})| \leq 2$ are $\{\chi_j, -\chi_j\}$. In both these cases $1 - \widehat{f}(0^r) = 1$ and $r/(k \log k) = 1/2$ (although the Fourier sparsity of χ_j is 1, we are implicitly working with a concept class of 2-sparse Boolean functions, hence $k = 2$).

Induction hypothesis. Suppose that for all $p \in \{1, \ldots, r-1\}$ and k-Fourier-sparse Boolean function $g : \{0,1\}^p \to \{-1,1\}$ with $\mathrm{Fdim}(g) = p$ and $\widehat{g}(j) \neq 0$ for all $j \in [p]$, we have $1 - \widehat{g}(0^p) \geq p/(k \log k)$.

Induction step. Let $i \in [r]$ be the index from Lemma 5.10. Note that $f^{(i)}$ is still k-Fourier sparse and $\widehat{f^{(i)}}(0^{r-1}) = 1 - 2\alpha + |\widehat{f}(i)|$. Since $|\widehat{f}(i)| \geq 1/k$ (by Lemma 5.6), we have

$$\widehat{f^{(i)}}(0^{r-1}) \geq 1 - 2\alpha + 1/k.$$

Since $r - \log k \leq \mathrm{Fdim}(f^{(i)}) \leq r-1$, we can use the induction hypothesis on the function $f^{(i)}$ to conclude that

$$2\alpha \geq 1 - \widehat{f^{(i)}}(0^{r-1}) + \frac{1}{k} \geq \frac{r - \log k}{k \log k} + \frac{1}{k} = \frac{r}{k \log k}.$$

This concludes the proof of the induction step and the theorem. We now prove Lemma 5.10.

Proof of Lemma 5.10. In order to construct B as in the lemma statement, we first make the following observation.

Observation 5.11. *For every Boolean function $f : \mathbb{F}_2^n \to \{-1,1\}$ with $\mathrm{Fdim}(f) = r$, there exists an invertible $B \in \mathbb{F}_2^{n \times n}$ such that:*

1. *The Fourier coefficient $\widehat{f_B}(1)$ is non-zero.*
2. *There exists a $t \in [r]$ such that, for all $j \in \{2, \ldots, t\}$, we have $\mathrm{Fdim}(f_B^{(j)}) \leq r - t$.*
3. *The Fourier span of $f_B^{(1)}$ is spanned by $\{e_{t+1}, \ldots, e_r\}$.*
4. *For $\ell \in \{t+1, \ldots, r\}$, the Fourier coefficients $\widehat{f_B^{(1)}}(\ell)$ are non-zero.*

We defer the proof of this observation to the end. We proceed to prove the lemma assuming the observation. Note that Property 3 gives the following simple corollary:

Corollary 5.12. *$f_B^{(1)}$ is a function of $x_{t+1}, \ldots, x_r$ and independent of $x_2, \ldots, x_t$ (and hence $f_B^{(1)} = f_B^{(i1)} = f_B^{(1i)}$ for every $i \in \{2, \ldots, t\}$).*

We now show that not only $f_B^{(1)}$, but all the functions $f_B^{(2)}, \ldots, f_B^{(t)}$ are independent of $x_2, \ldots, x_t$.

Claim 5.13. *For all $i \in \{2, \ldots, t\}$, $f_B^{(i)}$ is a function of $\{x_1, x_{t+1}, \ldots, x_r\}$ and independent of $x_2, \ldots, x_t$.*

Proof. Without loss of generality, let $i = 2$. By Observation 5.11 (property 4), the character functions $\chi_{t+1}, \ldots, \chi_r$ are present in the Fourier expansion of $f_B^{(1)}$. We have $f_B^{(21)} = f_B^{(1)}$ by Corollary 5.12. Hence, for every $\ell \in \{t+1, \ldots, r\}$, at least one of the characters χ_ℓ or

$\chi_1\chi_\ell$ is present in the Fourier expansion of $f_B^{(2)}$. Let y_ℓ be χ_ℓ or $\chi_1\chi_\ell$ (depending on which character function is present in the Fourier expansion of $f_B^{(2)}$). Note that the $r-t$ character functions $y_{t+1},\ldots,y_r$ are linearly independent. By Observation 5.11 (Property 2), we have $\mathrm{Fdim}(f_B^{(2)}) \leq r-t$, which implies $\mathrm{Fspan}(f_B^{(2)}) \subseteq \mathrm{span}\{y_{t+1},\ldots,y_r\}$ and $f_B^{(2)}$ is independent of $\{x_2,\ldots,x_t\}$. The same argument shows that for every $i,k \in \{2,\ldots,t\}$, $f_B^{(i)}$ is independent of x_k. □

Claim 5.14. *There exists an assignment of* $(x_1,x_{t+1},\ldots,x_r)$ *to* $(a_1,a_{t+1}\ldots,a_r)$ *in* f_B *such that the resulting function depends on* all *variables* $x_2,\ldots,x_t$.[1]

Proof. Before proving the claim we first make the following observation. Let us consider an assignment of $(x_1,x_{t+1},\ldots,x_r) = z$ in f_B and assume that the resulting function $f_{B,z}$ is independent of x_i for some $i \in \{2,\ldots,t\}$. Let us assign $x_i = (1+\mathrm{sign}(\widehat{f_B}(i)))/2$ in $f_{B,z}$ and call the resulting function $f_{B,z}^{(i)}$. Firstly, $f_{B,z}^{(i)} = f_{B,z}$ since $f_{B,z}$ was independent of x_i. Secondly, observe that $f_{B,z} = f_{B,z}^{(i)}$ could have alternatively been obtained by *first* fixing $x_i = (1+\mathrm{sign}(\widehat{f}(i)))/2$ in f_B and then fixing $(x_1,x_{t+1}\ldots,x_r) = z$. In this case, by Claim 5.13, after fixing x_i in f_B, $f_B^{(i)}$ is independent of $x_2,\ldots,x_t$ and after fixing $(x_1,x_{t+1},\ldots,x_r) = z$, $f_{B,z}$ is a constant. This in particular shows that if there exists a z such that $f_{B,z}$ is independent of x_i for some $i \in \{2,\ldots,t\}$, then $f_{B,z}$ is also independent of $x_2,\ldots,x_t$.

Towards a contradiction, suppose that for every assignment of $(x_1,x_{t+1},\ldots,x_r) = z$ to f_B, the resulting function $f_{B,z}$ is independent of x_i, for some $i \in \{2,\ldots,t\}$. Then by the argument in the previous paragraph, for every assignment z, $f_{B,z}$ is also independent of x_k for *every* $k \in \{2,\ldots,t\}$. This, however, contradicts the fact that $x_2,\ldots,x_t$ had non-zero influence on f_B (since B was chosen such that $\widehat{f_B}(j) \neq 0$ for every $j \in [r]$ in Lemma 5.10). This implies the existence of an assignment $(x_1,x_{t+1},\ldots,x_r) = (a_1,a_{t+1}\ldots,a_r)$, such that the resulting function depends on all the variables $x_2,\ldots,x_t$. □

We now argue that the assignment in Claim 5.14 results in a function which resembles the AND function on $x_2,\ldots,x_t$, and hence has Fourier sparsity 2^{t-1}.

Claim 5.15. *Consider the assignment* $(x_1,x_{t+1},\ldots,x_r) = (a_1,a_{t+1}\ldots,a_r)$ *in* f_B *as in Claim 5.14, then the resulting function* g *equals (up to possible negations of input and output bits) the* $(t-1)$*-bit* AND *function.*

[1] Observe that in this assignment, we have $x_1 = (1-\mathrm{sign}(\widehat{f}(1)))/2$. Otherwise, by assigning $x_1 = (1+\mathrm{sign}(\widehat{f}(1)))/2$ in f_B, we would obtain the function $f_B^{(1)}$ which we know is independent of $\{x_2,\ldots,x_t\}$ by Corollary 5.12.

Proof. By Claim 5.14, g depends on all the variables $x_2,\ldots,x_t$. This dependence is such that if *any one* of the variables $\{x_i : i \in \{2,\ldots,t\}\}$ is set to $x_i = (1+\text{sign}(\widehat{f_B}(i)))/2$, then by Claim 5.13 the resulting function $g^{(i)}$ is independent of $x_2,\ldots,x_t$. Hence, $g^{(i)}$ is some constant $b_i \in \{-1,1\}$ for every $i \in \{2,\ldots,t\}$. Note that these b_is are all the same bit b, because first fixing x_i (which collapses g to the constant b_i) and then x_j gives the same function as first fixing x_j (which collapses g to b_j) and then x_i. Additionally, by assigning $x_i = (1-\text{sign}(\widehat{f_B}(i)))/2$ for every $i \in \{2,\ldots,t\}$ in g, the resulting function must evaluate to $1-b$ because g is non-constant (it depends on $x_2,\ldots,x_t$). Therefore g equals (up to possible negations of input and output bits) the $(t-1)$-bit AND function. □

We now conclude the proof of Lemma 5.10. Let $f : \{0,1\}^n \to \{-1,1\}$ be such that $\text{Fdim}(f) = r$. Let B be as defined in Observation 5.11. Consider the assignment of $(x_{t+1},\ldots,x_r) = (a_{t+1},\ldots,a_r)$ to f_B as in Claim 5.15, and call the resulting function f'_B. From Claim 5.15, observe that by setting $x_1 = a_1$ in f'_B, the resulting function is $g(x_2,\ldots,x_t)$ and by setting $x_1 = 1-a_1$ in f'_B, the resulting function is a constant. Hence f'_B can be written as

$$f'_B(x_1,\ldots,x_t,a_{t+1},\ldots,a_r) = \frac{1-(-1)^{x_1+a_1}}{2} b_{a_1,a_{t+1},\ldots,a_r} + \frac{1+(-1)^{x_1+a_1}}{2} g(x_2,\ldots,x_t), \tag{5.4}$$

where $b_{a_1,a_{t+1},\ldots,a_r} \in \{-1,1\}$ (note that it is independent of $x_2,\ldots,x_t$ by Corollary 5.12). Since g essentially equals the $(t-1)$-bit AND function (by Claim 5.15), g has Fourier sparsity 2^{t-1} and $\widehat{g}(0^{t-1}) = 1-2^{-t+2}$. Hence the Fourier sparsity of f'_B in Eq. (5.4) equals 2^t. Since f'_B was a restriction of f_B, the Fourier sparsity of f'_B is at most k, hence $t \le \log k$. This implies $\text{Fdim}(f_B^{(1)}) = r-t \ge r-\log k$, concluding the proof. □

It remains to prove Observation 5.11, which we do now.

Proof of Observation 5.11. Let $D \in \mathbb{F}_2^{n\times n}$ be an invertible matrix that maximizes $\text{Fdim}(f_D^{(1)})$ subject to the constraint $\widehat{f_D}(1) \neq 0$. Suppose $\text{Fdim}(f_D^{(1)}) = r-t$. Let $d_1,\ldots,d_{r-t}$ be a basis of $\text{Fspan}(f_D^{(1)})$ such that $\widehat{f_D^{(1)}}(d_i) \neq 0$ for all $i \in [r-t]$. We now construct an invertible $C \in \mathbb{F}_2^{n\times n}$ whose first r columns form a basis for $\text{Fspan}(f_D)$, as follows: let $c_1 = e_1$, and for $i \in [r-t]$, fix $c_{t+i} = d_i$. Next, assign vectors $c_2,\ldots,c_t$ arbitrarily from $\text{Fspan}(f_D)$, ensuring that $c_2,\ldots,c_t$ are linearly independent from $\{c_1,c_{t+1},\ldots,c_r\}$. We then extend to a basis $\{c_1,\ldots,c_n\}$ arbitrarily. Define C as $C = [c_1,\ldots,c_n]$ (where the c_is are column vectors). Finally, define our desired matrix B as the product $B = DC$. We now verify the properties of B.

Property 1: Using Lemma 5.7 we have

$$\widehat{f_{DC}}(1) = \widehat{f_D}(Ce_1) = \widehat{f_D}(c_1) = \widehat{f_D}(1) \neq 0,$$

where the third equality used $c_1 = e_1$, and $\widehat{f_D}(1) \neq 0$ follows from the definition of D.

We next prove the following fact, which we use to verify the remaining three properties.

Fact 5.16. *Let C, D be invertible matrices as defined above. For every $i \in [t]$, let $(f_D^{(i)})_C$ be the function obtained after applying the invertible transformation C to $f_D^{(i)}$ and $(f_{DC})^{(i)}$ be the function obtained after fixing x_i to $(1 + sign(\widehat{f_{DC}}(i)))/2$ in f_{DC}. Then $(f_{DC})^{(i)} = (f_D^{(i)})_C$.*

Property 2: Fact 5.16 implies that $\mathrm{Fdim}((f_{DC})^{(i)}) = \mathrm{Fdim}((f_D^{(i)})_C)$. Since C is invertible, $\mathrm{Fdim}((f_D^{(i)})_C) = \mathrm{Fdim}(f_D^{(i)})$. From the choice of D, observe that for all $i \in \{2, \ldots, t\}$,

$$\mathrm{Fdim}(f_B^{(i)}) = \mathrm{Fdim}(f_{DC}^{(i)}) = \mathrm{Fdim}((f_D^{(i)})_C) = \mathrm{Fdim}(f_D^{(i)}) \leq \mathrm{Fdim}(f_D^{(1)}) = r - t,$$

where the inequality follows by definition of D.

Property 3: Note that $\mathrm{Fspan}(f_D^{(1)})$ is contained in $\mathrm{span}\{d_1, \ldots, d_{r-t}\}$ by construction. By making the invertible transformation by C, observe that $\mathrm{Fspan}((f_D^{(1)})_C) \subseteq \mathrm{span}\{e_{t+1}, \ldots, e_r\}$ (since for all $i \in [r-t]$, we defined $c_{t+i} = d_i$). Property 3 follows because $(f_D^{(1)})_C = f_{DC}^{(1)} = f_B^{(1)}$ by Fact 5.16.

Property 4: Using Fact 5.16, for every $\ell \in \{t+1, \ldots, r\}$, we have

$$\widehat{(f_B)^{(1)}}(\ell) = \widehat{(f_{DC})^{(1)}}(\ell) = \widehat{(f_D^{(1)})_C}(\ell) = \widehat{f_D^{(1)}}(c_\ell).$$

Since $c_\ell = d_{\ell-t}$, we have $\widehat{f_D^{(1)}}(c_\ell) = \widehat{f_D^{(1)}}(d_{\ell-t})$ and $\widehat{f_D^{(1)}}(d_1), \ldots, \widehat{f_D^{(1)}}(d_{r-t}) \neq 0$ by definition of d_i, hence the property follows.

Proof of Fact 5.16. Let $f_D = g$. We want to show that $(g^{(i)})_C = (g_C)^{(i)}$. For simplicity fix $i = 1$; the same proof works for every $i \in [t]$. Then,

$$(g^{(1)})(x) = \sum_{S \in \{0\} \times \{0,1\}^{n-1}} (\widehat{g}(S) + \widehat{g}(S \oplus e_1))\chi_S(x).$$

On transforming $g^{(1)}$ using the basis C we have:

$$(g^{(1)})_C(x) = \sum_{S\in\{0\}\times\{0,1\}^{n-1}} (\widehat{g}(CS) + \widehat{g}(C(S\oplus e_1))\chi_S(x). \tag{5.5}$$

Consider the function g_C. The Fourier expansion of g_C is $g_C(y) = \sum_{S\in\{0,1\}^n} \widehat{g}(CS)\chi_S(y)$ and the Fourier expansion of the $(g_C)^{(1)}$ can be written as

$$g_C^{(1)}(y) = \sum_{S\in\{0\}\times\{0,1\}^{n-1}} (\widehat{g}(CS) + \widehat{g}(CS\oplus Ce_1))\chi_S(y). \tag{5.6}$$

Using Eq. (5.5), (5.6), we conclude that $(g^{(1)})_C = (g_C)^{(1)}$, concluding the proof of the fact. □

This concludes the proof of the observation. □

This concludes the proof of the theorem. □

5.4 Proof of Theorem 5.4

Recall, from the introduction of this chapter, the refinement of Chang's lemma due to [38].

Theorem 5.17 ([38], Restatement of Theorem 5.3)**.** *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1-2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$r = O(\sqrt{\alpha k}\log k).$$

In this section we prove the tightness of the above theorem by proving Theorem 5.4, restated below.

Theorem 5.18 (Restatement of Theorem 5.4)**.** *For all $t \geq 2, t' \geq 4$, there exists a Boolean function f of Fourier sparsity $1+t^2(t'-1)$, Fourier dimension $(t\log t' + \log t')$ and satisfies $\widehat{f}(0^n) = 1 - 1/2t'$.*

Let us start with some definitions that will be useful for the rest of this section.

Recall that for any integer $n > 0$, the function $\mathsf{AND}_n : \{0,1\}^n \to \{-1,1\}$ is defined by $\mathsf{AND}_n(x) = -1$ if $x = (1)^n$, and 1 otherwise. We drop the subscript n when it is clear from the context. We state the Fourier expansion of AND below without proof.

Fact 5.19 (Fourier expansion of AND)**.** *Let $n \geq 1$ be any positive integer. Then*

$$\widehat{\mathsf{AND}_n}(S) = \begin{cases} 1 - \frac{2}{2^n} & S = \emptyset, \\ \frac{2\cdot(-1)^{|S|+1}}{2^n} & \textit{otherwise}. \end{cases}$$

The following observation follows directly from the above fact.

Observation 5.20. *Let $n \geq 4$ be a positive integers and let $\mathsf{AND}_n : \{0,1\}^{\log n} \to \{-1,1\}$. Then the Fourier dimension, Fourier sparsity, max-supp-entropy, max-rank-entropy and weight of $\mathsf{AND}_{\log n}$ are $\log n$, n, $n/2$, $n/2$ and $\frac{1}{n}$, respectively.*

Definition 5.21 (Indicator function)**.** *For any integer $n \geq 1$ and $b \in \{0,1\}^n$, define the function $\mathbb{I}_b : \{0,1\}^n \to \{0,1\}$ by*

$$\mathbb{I}_b(x) = \begin{cases} 1 & x = b, \\ 0 & \textit{otherwise}. \end{cases}$$

We require the following observation about the Fourier expansion of Indicator functions, which we state without proof.

Observation 5.22 (Fourier expansion of Indicator functions)**.** *For any integer $n \geq 1$ and $b \in \{0,1\}^n$, let $\mathbb{I}_b$ be as in Definition 5.21. Then,*

$$\widehat{\mathbb{I}}_b(S) = \frac{\prod_{i \in S}(-1)^{b_i}}{2^n} \quad \textit{for all } S \subseteq [n].$$

Definition 5.23 (Addressing function)**.** *For any integer $t \geq 2$, define the Addressing function $\mathsf{ADDR}_t : \{0,1\}^{\log t} \times \{0,1\}^t \to \{-1,1\}$ by*

$$\mathsf{ADDR}_t(x,y) = (-1)^{y_{bin(x)}},$$

where $x \in \{-1,1\}^{\log t}$ and $y \in \{-1,1\}^t$, and $bin(x)$ denotes the integer in $[t]$ whose binary representation is given by x. We refer to the x-variables as addressing variables*, and the y-variables as* target variables*.*

The following combinatorial observation is useful to us.

Observation 5.24. *For any integer $n \geq 1$ and non-empty subset $S \subseteq [n]$,*

$$\sum_{b \in \{0,1\}^n} \prod_{i \in S} (-1)^{b_i} = 0.$$

We require the following representation of Addressing functions.

Observation 5.25. *For any integer $t \geq 2$, $x \in \{0,1\}^{\log t}$ and $y \in \{0,1\}^t$, we have*

$$\mathsf{ADDR}_t(x,y) = \sum_{b \in \{-1,1\}^{\log t}} y_b \mathbb{I}_b(x).$$

We next define a way of modifying the Addressing function that is of use to us. In this modification, we replace target variables by functions, each acting on disjoint variables.

Definition 5.26 (Composed addressing functions)**.** *Let $t \geq 2$, $\ell_1, \ldots, \ell_t \geq 1$ be any integers. Let $g_i : \{0,1\}^{\ell_i} \to \{-1,1\}$ be any functions for $i \in [t]$. Define the function $\mathsf{ADDR}_t \circ_{\text{target}} (g_1, \ldots, g_t) : \{0,1\}^{\log t} \times \{0,1\}^{\ell_1 + \cdots + \ell_t} \to \{-1,1\}$ by*

$$\mathsf{ADDR}_t \circ_{\text{target}} (g_1, \ldots, g_t)(x, y_1, \ldots, y_t) = \mathsf{ADDR}_t(x, (1 - g_1(y_1))/2, \ldots, (1 - g_t(y_t))/2),$$

where $x \in \{0,1\}^{\log t}$ and $y_i \in \{0,1\}^{\ell_i}$ for all $i \in [t]$.

For any function $g : \{0,1\}^s \to \{-1,1\}$, we use the notation $\mathsf{ADDR}_t \circ_{\text{target}} g$ to denote the function $\mathsf{ADDR}_t \circ_{\text{target}} (g, g, \ldots, g) : \{0,1\}^{\log t} \times \{0,1\}^{ts} \to \{-1,1\}$.

Composed addressing functions are very well-behaved with respect to rank, sparsity, max-supp-entropy and max-rank-entropy. This is captured in Lemma 5.28 and is the main technical content of this section.

Definition 5.27 (AND-Target-Addressing Function)**.** *For any integers $t, t' \geq 2$, define the function $\mathsf{ADDR}_{t,t'} : \{0,1\}^{\log t} \times \{0,1\}^{t \log t'} \to \{-1,1\}$ by*

$$\mathsf{ADDR}_{t,t'} = \mathsf{ADDR}_t \circ_{\text{target}} \mathsf{AND}_{\log t'}.$$

5.4.1 Proof of tightness of Chang's lemma

We now prove that $\mathsf{ADDR}_{t,t'}$ is the desired function of this section, i.e. $\mathsf{ADDR}_{t,t'}$ witnessed tightness of Theorem 5.3 ([38]). We prove this in the next lemma, Lemma 5.28. The proof Lemma 5.28 relies on Lemma 5.29, which prove after the proof of Lemma 5.28.

Lemma 5.28 (Properties of $\mathsf{ADDR}_{t,t'}$)**.** *Fix any integers $t \geq 2$, $t' > 4$ let $\mathsf{ADDR}_{t,t'} : \{0,1\}^{\log t} \times \{0,1\}^{t \log t'} \to \{-1,1\}$ be as in Definition 5.27 and let $r(\mathsf{ADDR}_{t,t'}), k(\mathsf{ADDR}_{t,t'}), k'(\mathsf{ADDR}_{t,t'}), k''(\mathsf{ADDR}_{t,t'})$ and $\delta(\mathsf{ADDR}_{t,t'})$ denote the Fourier dimension, Fourier sparsity, max-entropy, max-rank-entropy and wight of $\mathsf{ADDR}_{t,t'}$, respectively. Then,*

- $r(\mathsf{ADDR}_{t,t'}) = t \log t' + \log t$,
- $k(\mathsf{ADDR}_{t,t'}) = 1 + t^2(t'-1)$,
- $k'(\mathsf{ADDR}_{t,t'}) = k''(\mathsf{ADDR}_{t,t'}) = \frac{tt'}{2}$, *and*
- $\delta(\mathsf{ADDR}_{t,t'}) = \frac{1}{t'}$.

Proof. Recall from Definition 5.27 that $\mathsf{ADDR}_{t,t'} = \mathsf{ADDR} \circ_{\text{target}} \mathsf{AND}$ where AND is on $\log t'$ bits. Since $t' > 4$, by Observation 5.20, $|\widehat{\mathsf{AND}}(\emptyset)| = 1 - \frac{2}{t'} > \frac{2}{t'} = |\widehat{\mathsf{AND}}(S)|$ for all $S \neq \emptyset$. Therefore the claim follows by Lemma 5.29 and Observation 5.20. □

We now state and prove Lemma 5.29.

Lemma 5.29 (Composition lemma)**.** *Let $t \geq 2, m \geq 1$ be any positive integers, and let $g : \{0,1\}^m \to \{-1,1\}$ be a non-constant function such that there exists a non-empty set $S \subseteq [m]$ with $0 \neq |\widehat{g}(S)| \leq |\widehat{g}(\emptyset)|$. Let $f : \{0,1\}^{\log t + mt} \to \{-1,1\}$ be defined as*

$$f = \mathsf{ADDR}_t \circ_{\text{target}} g.$$

Then

$$r(f) = t \cdot r(g) + \log t, \tag{5.7}$$
$$k(f) = 1 + t^2(k(g) - 1), \tag{5.8}$$
$$k'(f) = t \cdot k'(g), \tag{5.9}$$
$$k''(f) = t \cdot k''(g), \tag{5.10}$$
$$\delta(f) = \delta(g). \tag{5.11}$$

Proof of Composition lemma (Lemma 5.29)

Let $t \geq 2$ and $m \geq 1$ be any integers. For the purpose of the following proof, we introduce the following notation. For any $b \in \{0,1\}^{\log t}$, $T \subseteq [\log t]$ and non-empty $S_b \subseteq [m]$, define characters $\chi_{b,S_b,T} : \{0,1\}^{\log t} \times \{0,1\}^{tm} \to \{-1,1\}$ by $\chi_{b,S_b,T}(x,z) = \prod_{j \in S_b} (-1)^{z_{b,j}} \prod_{i \in T} (-1)^{x_i}$. Here $z = (\dots, z_b, \dots)$, where $b \in \{0,1\}^{\log t}$ and $z_b \in \{0,1\}^m$ for all $b \in \{0,1\}^{\log t}$.

Proof of Lemma 5.29. Let $x \in \{0,1\}^{\log t}$ and $z \in \{0,1\}^{tm}$. By Definition 5.26 and Observation 5.25,

$$\begin{aligned}
f(x,z) &= \sum_{b\in\{-1,1\}^{\log t}} g(z_b)\cdot \mathbb{I}_b(x) \\
&= \sum_b \left(\sum_{S_b\subseteq[m]} \widehat{g}(S_b)\chi_{S_b}(z_b)\right)\left(\sum_{T\subseteq[\log t]} \widehat{\mathbb{I}}_b(T)\chi_T(x)\right) \\
&= \sum_b (\widehat{g}(\emptyset)\widehat{\mathbb{I}}_b(\emptyset) + \widehat{g}(\emptyset)\sum_{T\neq\emptyset}\widehat{\mathbb{I}}_b(T)\chi_T(x) + \sum_{S_b\neq\emptyset}\sum_T \widehat{g}(S_b)\widehat{\mathbb{I}}_b(T)\chi_{S_b}(z_b)\chi_T(x)| \\
&= \underbrace{\sum_b \widehat{g}(\emptyset)\widehat{\mathbb{I}}_b(\emptyset)}_{A_1} + \underbrace{\sum_b \widehat{g}(\emptyset)\sum_{T\neq\emptyset}\widehat{\mathbb{I}}_b(T)\chi_T(x)}_{A_2} + \underbrace{\sum_{b,S_b\neq\emptyset,T} \widehat{g}(S_b)\widehat{\mathbb{I}}_b(T)\chi_{S_b}(z_b)\chi_T(x)}_{A_3}.
\end{aligned} \tag{5.12}$$

By Observation 5.22,

$$\begin{aligned}
A_1 &= \sum_b \widehat{g}(\emptyset)\frac{1}{t} = \widehat{g}(\emptyset), \\
A_2 &= \widehat{g}(\emptyset)\sum_b\sum_{T\neq\emptyset}\frac{\prod_{i\in T} b_i}{t}\chi_T(x) \\
&= \widehat{g}(\emptyset)\sum_{T\neq\emptyset}\chi_T(x)\sum_b\frac{\prod_{i\in T} b_i}{t} = 0, && \text{by Observation 5.24} \\
A_3 &= \sum_b\sum_{S_b\neq\emptyset}\sum_T \frac{\widehat{g}(S_b)\cdot\prod_{i\in T} b_i}{t}\chi_{S_b}(z_b)\chi_T(x) \\
&= \sum_{b,S_b\neq\emptyset,T} c_{b,S_b,T}\cdot\chi_{b,S_b,T}(x,z),
\end{aligned}$$

where $|c_{b,S_b,T}| = \frac{|\widehat{g}(S_b)|}{t}$ for all $b\in\{0,1\}^{\log t}, T\subseteq[\log t]$ and non-empty $S_b\subseteq[m]$. From Equation (5.12) and the above expressions for A_1, A_2 and A_3, we obtain the following Fourier expansion for f.

$$f = \widehat{g}(\emptyset) + \sum_{b,\emptyset\neq S_b\in\mathrm{supp}(g),T} c_{b,S_b,T}\cdot\chi_{b,S_b,T}, \tag{5.13}$$

since $|c_{b,S_b,T}| = \frac{|\widehat{g}(S_b)|}{t}$, $c_{b,S_b,t}$ is non-zero iff $\widehat{g}(S_b)$ is non-zero. Therefore

$$\mathrm{supp}(f) = \{\chi_\emptyset\}\cup\left\{\chi_{b,S_b,T} | b\in\{-1,1\}^{\log t}, \emptyset\neq S_b\in\mathrm{supp}(g), T\subseteq[\log t]\right\}. \tag{5.14}$$

- Rank: Fix a Fourier basis $\mathscr{B}_g$ of g such that $\mathscr{B}_g \subseteq \mathrm{supp}(g)$ and a character $\chi_U \in \mathscr{B}_g$. Consider the set of characters

$$\mathscr{B}_f = \left\{\chi_{b,S_b,\emptyset} | b \in \{-1,1\}^{\log t}, S_b \in \mathscr{B}_g\right\} \cup \{\chi_{\mathbf{1},U,\{i\}} | i \in [\log t]\}.$$

  By Equation (5.14), $\mathscr{B}_f \subseteq \mathrm{supp}(f)$ and $\mathrm{supp}(f) \subseteq \mathrm{span}\{()\mathscr{B}_f)$. Therefore,

$$r(f) = |\mathscr{B}_f| = t|\mathscr{B}_g| + \log t = t \cdot r(g) + \log t.$$

- Sparsity: By Equation (5.14),

$$k(f) = |\mathrm{supp}(f)| = 1 + t^2(k(g)-1).$$

- Max-supp-entropy: Recall from Definition 5.8 that $k'(f)$ equals the smallest non-zero Fourier coefficient of f in absolute value. From the Fourier expansion of f given in Equation (5.13),

$$\begin{aligned} k'(f) &= \max\left\{\frac{1}{|\widehat{g}(\emptyset)|}, \max\left\{\frac{t}{|\widehat{g}(S)|} : \emptyset \neq S \in \mathrm{supp}(g)\right\}\right\} \\ &= \max\left\{\frac{t}{|\widehat{g}(S)|} : \emptyset \neq S \in \mathrm{supp}(g)\right\} \\ &\qquad\qquad \text{since } |\widehat{g}(\emptyset)| \geq \tfrac{1}{k'(g)} \text{ and } t \geq 1 \text{ by assumption} \\ &= t \cdot k'(g). \end{aligned}$$

- Max-rank-entropy: Recall from Definition 5.8 that $k''(f) = \mathrm{argmin}_\theta\{\dim(\mathscr{S}_\theta) = r(f)\}$, where $\mathscr{S}_\theta = \left\{S : |\widehat{f}(S)| \geq \frac{1}{\theta}\right\}$.

  From the Fourier expansion of f given in Equation (5.13), the following set $\mathscr{B}_f$ is a spanning set for the Fourier support of f. Let $\mathscr{B}_g$ be a Fourier basis for g such that $|\widehat{g}(S)| \geq \frac{1}{k''(g)}$ for all $S \in \mathscr{B}_g$. Define

$$\mathscr{B}_f = \left\{\chi_{b,S_b,T} : b \in \{-1,1\}^{\log t}, S_b \in \mathscr{B}_g, T \subseteq [\log t]\right\} \tag{5.15}$$

  One may verify that $\mathscr{B}_f$ indeed is a spanning set for $\mathrm{supp}(f)$. By Equation (5.13), $|c_{b,S_b,T}| = \frac{|\widehat{g}(S_b)|}{t}$ for all $b \in \{-1,1\}^{\log t}, T \subseteq [\log t]$ and non-empty $S_b \subseteq [m]$. Hence $k''(f) \leq t \cdot k''(g)$.

It now remains to show that $k''(f) \geq t \cdot k''(g)$. Towards a contradiction, consider a basis $T_f \subseteq \left\{\chi_{b,S_b,T} : b \in \{0,1\}^{\log t}, S_b \in \mathrm{supp}(g)\right\}$ for $\mathrm{supp}(f)$, with $|\widehat{f}(S)| > \frac{1}{t \cdot k''(g)}$ for all $S \in T_f$. Fix any $b \in \{0,1\}^{\log t}$. Observe that the set $\left\{\chi_{S_b} : \chi_{b,S_b,T} \in T_f\right\}$ forms a spanning set for $\mathrm{supp}(g)$. Moreover, since $|c_{b,S_b,T}| = \frac{|\widehat{g}(S_b)|}{t}$ for all $b \in \{0,1\}^{\log t}, T \subseteq [\log t]$ and non-empty $S_b \subseteq [m]$ by Equation (5.13), the set $\left\{\chi_{S_b} : \chi_{b,S_b,T} \in T_f\right\}$ is such that each of its Fourier coefficients (i.e. $\widehat{g}(S_b)$) has absolute value strictly larger than $\frac{1}{k''(g)}$, which is a contradiction by the definition of $k''(g)$.

- Weight: By Observation 5.5,

$$\delta(f) = \frac{1-\widehat{f}(\emptyset)}{2} = \frac{1-\widehat{g}(\emptyset)}{2} = \delta(g),$$

where the second equality follows by Equation (5.13).

□

5.5 Chang's lemma and quantum learning

In this section we give a potential direction to prove that in expectation $\tilde{O}(k)$ Fourier-samples are sufficient for Phase 1 of our learning algorithm presented in Section 3.3.1. Recall Phase 1 of our learning algorithm. Given a k-Fourier-sparse function c, Phase 1 starts by finding an $S \in \mathrm{supp}(\widehat{c})$ such that $S \neq 0^n$. Lemma 3.13 implies that an expected number of $O(k)$ many Fourier-samples are sufficient to sample such an S.

As we discussed in the introduction of this chapter, the Chang's lemma gives an improvement over Lemma 3.13 for the expected number of Fourier-samples sufficient to obtain an $S \in \mathrm{supp}(\widehat{c})$ such that $S \neq 0^n$ in Phase 1. We gave the following refinement of Chang;s lemma in this chapter:

Theorem 5.30 (Restatement of Theorem 5.2)**.** *Let $\alpha \in (0,1)$ and $k \geq 2$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ that satisfies $\widehat{f}(0^n) = 1-2\alpha$ and $\mathrm{Fdim}(f) = r$, we have*

$$\widehat{f}(0^n) \leq 1 - \frac{r}{k \log k}.$$

We remark that in a follow-up paper [38], a subset of the authors gave a refinement of the theorem above.

Let us discuss how the above theorem improves the analysis of Phase 1 of our learning algorithm. Theorem 5.2 implies that for a k-Fourier-sparse Boolean function $c : \{0,1\}^n \to$

$\{-1,1\}$ of Fourier dimension r,

$$\sum_{S:S\neq 0^n} \widehat{c}(S)^2 = \Omega(r/(k\log k)).$$

This is a better lower bound on the Fourier weight of c on the set $\{0,1\}^n \setminus \{0^n\}$ than that obtained from Chang's lemma (Equation 5.3). Thus an expected number of $O((k\log k)/r)$ many uniformly quantum samples is sufficient to obtain an $S \in \mathrm{supp}(\widehat{c})$ such that $S \neq 0^n$.

We suspect that Theorem 5.2 can in fact lead to an $\tilde{O}(k)$ learning algorithm for Phase 1. Towards that end we make the following conjecture which can be viewed as a generalization of Theorem 5.2.

Conjecture 5.31. *Let $n > 0$ and $1 \leq k \leq 2^n$. For every k-Fourier-sparse $f : \{0,1\}^n \to \{-1,1\}$ with Fourier span $\mathscr{V}$ and Fourier dimension r, the following holds: for every $r' > 0$ and $\mathscr{S} \subset \mathscr{V}$ satisfying $\dim(span(\mathscr{S})) = r'$, we have*

$$\sum_{S\in span(\mathscr{S})} \widehat{f}(S)^2 \leq 1 - \frac{r-r'}{k\log k}.$$

If the above conjecture is true then it would be imply an $\tilde{O}(k)$ learning algorithm for Phase 1. Let $c : \{0,1\}^n \to \{-1,1\}$ be a k-Fourier-sparse function of Fourier dimension r. Assuming Conjecture 5.31 to be true we have

$$\sum_{S\notin \mathrm{span}(\mathscr{S})} \widehat{c}(S)^2 \geq \frac{r-r'}{k\log k}.$$

So the expected number of samples to increase the dimension by 1 is $\leq \frac{k\log k}{r-r'}$. Accordingly, the expected number of Fourier-samples needed to learn the whole Fourier span of f is at most

$$\sum_{i=1}^{r} \frac{k\log k}{i} \leq O(k\log k\log r),$$

where the final inequality used $\sum_{i=1}^{r} \frac{1}{i} = O(\log r)$.

5.6 Summary

In this chapter we considered Chang's lemma for Boolean functions and gave a refinement of this lemma. We showed that the Fourier dimension is bounded by at most (roughly) weight times Fourier sparsity, a quadratic improvement over the previously best known bound. For

functions with lower weight, [38] gave a further refinement. We proved tightness of their refinement in this chapter. Finally, we showed how Chang's lemma is connected to quantum learning and gave a possible direction to improve the first phase of our learning algorithm from Chapter 3.

Part II

Quantum Query and Communication Complexity

Chapter 6

Overhead in Query-to-Communication Simulation for XOR Functions

6.1 Introduction

Classical communication complexity, introduced by Yao [154], is aptly called the 'swiss-army-knife' for understanding, especially the limitations of, classical computing. Communication complexity has important applications in several disciplines, in particular for lower bounds on circuits, data structures, streaming algorithms, and many other complexity measures (see, for example, [103] and the references therein). Quantum communication complexity, also introduced by Yao [155], holds the same promise with regards to quantum computing. Yet, there are many problems that remain open. One broad theme is to understand the fundamental differences between classical randomized and quantum protocols, especially for computing total functions.

Before we proceed, let us note that Boolean functions in this part of the thesis will be denoted by:

$$f : \{-1,1\}^n \to \{-1,1\}.$$

We also consider Boolean functions where input can be seen to be naturally consisting of two parts, in this case we view functions as:

$$f : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}.$$

The reason for such a choice of notation is that we heavily study composition of Boolean functions. The above choice of notation makes the technical content simpler notation wise and more intuitive.

A natural way to derive a communication problem from a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ is via composition. Let $f : \{-1,1\}^n \to \{-1,1\}$ be a function and let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a "two-party function". Then $F = f \circ G : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ denotes the function corresponding to the communication problem in which Alice is given input $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$, Bob is given $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$, and their task is to compute $F(X,Y) = f(G(X_1,Y_1), \ldots, G(X_n,Y_n))$. We refer to f as the *outer function* and G as the *inner function*.

Many well-known functions in communication complexity are derived in this way, such as Set-Disjointness ($\mathsf{DISJ}_n := \mathsf{NOR}_n \circ \mathsf{AND}_2$), Inner Product ($\mathsf{IP}_n := \mathsf{PARITY}_n \circ \mathsf{AND}_2$) and Equality ($\mathsf{EQ}_n := \mathsf{NOR}_n \circ \mathsf{XOR}_2$).

A well studied question is this regard is what is the relationship between the query complexity of f and the communication problem of $f \circ G$, when the G is AND_2 or XOR_2. This question has been studied for particular interesting functions or special classes of functions. Classically, it is folklore that

$$R^{cc}(f \circ G) \leq 2R(f),$$

where $R(f)$ denotes the bounded-error randomized query complexity of f and $R^{cc}(f \circ G)$ denotes the bounded-error randomized communication complexity for computing $f \circ G$. In an influential work, Buhrman, Cleve and Wigderson [34] observed that a general and natural recipe exists for constructing a quantum communication protocol for $f \circ G$, using a quantum query algorithm for f as a black-box.

Theorem 6.1 ([34]). *For any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, we have*

$$Q^{cc}(f \circ G) = O\big(Q(f) \cdot \log n\big),$$

where G is either AND_2 or XOR_2.

Here $Q(f)$ denotes the bounded-error quantum query complexity of f, and $Q^{cc}(f \circ G)$ denotes the bounded-error quantum communication complexity for computing $f \circ G$. Thus in the quantum world one incurs a logarithmic factor in the natural BCW simulation while no such factor is needed in the randomized setting. The basic question that arises naturally and which we completely answer in this work, is the following: analogous to the classical

model, can this multiplicative $\log n$ blow-up in the communication cost be always avoided by designing quantum communication protocols that more cleverly simulate quantum query algorithms?

A priori, it is not clear what the answer to this question ought to be. For certain special functions and some classes of functions, quantum protocols exist where the $\log n$ factor can be saved. First, Høyer and de Wolf [87] designed a quantum protocol for Set-Disjointness of cost $O(\sqrt{n}c^{\log^* n})$, speeding up the BCW simulation significantly. Later, Aaronson and Ambainis [1] gave a more clever protocol that only incurred a constant factor overhead from Grover's search using more involved ideas.

For partial functions, tightness of the BCW simulation is known in *some* settings. For example, consider the Deutsch-Jozsa (DJ) problem, where the input is an n-bit string with the promise that its Hamming weight is either 0 or $n/2$, and DJ outputs -1 if the Hamming weight is $n/2$, and 1 otherwise. DJ has quantum query complexity 1 whereas the *exact* quantum communication complexity of $(\mathsf{DJ} \circ \mathsf{XOR}_2)$ is $\log n$. Note that it is unclear whether the $\log n$ factor loss here is additive or multiplicative.[1] Montanaro, Nishimura and Raymond [115] exhibited a partial function for which the BCW simulation is tight (up to constants) in the *exact* and *non-deterministic* quantum settings. They also observed the existence of a total function for which the BCW simulation is tight (up to constants) in the *unbounded-error* setting.

To the best of our knowledge, there was no (partial or total) Boolean-valued function f known prior to our work for which the *bounded-error* quantum communication complexity of $f \circ G$ (i.e. $Q^{cc}(f \circ G)$) is even $\omega(Q(f))$, where G is either AND_2 or XOR_2. In this chapter construct a total function that show that $\log n$ overhead is required when the inner function $G = \mathsf{AND}_2$. In next chapter we show that the this overhead is required even when $G = \mathsf{AND}_2$ and also give a general recipe for constructing such out functions.

6.1.1 Results and Organization

In this chapter, we exhibit the first *total function* witnessing the tightness of the BCW simulation in arguably the most well-known quantum model, which is the bounded-error model.

[1] Indeed, there are well-known situations where complexity of 1 vs. $\log n$ can be deceptive. The classical private-coin randomized communication complexity of Equality is $\Theta(\log n)$, whereas the public-coin cost is well known to be $O(1)$. Newman's Theorem shows that this difference in costs, in general, is *not multiplicative* but merely *additive*.

Theorem 6.2. *There exists a total function $F : \{-1,1\}^n \to \{-1,1\}$ for which,*

$$Q^{cc}(F \circ \mathsf{XOR}_2) = \Theta(Q(F)\log n). \tag{6.1}$$

The statement above does not necessarily guarantee that a function exists that both satisfies Equation 6.1 and has bounded-error quantum query complexity (as a function of n) arbitrarily close to n. We answer this question by proving a more general result, from which Theorem 6.2 follows.

Theorem 6.3 (Main Theorem)**.** *For any constant $0 < \delta < 1$, there exists a total function $F : \{-1,1\}^n \to \{-1,1\}$ for which $Q(F) = \Theta(n^\delta)$ and*

$$Q^{cc}(F \circ \mathsf{XOR}_2) = \Theta(Q(F)\log n).$$

In Section 6.3 we give overview of our approaches and techniques and intuition behind the construction of our function. In Section 6.4 we give the proof our the main theorem. Finally, we conclude with summary and future work.

Some other implications of our result

Zhang [156] showed that for all Boolean functions f, there must exists gadgets g_i, each either AND_2 or OR_2, such that $Q^{cc}(f(g_1,\ldots,g_n)) = \Omega(\mathrm{poly}(Q(f)))$. For monotone f, they showed that either $Q^{cc}(f \circ \mathsf{AND}_2) = \Omega(\mathrm{poly}(Q(f)))$ or $Q^{cc}(f \circ \mathsf{OR}_2) = \Omega(\mathrm{poly}(Q(f)))$. They also state that it is unclear how tight the BCW simulation is. Our result implies that there exists a function for which it is tight up to constants (on composition with XOR_2).

Another implication of our result is related to the Entropy Influence Conjecture, which is an interesting question in the field of analysis of Boolean functions, posed by Friedgut and Kalai [62]. This conjecture is wide open for general functions. A much weaker version of this conjecture is called the Min-Entropy Influence Conjecture. For the statement of the conjecture we need to consider the Fourier expansion of Boolean functions $f : \{-1,1\}^n \to \{-1,1\}$ as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x),$$

where $\{\chi_S : S \subseteq [n]\}$ are the *parity* functions ($\chi_S(x) = \Pi_{i \in S} x_i$, when $x = (x_1,\ldots,x_n) \in \{-1,1\}^n$) and $\{\widehat{f}(S) : S \subseteq [n]\}$ are the corresponding Fourier coefficients.

Conjecture 6.4. *(Min-Entropy Influence Conjecture) For any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ there exists a non-zero Fourier coefficient $\widehat{f}(S)$ such that*

$$\log\left(1/|\widehat{f}(S)|\right) = O(I(f)),$$

where $I(f)$ denotes the influence (or average sensitivity) of f ($I(f) = \sum_{S\subseteq[n]} |S|\widehat{f}(S)^2$).

While this conjecture is also wide open, some attempts have been made to prove various implications of this conjecture. One interesting implication of the Min-Entropy Influence Conjecture that is still open is whether the min-entropy of the Fourier spectrum (that is, $\log\left(1/\max_{S\subseteq[n]}|\widehat{f}(S)|\right)$) is less than $O(Q(f))$. In [13] using a primal-dual technique it was shown that the min-entropy of the Fourier spectrum is less than a constant times $\log(\|\hat{f}\|_{1,\varepsilon})$, where the constant depends on ε. Thus if it were the case that $\log(\|\hat{f}\|_{1,\varepsilon}) = O(Q(f))$, we would have upper bounded the min-entropy of Fourier spectrum by $O(Q(f))$. This was stated in [13] as a possible approach and was left as an open problem. While their conjecture holds for certain special class of functions like the symmetric functions (proof given in the Section 6.2.4), our result in this chapter nullifies this approach for general Boolean functions.

6.2 Preliminaries

In this section we review the necessary preliminaries and prove some basic facts.

For any positive integer n, we denote the set $\{1,\ldots,n\}$ by $[n]$. For $d \leq n$ we use the notation $\binom{n}{\leq d} := \binom{n}{0} + \cdots + \binom{n}{d}$. Note that $\binom{n}{\leq d} < (n+1)^d$.

We present some basic notions of Fourier analysis on the Boolean cube when the domain of function is $\{-1,1\}^n$. This only changes the characters and the Fourier spectrum remains unchanged. Consider the vector space of functions from $\{-1,1\}^n$ to $\mathbb{R}$, equipped with an inner product defined by

$$\langle f,g\rangle := \mathbb{E}_{x\in\{-1,1\}^n}[f(x)g(x)] = \frac{1}{2^n}\sum_{x\in\{-1,1\}^n} f(x)g(x)$$

for every $f,g : \{-1,1\}^n \to \mathbb{R}$. For any set $S \subseteq [n]$, define the associated *parity* function χ_S by $\chi_S(x) = \prod_{i\in S} x_i$. The set of parity functions $\{\chi_S : S \subseteq [n]\}$, forms an orthonormal basis for this vector space. Thus, every function $f : \{-1,1\}^n \to \mathbb{R}$ has a unique multilinear expression as $f = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S$. The coefficients $\{\widehat{f}(S) : S \subseteq [n]\}$ are called the *Fourier coefficients* of f.

Fact 6.5 (Parseval's Identity)**.** *For any function $f : \{-1,1\}^n \to \mathbb{R}$, we have $\sum_{S\subseteq[n]} \widehat{f}(S)^2 = \frac{\sum_{x\in\{-1,1\}^n} f(x)^2}{2^n}$. The degree of f is the degree of the unique polynomial representing f.*

Definition 6.6 (Spectral Norm)**.** *For any function $f : \{-1,1\}^n \to \mathbb{R}$, define its* spectral norm*, which we denote $\|\widehat{f}\|_1$, to be the sum of absolute values of the Fourier coefficients of f. That is, $\|\widehat{f}\|_1 := \sum_{S\subseteq[n]} \left|\widehat{f}(S)\right|$.*

Definition 6.7 (Hadamard Codeword)**.** *If an ℓ-bit string $(x_1,\ldots,x_\ell) \in \{-1,1\}^\ell$ (alternatively, view the indices of x as subsets of $[\log \ell]$) is of the form $x_S = \prod_{i\in S} z_i$ for all $S \subseteq [\log \ell]$ for some $z \in \{-1,1\}^{\log \ell}$, then define such an $x = x_1 \ldots x_\ell$ to be the ℓ-bit* Hadamard codeword *$h(z)$ of the $(\log \ell)$-bit string z.*

6.2.1 Addressing functions

Definition 6.8 ((m,k)-addressing function)**.** *We define a (partial) function $f : \{-1,1\}^{m+k} \to \{-1,1,\star\}$ to be an (m,k)-*addressing function *if there exists $g : \{-1,1\}^m \to \{[k] \cup \star\}$ such that*

- *$f(x_1,\ldots,x_m,y_1,\ldots,y_k) = y_{g(x_1,\ldots,x_m)}$ if $g(x_1,\ldots,x_m) \in [k]$, and $f(x_1,\ldots,x_m,y_1,\ldots,y_k) = \star$ otherwise.*
- *For all $j \in [k]$, there exists $(x_1,\ldots,x_m) \in \{-1,1\}^m$ such that $g(x_1,\ldots,x_m) = j$.*

We call the variables $\{x_1,\ldots,x_m\}$ the address variables *and the variables $\{y_1,\ldots,y_k\}$ the* target variables*. The function g is called the* selector *function of f.*

Definition 6.9 (Indexing Function)**.** *The Indexing function, which we denote by IND_k, is a $(k,2^k)$-addressing function defined by $\mathsf{IND}(x_1,\ldots,x_k,y_1,\ldots,y_{2^k}) = y_{\mathrm{bin}(x)}$, where $\mathrm{bin}(x)$ denotes the integer represented by the binary string $x_1,\ldots,x_k$.*

Definition 6.10 (Composition with addressing functions)**.** *For any function $f : \{-1,1\}^n \to \{-1,1\}$ and an (m,k)-addressing function ADDR, define the (partial) function $f^{\mathsf{ADDR}} : \{-1,1\}^{n(m+k)} \to \{-1,1,\star\}$ by*

$$f^{\mathsf{ADDR}}(x_1,y_1,\ldots,x_n,y_n) = \begin{cases} f(\mathsf{ADDR}(x_1,y_1),\ldots,\mathsf{ADDR}(x_n,y_n)) & \text{if } \forall i \in [n], \mathsf{ADDR}(x_i,y_i) \in \{-1,1\} \\ \star & \text{otherwise.} \end{cases}$$

where $x_i \in \{-1,1\}^m$ and $y_i \in \{-1,1\}^k$ for all $i \in [n]$.

Definition 6.11 (Hadamard Addressing Function)**.** *We define the* Hadamard addressing function*, which we denote $\mathsf{HADD}_\ell : \{-1,1\}^{2\ell} \to \{-1,1,\star\}$, as follows. Fix an arbitrary*

order on the ℓ-many Hadamard codewords of $(\log \ell)$-bit strings, say $w_1, \ldots, w_\ell$. Define the selector function of HADD_ℓ by

$$g(x) = \begin{cases} i & \text{if } x = w_i \text{ for some } i \in [\ell] \\ \star & \text{otherwise.} \end{cases}$$

Note that HADD_ℓ is an (ℓ, ℓ)-addressing function.

6.2.2 Polynomial approximation

Definition 6.12 (Approximate Degree)**.** *The* ε-approximate degree *of a function $f : \{-1,1\}^n \to \{-1,1,\star\}$, denoted by $\widetilde{\deg}_\varepsilon(f)$ is defined to be the minimum degree of a real polynomial $p : \{-1,1\}^n \to \mathbb{R}$ that satisfies $\left|p(x) - f(x)\right| \leq \varepsilon$ for all $x \in \{-1,1\}^n$ for which $f(x) \in \{-1,1\}$.*[2] *That is,*

$$\widetilde{\deg}_\varepsilon(f) := \min\{d : \deg(p) \leq d, \left|p(x) - f(x)\right| \leq \varepsilon \text{ for all } x \in \{-1,1\}^n \text{ for which } f(x) \in \{-1,1\}\}.$$

Henceforth, we will use the notation $\widetilde{\deg}(f)$ to denote $\widetilde{\deg}_{1/3}(f)$.

Definition 6.13 (Approximate Spectral Norm)**.** *The* approximate spectral norm *of a function $f : \{-1,1\}^n \to \{-1,1,\star\}$, denoted by $\|\widehat{f}\|_{1,\varepsilon}$ is defined to be the minimum spectral norm of a real polynomial $p : \{-1,1\}^n \to \mathbb{R}$ that satisfies $\left|p(x) - f(x)\right| \leq \varepsilon$ for all $x \in \{-1,1\}^n$ for which $f(x) \in \{-1,1\}$.*

$$\|\widehat{f}\|_{1,\varepsilon} := \min\{\|\widehat{p}\|_1 : \left|p(x) - f(x)\right| \leq \varepsilon \text{ for all } x \in \{-1,1\}^n \text{ for which } f(x) \in \{-1,1\}\}.$$

Lemma 6.14 ([35])**.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a total function. Then for all constants $0 < \delta, \varepsilon < 1$ we have $\widetilde{\deg}_\varepsilon(f) = \Theta(\widetilde{\deg}_\delta(f))$.*

The following is a standard upper bound on the approximate spectral norm of a Boolean function in terms of its approximate degree.

Claim 6.15. *For all total functions $f : \{-1,1\}^n \to \{-1,1\}$, we have $\log \|\widehat{f}\|_{1,1/3} = O(\widetilde{\deg}(f) \log n)$.*

Proof. Let d denote the approximate degree of f. Take any 1/3-approximating polynomial of degree d, say p, to f. Then, $\sum_{S \subseteq [n]} \left|\widehat{p}(S)\right| \leq \sqrt{\binom{n}{\leq d}} \cdot \sqrt{\sum_{S : |S| \leq d} \widehat{p}(S)^2} \leq 4/3 \cdot (n+1)^{d/2} =$

[2] When dealing with partial functions, another notion of approximation is sometimes considered, where the approximating polynomial p is required to have bounded values even on the non-promise inputs of f. For the purpose of this paper, we do not require this constraint.

$2^{O(d \log n)}$, where the first inequality follows by the Cauchy-Schwarz inequality, the second inequality follows by Parseval's identity (Fact 6.5) and the fact that the absolute value of p is at most $4/3$ for any input $x \in \{-1,1\}^n$. □

It is easy to exhibit functions $f : \{-1,1\}^n \to \{-1,1\}$ such that $\log \|\widehat{f}\|_{1,1/3} = \Omega(\widetilde{\deg}(f))$. Bent functions satisfy this bound, for example.

Building upon ideas in [100], the approximate spectral norm of $f \circ \mathsf{IND}_1$ was shown to be bounded below by $2^{\Omega(\widetilde{\deg}(f))}$ in [42].

Theorem 6.16 ([42]). *Let $f : \{-1,1\}^n \to \{-1,1\}$ be any function. Then $\|\widehat{f \circ \mathsf{IND}_1}\|_{1,1/3} \geq 2^{c \cdot \widetilde{\deg}_{2/3}(f)}$ for any constant $c < 1 - 3/\widetilde{\deg}_{2/3}(f)$.*

We now recall the model of communication complexity.

6.2.3 Communication complexity

We briefly recall the model of classical communication complexity. The classical model of communication complexity was introduced by Yao in [154]. In this model two parties, say Alice and Bob, wish to compute a function whose output depends on both their inputs. Alice is given an input $x \in \mathscr{X}$, Bob is given $y \in \mathscr{Y}$, and they want to jointly compute the value of a given function $F(x,y)$ by communicating with each other. Alice and Bob individually have unbounded computational power and the number of bits communicated is the resource we wish to minimize. Alice and Bob communicate using a *protocol* that is agreed upon in advance. In the randomized model, Alice and Bob have access to unlimited public random bits and the goal is to compute the correct value of $F(x,y)$ with probability at least $2/3$ for all inputs $(x,y) \in \mathscr{X} \times \mathscr{Y}$. The *bounded-error randomized communication complexity* of a function F, denoted $R^{cc}(F)$, is the number of bits that must be communicated in the worst case by any randomized protocol to compute the correct value of the function $F(x,y)$, with probability at least $2/3$, for every $(x,y) \in \mathscr{X} \times \mathscr{Y}$.

The quantum model of communication complexity was introduced by Yao in [155]. We refer the reader to the survey [50] for details. The *bounded-error quantum communication complexity* of a function F, denoted $Q^{cc}(F)$ is the number of bits that must be communicated by any quantum communication protocol in the worst case to compute the correct value of the function $F(x,y)$, with probability at least $2/3$, for every (x,y) in domain of F. Buhrman, Cleve and Wigderson [34] observed a quantum simulation theorem, which gives an upper bound on the bounded-error quantum communication complexity of a composed function of

the form $f \circ \mathsf{AND}_2$ or $f \circ \mathsf{XOR}_2$ in terms of the bounded-error quantum query complexity of f (see Theorem 6.1).

Lee and Shraibman [104] showed that the bounded-error quantum communication complexity of $f \circ \mathsf{XOR}_2$ is bounded below by the logarithm of the approximate spectral norm of f. Also see [42] for an alternate proof.

Theorem 6.17 ([104]). *For any Boolean function $f : \{-1,1\}^n \to \{-1,1\}$,*

$$Q^{cc}(f \circ \mathsf{XOR}_2) = \Omega(\log \|\widehat{f}\|_{1,1/3}).$$

6.2.4 Approximate spectral norm of symmetric functions

Definition 6.18 (Multilinear Polynomial). *A function $\phi : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial if ϕ is of the form:*

$$\phi(x_1,\ldots,x_n) = \sum_{S\subseteq[n]} a_S \prod_{i\in S} x_i$$

where $a_S \in \mathbb{R}$.

Definition 6.19 (Spectral Norm of a Multilinear Polynomial). *Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a multilinear polynomial of the form $\phi(x_1,\ldots,x_n) = \sum_{S\subseteq[n]} a_S \prod_{i\in S} x_i$. The spectral norm of ϕ, denoted by $\|\phi\|_1$, is defined as*

$$\|\phi\|_1 = \sum_{S\subseteq[n]} |a_S|.$$

Fact 6.20 (Properties of Spectral Norm of Multilinear Polynomials). *Let $f,g : \mathbb{R}^n \to \mathbb{R}$ be any symmetric polynomials and let $\alpha \in \mathbb{R}$ be any real number. Then,*

1. $\|\alpha f\|_1 = |\alpha|\|f\|_1$,
2. $\|f+g\|_1 \leq \|f\|_1 + \|g\|_1$,
3. $\|fg\|_1 \leq \|f\|_1\|g\|_1$.

Lemma 6.21. *Let $S \subseteq [n]$ and $\chi_S : \{-1,1\}^n \to \mathbb{R}$ be the symmetric multilinear polynomial defined as*

$$\chi_S(x_1,\ldots,x_n) = \prod_{i\in S} \frac{(1-x_i)}{2}.$$

Then $\|\chi_S\|_1 = 1$.

Proof. Since for all $i \in [n]$, the spectral norm of $\frac{(1-x_i)}{2} = 1$, the proof follows from 6.20 (3). □

Definition 6.22 (Symmetric Multilinear Polynomial)**.** *A multilinear polynomial $\phi : \mathbb{R}^n \to \mathbb{R}$ is said to be symmetric if $\phi(x_1,\ldots,x_n) = \phi(x_{\sigma(1)},\ldots,x_{\sigma(n)})$ for all $(x_1,\ldots,x_n) \in X$ and $\sigma \in S_n$.*

Sherstov [138] showed the following upper bound on the spectral norm of symmetric multilinear polynomials.

Claim 6.23 ([138])**.** *Let $\phi : \mathbb{R}^n \to \mathbb{R}$ be a symmetric multilinear polynomial. Then*

$$\|\phi\|_1 \leq 8^{\deg(\phi)} \max_{x\in\{0,1\}^n} |\phi(x)|.$$

Lemma 6.24. *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a symmetric Boolean function. Then*

$$\log(\|f\|_{1,1/3}) = O(\widetilde{\deg}(f)).$$

Proof. Let $f' : \{0,1\}^n \to \{-1,1\}$ be defined as $f'(x_1,\ldots,x_n) = f\left(\frac{1-x_1}{2},\ldots,\frac{1-x_n}{2}\right)$. It is not hard to show, since we have done a linear transformation on the input domain, that $\widetilde{\deg}(f') = \widetilde{\deg}(f)$. Let p' be a polynomial that $1/3$-approximates the symmetric function f'. By symmetrization we can assume that p' is symmetric, and is of the form $p'(x) = \sum_{S\subseteq[n]} a_S \prod_{i\in S} x_i$.

Define the polynomial $p : \{-1,1\}^n \to \mathbb{R}$ as follows:

$$p(x_1,\ldots,x_n) = p'\left(\frac{1-x_1}{2},\ldots,\frac{1-x_n}{2}\right).$$

Clearly p is a $1/3$-approximation to f since p' is a $1/3$-approximation to f' and we can write

$$p(x) = \sum_{S\subseteq[n]} a_S \prod_{i\in S} \frac{(1-x_i)}{2}.$$

Thus we can upper bound the ℓ_1-norm of p as follows:

$$\|p\|_1 = \left\| \sum_{S\subseteq[n]} a_S \prod_{i\in S} \frac{(1-x_i)}{2} \right\|_1 \leq \sum_{S\subseteq[n]} \left\| a_S \prod_{i\in S} \frac{(1-x_i)}{2} \right\|_1 \tag{6.2}$$

$$\leq \sum_{S\subseteq[n]} |a_S| \left\| \prod_{i\in S} \frac{(1-x_i)}{2} \right\|_1 \tag{6.3}$$

$$= \sum_{S\subseteq[n]} |a_S| = \|p'\|_1, \tag{6.4}$$

where Equation (6.2) follows from Fact 6.20 (2), Equation (6.3) follows from Fact 6.20 (1) and Equation (6.4) follows from 6.21.

Hence, $\log(\|f\|_{1,1/3}) \leq \log(\|p\|_1) \leq \log(\|p'\|_1) = O(\deg(p'))$, where the last equality follows by Claim 6.23 since p' is symmetric. Since p' was assumed to have degree $\deg(p') = \widetilde{\deg}(f)$, the lemma follows. □

6.3 Overview of our approach and techniques

To demonstrate the tightness of the BCW simulation for a total function in the quantum bounded-error setting we have to find a function F such that $Q^{cc}(F \circ G) = \Theta(Q(F)\log n)$ for some choice of G (that is, either G is AND_2 or XOR_2). This requires us to prove an upper bound of $Q(F)$ and a lower bound on $Q^{cc}(F \circ G)$. We consider the case when G is the XOR_2 function.

For the inner function the XOR_2 function is preferred over the AND_2 function for one crucial reason: we have an analytical technique for proving lower bounds on $Q^{cc}(F \circ \mathsf{XOR}_2)$, due to Lee and Shraibman [104]. They reduced the problem of lower bounding the bounded-error quantum communication complexity of $(F \circ \mathsf{XOR}_2)$ to proving lower bounds on an analytic property of F, called its *approximate spectral norm*. The ε-approximate spectral norm of F, denoted by $\|\hat{F}\|_{1,\varepsilon}$, is defined to be the minimum ℓ_1-norm of the coefficients of a polynomial that approximates F uniformly to error ε (see Definition 7.31). Lee and Shraibman [104] showed that $Q^{cc}(F \circ \mathsf{XOR}_2) = \Omega(\log \|\widehat{F}\|_{1,1/3})$. Thus, the lower bound of Theorem 6.3 follows immediately from our result below.

Theorem 6.25. *For any constant $0 < \delta < 1$, there exists a total function $F : \{-1,1\}^n \to \{-1,1\}$ for which $Q(F) = \Theta(n^\delta)$ and*

$$\log\left(\|\widehat{F}\|_{1,1/3}\right) = \Theta(Q(F)\log n).$$

There are not many techniques known to bound the approximate spectral norm of a function. This sentiment was expressed both in [104] and in the work of Ada, Fawzi and Hatami [2]. On the other hand, classical approximation theory offers tools to prove bounds on a simpler and better known concept called *approximate degree* which has been invaluable, particularly for quantum query complexity. The ε-approximate degree of f, denoted by $\widetilde{\deg}_{\varepsilon}(f)$, is the minimum degree required by a real polynomial to uniformly approximate f to error ε (see Definition 6.12). Recently, two of the authors [42] devised a way of lifting approximate degree bounds to approximate spectral norm bounds. We first show here that technique works a bit more generally, to yield the following: let $\mathsf{ADDR}_{m,t} : \{-1,1\}^m \to [t]$ be a (possibly partial) addressing function (see Definition 6.8). For any function $f : \{-1,1\}^n \to \{-1,1\}$, define the (partial) function $f^{\mathsf{ADDR}_{m,t}} : \{-1,1\}^{n\times t} \times \{-1,1\}^{n\times m} \to \{-1,1\}$ as follows (formally defined in Definition 6.10):

$$f^{\mathsf{ADDR}_{m,t}}(x,y) = f\left(x_{1,\mathsf{ADDR}_{m,t}(y_1)}, x_{2,\mathsf{ADDR}_{m,t}(y_2)}, \ldots, x_{n,\mathsf{ADDR}_{m,t}(y_n)}\right).$$

Our main result on lower bounding the spectral norm is stated below.

Lemma 6.26 (extending [42])**.** *Let $t > 1$ be any integer, $\mathsf{ADDR}_{m,t}$ be any (partial) addressing function and $f : \{-1,1\}^n \to \{-1,1\}$ be any function. Then,*

$$\log\left(\|\widehat{f^{\mathsf{ADDR}_{m,t}}}\|_{1,1/3}\right) = \Omega\left(\widetilde{\deg}(f)\log t\right).$$

The functions F constructed for the proof of Theorem 6.25 are completions of instances of $\mathsf{PARITY}^{\mathsf{ADDR}_{\ell,\ell}}$, and hence Lemma 6.26 yields lower bounds on the approximate spectral norm of F in terms of the approximate degree of PARITY (which is known to be maximal).

For the upper bound on $Q(F)$ we use two famous query algorithms - Grover's search [76] and the Bernstein-Vazirani algorithm [24]. The use of these algorithms for upper bounding $Q(F)$ is in the same taste as in the work of Ambainis and de Wolf [8] although their motivation was quite different than ours. Interestingly, Ambainis and de Wolf used their function to pin down the minimal *approximate degree* of a total Boolean function, all of whose input variables are influential.

6.3.1 Intuition behind the function construction

- From Theorem 6.1 it is known that for all Boolean functions f, $Q^{cc}(f \circ \mathsf{XOR}_2) \leq O(Q(f)\log n)$.

- In order to prove a matching lower bound, we construct a Boolean function F on n variables such that $\|\widehat{F}\|_{1,1/3} = 2^{\Omega(Q(F)\log n)}$ (Theorem 6.25). From Theorem 6.17, this shows that $Q^{cc}(F \circ \mathsf{XOR}_2) = \Omega(\log \|\widehat{F}\|_{1,1/3}) = \Omega(Q(F)\log n)$. We want to additionally ensure that $Q(F) = \Theta(n^\delta)$ for a given constant $0 < \delta < 1$. A formal definition of F is given in Figure 6.1, we attempt to provide an overview on how we arrived at this function below.

- Assume δ is a constant that is least $1/2$, else the argument follows along similar lines by ignoring suitably many input variables when defining the function. A natural first attempt is to try to construct a composed function of the form $F = f^{\mathsf{ADDR}}$, for some addressing function ADDR (see Definition 6.8) with $\Omega(n^{1-\delta})$ many target bits, for which $Q(f^{\mathsf{ADDR}}) = \Theta(\widetilde{\deg}(f))$. For the lower bound we use Lemma 6.26 to show that $\log \|\widehat{f^{\mathsf{ADDR}}}\|_{1,1/3} = \Omega(\widetilde{\deg}(f)\log(n^{1-\delta})) = \Omega(\widetilde{\deg}(f)\log n)$.

- Given the upper bound target, we are led to a natural choice of addressing function. Let $\mathsf{HADD}_{n^{1-\delta}}$ be the $(n^{1-\delta}, n^{1-\delta})$-addressing function defined as follows. Fix an arbitrary order on the $n^{1-\delta}$-bit Hadamard codewords (see Definition 6.7) , say $w_1, \ldots, w_{n^{1-\delta}}$. Define g to be the selector function of $\mathsf{HADD}_{n^{1-\delta}}$ such that $g(w_i) = i$ for all $i \in [n^{1-\delta}]$, and $g(x) = \star$ for $x \neq w_i$ for any $i \in [n^{1-\delta}]$.

- For any function f on $n^\delta/2$ bits, the *partial* function $f^{\mathsf{HADD}_{n^{1-\delta}}}$ on n inputs has quantum query complexity $O(Q(f) + n^\delta/2)$, as we sketch in the next step. We select f appropriately such that this is $\Theta(Q(f))$. Finally, we define the *total* function $F = f^{\mathsf{HADD}_{n^{1-\delta}}}$ to be the completion of $f^{\mathsf{HADD}_{n^{1-\delta}}}$ that evaluates to -1 on the non-promise inputs of $f^{\mathsf{HADD}_{n^{1-\delta}}}$.

- We choose the outer function to be $f = \mathsf{PARITY}_{n^\delta/2}$ to ensure $Q(F) = \Theta(n^\delta)$. To prove the upper bound on $Q(F)$, we crucially use the Bernstein-Vazirani and Grover's search algorithms.

  - Run $n^\delta/2$ instances of the Bernstein-Vazirani algorithm [24], one on each block. This algorithm guarantees that if the address variables were all Hadamard codewords, then we would receive the correct indices of the target variables with probability 1, and just $n^\delta/2$ queries.

  - In the next step, we run Grover's search [76, 28] on two $n/2$-bit strings to test whether the output of the first step was correct. If it was correct, we succeed with probability 1, and proceed to query the $n^\delta/2$ selected target variables and output the parity of them. If it was not correct, Grover's search catches a discrepancy

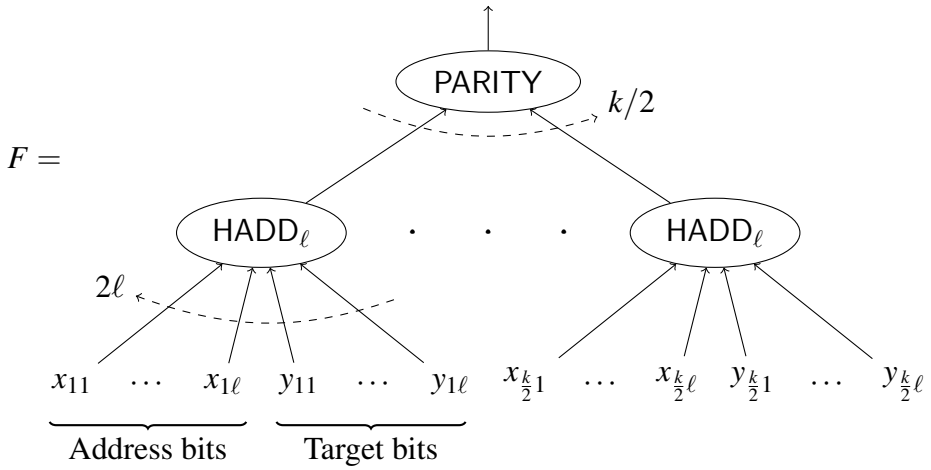


Fig. 6.1 $k = n^{\delta}, \ell = n^{1-\delta}$. If the address bits of an input to the r'th HADD_ℓ is the j'th Hadamard codeword, then y_{rj} is selected. If on an input, there exists at least one HADD_ℓ for which the address bits do not correspond to a Hadamard codeword, F outputs -1. Else it outputs the parity of the $k/2$ selected target bits.

with probability at least $2/3$ and we output -1, succeeding with probability at least $2/3$ in this case.

– The $n^{\delta}/2$ invocations of the Bernstein-Vazirani algorithm use a total of $n^{\delta}/2$ queries, Grover's search uses another $O(\sqrt{n})$ queries, and the final parity (if Grover's search outputs that the strings are equal) uses another $n^{\delta}/2$ queries, for a cumulative total of $O(n^{\delta} + \sqrt{n}) = O(n^{\delta})$ queries (recall that we assume $\delta \geq 1/2$).

6.4 Proof of Theorem 6.3

In this section, we prove Theorem 6.3. We first formally define the function we use.

6.4.1 Definition of the function

If $\delta < 1/2$, then ignore the last $n - 2n^{2\delta}$ bits of the input, and define the following function on the first $2n^{\delta}$ bits of the input. The same argument as in Sections 6.4.2 and 6.4.3 give the required bounds for Theorem 6.3 and Theorem 6.25. Hence, we may assume without loss of generality that $\delta \geq 1/2$.

Define the partial function $f : \{-1,1\}^n \to \{-1,1,\star\}$ by $f = \mathsf{PARITY}_{n^\delta/2}^{\mathsf{HADD}_{n^{1-\delta}}}$. Define F to be the completion of f that evaluates to -1 on the non-promise domain of f (see Figure 6.1).

6.4.2 Upper bound

In this section, we prove the following.

Claim 6.27. *For $F : \{-1,1\}^n \to \{-1,1\}$ defined as in Section 6.4.1, we have $Q(F) = \Theta(n^\delta)$.*

The upper bound follows along the lines of a proof in [8], and the lower bound just uses the fact that F is at least as hard as $\mathsf{PARITY}_{n^\delta/2}$.

Proof. Recall that an input to F is viewed as $(x_{11},\ldots,x_{1,\ell},y_{11},\ldots,y_{1,\ell},\ldots,x_{\frac{k}{2}1},\ldots,x_{\frac{k}{2}\ell},y_{\frac{k}{2}1},\ldots,y_{\frac{k}{2}\ell})$. The following is an $O(n^\delta)$-query quantum algorithm computing F. For convenience, set $\ell = n^{1-\delta}$ and $k = n^\delta$. Note that since $\delta \geq 1/2$, we have $k = \Omega(\ell)$.

1. Run $k/2$ instances of Bernstein-Vazirani algorithm on inputs $(x_{11},\ldots,x_{1\ell}),\ldots,(x_{\frac{k}{2}1},\ldots,x_{\frac{k}{2}\ell})$ to obtain $k/2$ strings $z_1,\ldots,z_{k/2}$.
2. Run Grover's search [76, 28] to check equality of the two strings: $h(z_1),\ldots,h(z_\ell)$ and $x_{11},\ldots,x_{1\ell},\ldots,x_{\frac{k}{2}1},\ldots,x_{\frac{k}{2}\ell}$, i.e. to check whether the addressing bits of the input are indeed all Hadamard codewords which are output by the first step.
3. If the step above outputs that the strings are equal, then query the $k/2$ selected variables and output their parity. Else, output -1.
    - If the input was indeed of the form as claimed in the first step, then Bernstein-Vazirani outputs the correct $z_1,\ldots,z_\ell$ with probability 1, and Grover's search verifies that the strings are equal with probability 1. Hence the algorithm is correct with probability 1 in this case.
    - If the input was not of the claimed form, then the two strings for which equality is to be checked in the second step are not equal. Grover's search catches a discrepancy with probability at least $2/3$. Hence, the algorithm is correct with probability at least $2/3$ in this case.

The correctness of the algorithm is argued above, and the cost is $k/2$ queries for the first step, $O(\sqrt{k\ell})$ queries for the second step, and at most $k/2$ for the third step. Thus, we have $Q(F) = O(k+\sqrt{k\ell}) = O(k)$, since $k = \Omega(\ell)$. The upper bound in the lemma follows.

For the lower bound, we argue that F is at least as hard as $\mathsf{PARITY}_{k/2}$. To see this formally, set all the address variables such that the selected target variables are the first target variable in each block. Under this restriction, F equals $\mathsf{PARITY}(y_{11},\dots,y_{\frac{k}{2}1})$. Thus any quantum query algorithm computing F must be able to compute $\mathsf{PARITY}_{k/2}$, and thus $Q(F)=\Omega(k)$. □

Remark 1. The same argument as above works when the function f is defined to be g^{HADD_ℓ} for any $g:\{-1,1\}^{n^\delta}\to\{-1,1\}$ satisfying $\widetilde{\deg}(g)=\Omega(n^\delta)$, and F is the completion of f that evaluates to -1 on all non-promise inputs. The same proof of Theorem 6.3 also goes through, but we fix $g=\mathsf{PARITY}_{n^\delta/2}$ for convenience.

6.4.3 Lower bound

In this section, we first prove Lemma 6.26. We require the following observation.

Observation 6.28. *For any $S\subseteq[n]$ and any $j\in S$, we have $\mathbb{E}_{x_j\sim\{-1,1\}}[\chi_S(x)]=0$, where x_j is distributed uniformly over $\{-1,1\}$.*

Proof of Lemma 6.26. Let $F=f^{\mathsf{ADDR}_{m,t}}$. Recall that our goal is to show that $\log\|\widehat{F}\|_{1,1/3}=\Omega(\widetilde{\deg}(f)\log t)$. We may assume $\widetilde{\deg}(f)\geq 1$, because the lemma is trivially true otherwise.

Towards a contradiction, suppose there exists a polynomial P of spectral norm strictly less than $(\frac{1}{10}\widetilde{\deg}_{0.99}(f)\log t)$ uniformly approximating F to error $1/3$ on the promise inputs (recall that from Lemma 6.14, we have $\widetilde{\deg}(f)=\Theta(\widetilde{\deg}_{0.99}(f))$).

Let ν be a distribution on the address bits of $\mathsf{ADDR}_{m,t}$ such that ν is supported only on assignments to the address variables that do not select $\star$, and is the uniform distribution over these assignments. Let $\mu=\nu^n$ be the product distribution over the address bits of the addressing functions in F.

- For any assignment z of the address variables from the support of μ, define a relevant (target) variable, with respect to z, to be one that is selected by z. Analogously, define a target variable to be irrelevant if it is not selected by z. Define a monomial to be relevant if it does not contain irrelevant variables, and irrelevant otherwise.
- Note that for any target variable, the probability with which it is selected is exactly $1/t$.

- Thus under any assignment z drawn from μ, for any monomial of the function P of degree $t \geq \widetilde{\deg}_{0.99}(f)$, the probability that it is relevant is at most $1/t^{\widetilde{\deg}_{0.99}(f)}$. Hence

$$\begin{aligned}
&\mathop{\mathbb{E}}_{z\sim\mu}[\ell_1\text{-norm of relevant monomials w.r.t. } z \text{ in } P \text{ of degree } \geq \widetilde{\deg}_{0.99}(f)] \\
&= \sum_{|S|\geq\widetilde{\deg}_{0.99}(f)} |\widehat{P}(S)| \Pr_{z\sim\mu}[\chi_S \text{ is relevant w.r.t. } z] \\
&\leq \max_{|S|\geq\widetilde{\deg}_{0.99}(f)} \{\Pr_{z\sim\mu}[\chi_S \text{ is relevant w.r.t. } z]\} \cdot \|\widehat{P}\|_1 \\
&< \frac{1}{t^{\widetilde{\deg}_{0.99}(f)}} \cdot 2^{\frac{1}{10}\widetilde{\deg}_{0.99}(f)\log t} = 2^{(-\frac{9}{10})\widetilde{\deg}_{0.99}(f)\log t} < \frac{3}{5},
\end{aligned}$$

  where the last inequality holds because $t \geq 2$ and $\widetilde{\deg}_{0.99}(f) \geq 1$.

- Fix an assignment to the address variables from the support of μ such that under this assignment, the ℓ_1-norm of relevant monomials in P of degree $\geq \widetilde{\deg}_{0.99}(f)$ is less than $3/5$.

- Note that under this assignment (in fact under any assignment in the support of μ), the restricted F is just the function f on the n variables selected by the addressing functions. Denote by P_1 the polynomial on the target variables obtained from P by fixing address variables as per this assignment.

- Drop the relevant monomials of degree $\geq \widetilde{\deg}_{0.99}(f)$ from P_1 to get a polynomial P_2, which uniformly approximates the restricted F (which is f on n variables) to error $1/3+3/5 < 0.99$.

- Take expectation over irrelevant variables (from the distribution where each irrelevant variable independently takes values uniformly from $\{-1,1\}$). Under this expectation, the value of F does not change (since irrelevant variables do not affect F's output by definition), and all irrelevant monomials of P_2 become 0 (using Observation 6.28 and linearity of expectation). Hence, under this expectation we have $\mathbb{E}[P_2] = P_3$, where P_3 is a polynomial of degree strictly less than $\widetilde{\deg}_{0.99}(f)$. Furthermore, P_3 uniformly approximates f to error less than 0.99 which is a contradiction. □

As a corollary of Lemma 6.26, we obtain a lower bound on the approximate spectral norm of F, where F is defined as in Section 6.4.1. This yields a proof of Theorem 6.25.

Proof of Theorem 6.25. Construct F as in Section 6.4.1. Claim 6.27 implies $Q(F) = \Theta(n^\delta)$.

Let $f = \mathsf{PARITY}_{n^\delta/2}^{\mathsf{HADD}_{n^{1-\delta}}}$. Lemma 6.26 implies that $\|\widehat{f}\|_{1,1/3} = \Omega(n^\delta \log n)$.

Since F is a completion of f, we have $\|\widehat{F}\|_{1,1/3} = \Omega(n^{\delta}\log n)$, which proves the lower bound in Theorem 6.25. The upper bound follows from Theorem 6.1. □

We are now ready to prove our main theorem.

Proof of Theorem 6.3. It immediately follows from Theorem 6.25 and Theorem 6.17. □

6.5 Summary

We conclude with the following points: first, we find our main result somewhat surprising that simulating a query algorithm by a communication protocol in the quantum context has a larger overhead than in the classical context. Second, it is remarkable that this relatively fine overhead of $\log n$ can be detected using analytic techniques that are an adaptation of the generalized discrepancy method. Third, the function that we used in this work is an XOR function. Study of this class of functions is proving to be very insightful. A recent example is the refutation of the log-approximate-rank conjecture [43] and even its quantum version [10, 141]. Our work further advocates the study of XOR functions.

Along with the fact that $\widetilde{\deg}(f) \leq 2Q(f)$ [19], Theorem 6.25 yields the following corollary.

Corollary 6.29. *For any constant $0 < \delta < 1$, there exists a total function $F : \{-1,1\}^n \to \{-1,1\}$ for which $\widetilde{\deg}(F) = O(n^{\delta})$ and*

$$\log \|\widehat{F}\|_{1,1/3} = \Omega(\widetilde{\deg}(F)\log n).$$

It is easy to verify that the constructions of F that yield Theorem 6.25 for any fixed constant $0 < \delta < 1$, also satisfy $\widetilde{\deg}(F) = \Theta(n^{\delta})$. Thus, Corollary 6.29 also gives a negative answer to Open Problem 2 in [13, Section 6], where it was asked if any degree-d approximating polynomial to a Boolean function of approximate degree d has spectral norm at most $2^{O(d)}$. Thus to prove min-entropy of the Fourier spectrum of a Boolean function is upper bounded by approximate degree, it cannot follow from their observation that min-entropy is upper bounded by the logarithm of the approximate spectral norm. The following remains an interesting and important open problem: (how) can one prove that the min-entropy of the Fourier spectrum of a Boolean function is upper bounded by a constant multiple of its approximate degree? Such an inequality is implied by the Fourier Entropy Influence (FEI) Conjecture.

Chapter 7

Overhead in Query-to-Communication Simulation for general functions

7.1 Introduction

This chapter we continues the study of outer functions for which the BCW-Simulation Theorem is tight. See the last chapter for the setup of query-to-communication and the BCW-Simulation Theorem. We saw in the last chapter the construction of an outer function such that when composed with XOR_2 as the inner function, the query-to-communication simulation required $\log n$ overhead. Some natural questions that follow are:

1. Are there outer functions for which $\log n$ overhead is required in quantum query-to-communication when the inner function is AND_2?
2. Can we come up with a general recipe of constructing such functions?
3. Is there a natural class of functions for which the $\log n$ overhead is not required?

In this chapter we answer the first two questions and we address the third question in the next chapter. In addition to constructing outer functions for which $\log n$ overhead is required in quantum query-to-communication when the inner function is AND_2, we show that the BCW-Simulation theorem is tight in the following strong sense.

Theorem 7.1. *There exists total function $f : \{-1,1\}^n \to \{-1,1\}$, such that*

$$\mathsf{UPP}^{cc}(f \circ G) = \Omega(\mathsf{Q}(f)\log n)$$

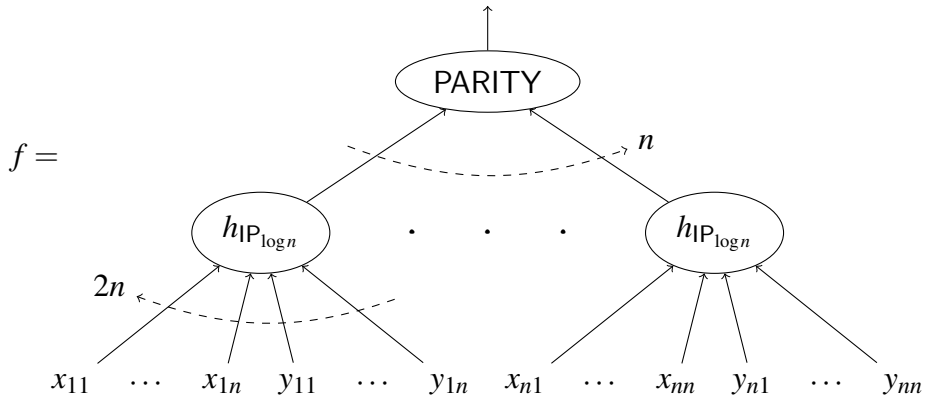


Fig. 7.1 If the inputs to the j-th $h_{\mathsf{IP}_{\log n}}$ are the Hadamard codewords $H(s_j)$ and $H(t_j)$ for all $j \in [n]$ and some $s_j, t_j \in \{-1,1\}^{\log n}$, then $f = \mathsf{PARITY}(\mathsf{IP}_{\log n}(s_1,t_1),\ldots,\mathsf{IP}_{\log n}(s_n,t_n))$. If there exists at least one $j \in [n]$ for which either $x_{j1},\ldots,x_{jn}$ or $y_{j1},\ldots,y_{jn}$ is not a Hadamard codeword, then f outputs -1.

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Here $\mathsf{UPP}^{cc}(f \circ G)$ denotes the unbounded-error quantum communication complexity of $f \circ G$ (adding "quantum" here only changes the communication complexity by a constant factor). The unbounded-error model of communication was introduced by Paturi and Simon [124] and is the strongest communication complexity model against which we know how to prove explicit lower bounds. This model is known to be strictly stronger than the bounded-error quantum model. For instance, the Set-Disjointness function on n inputs requires $\Omega(n)$ bits or $\Omega(\sqrt{n})$ qubits of communication in the bounded-error model, but only requires $O(\log n)$ bits of communication in the unbounded-error model. In fact, it follows from a recent result of Hatami, Hosseini and Lovett [83] that there exists a function $F : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$ with $\mathsf{Q}^{cc,*}(F) = \Omega(n)$ while $\mathsf{UPP}^{cc}(F) = O(1)$.

For proving Theorem 7.1, we exhibit a function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ whose bounded-error quantum query complexity is $O(n)$ and the unbounded-error communication complexity of $f \circ G$ is $\Omega(n \log n)$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Function construction: For the construction of f we first require the definition of Hadamard codewords. The Hadamard codeword of $s \in \{-1,1\}^{\log n}$, denoted by $H(s) \in \{-1,1\}^n$, is a list of all parities of s. See Figure 7.1 for a graphical visualization of f.

Query upper bound: The query upper bound of $O(n)$ follows along the lines of the last chapter, using the Bernstein-Vazirani algorithm to decode the Hadamard codewords, and Grover's algorithm to check that they actually are Hadamard codewords. This is similar to

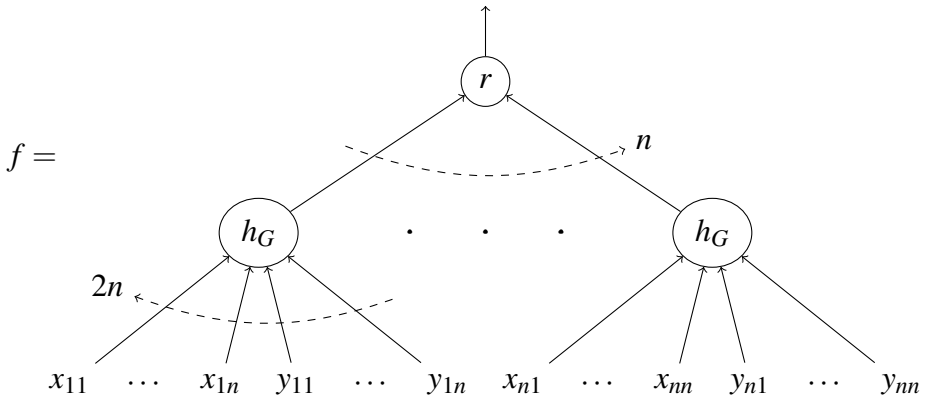


Fig. 7.2 In this figure, $G : \{-1,1\}^{\log} \times \{-1,1\}^{\log n} \to \{-1,1\}$. If the inputs to the j-th h_G are Hadamard codewords in $\pm H(s_j)$ and $\pm H(t_j)$ for all $j \in [n]$ and some $s_j, t_j \in \{-1,1\}^{\log n}$, then $f = r(G(s_1,t_1),\ldots,G(s_n,t_n))$. If there exists at least one $j \in [n]$ for which either $x_{j1},\ldots,x_{jn}$ or $y_{j1},\ldots,y_{jn}$ is not a Hadamard codeword, then f outputs -1.

the query upper bound from the last chapter. See the proof of Theorem 7.21 for the query algorithm and its analysis.

Communication lower bound: Towards the unbounded-error communication lower bound, we first recall that each input block of f equals $\mathsf{IP}_{\log n}$ if the inputs to each block are promised to be Hadamard codewords. Hence f equals $\mathsf{IP}_{n\log n}$ under this promise, since $\mathsf{PARITY}_n \circ \mathsf{IP}_{\log n} = \mathsf{IP}_{n\log n}$. Thus by setting certain inputs to Alice and Bob suitably, $f \circ G$ is at least as hard as $\mathsf{IP}_{n\log n}$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ (for a formal statement, see Lemma 7.23 with $r = \mathsf{PARITY}_n$ and $g = \mathsf{IP}_{\log n}$). It is known from a seminal result of Forster [61] that the unbounded-error communication complexity of $\mathsf{IP}_{n\log n}$ equals $\Omega(n\log n)$, completing the proof of the lower bound. This proof is more general than and arguably simpler than the proof of the lower bound for bounded-error quantum communication that we discussed in the last chapter.

We give a general recipe for constructing a class of functions that witness tightness of the BCW simulation where the inner gadget is either AND_2 or XOR_2. However, the communication lower bound we obtain here is in the bounded-error model in contrast to Theorem 7.1, where the communication lower bound is proven in the unbounded-error model.

The functions f constructed for this purpose are composed functions similar to the construction in Figure 7.1, except that we are able to use a more general class of functions in place of the outer PARITY function, and also a more general class of functions in place of the inner $\mathsf{IP}_{\log n}$ functions. See Figure 7.2 and its caption for an illustration and a more precise definition.

We require some additional constraints on the outer and inner functions. First, the approximate degree of r should be $\Omega(n)$. Second, the discrepancy of G should be small with respect to some "balanced" probability distribution (see Definition 7.15 and Definition 7.16 for formal definitions of these notions).

Theorem 7.2 (Formally stated in Theorem 7.4)**.** *Let $r : \{-1,1\}^n \to \{-1,1\}$ be such that $\widetilde{\deg}(r) = \Omega(n)$ and let $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a total function. Define $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ as in Figure 7.2. If there exists $\mu : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \mathbb{R}$ that is a balanced probability distribution with respect to G and $\mathrm{disc}_\mu(G) = n^{-\Omega(1)}$, then*

$$\mathsf{Q}(f) = O(n),$$
$$\mathsf{Q}^{cc,*}(f \circ G) = \Omega(n \log n),$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

The query upper bound follows along similar lines as that of Theorem 7.1. For the lower bound, we first show via a reduction that for f as described in Figure 7.2 and $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$, the communication problem $f \circ G$ is at least as hard as $r \circ G$ (see Lemma 7.23). This part of the lower bound proof is the same as in the proof of Theorem 7.1. For the hardness of $r \circ G$ (which in the case of Theorem 7.1 turned out to be $\mathsf{IP}_{n \log n}$, for which Forster's theorem yields an unbounded-error communication lower bound), we are able to use a theorem implicit in a work of Lee and Zhang [105]. This theorem gives a lower bound on the bounded-error communication complexity of $r \circ G$ in terms of the approximate degree of r and the discrepancy of G under a balanced distribution. For completeness, we provide an explicit proof in Section 7.6.

7.1.1 Results and Organization

As already discussed in the introduction, we prove the following two theorems in this chapter

Theorem 7.3 (Restatement of Theorem 7.1)**.** *There exists total function $f : \{-1,1\}^n \to \{-1,1\}$, such that*

$$\mathsf{UPP}^{cc}(f \circ G) = \Omega(\mathsf{Q}(f) \log n)$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Theorem 7.4 (Restatement of Theorem 7.2)**.** *Let $r : \{-1,1\}^n \to \{-1,1\}$ be a total function with $\widetilde{\deg}(r) = \Omega(n)$, $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a total function and $G \in$*

$\{\mathsf{AND}_2, \mathsf{XOR}_2\}$. *Define* $f = r \widetilde{\circ} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$. *If there exists* $\mu : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \mathbb{R}$ *that is a balanced probability distribution with respect to G and* $\mathrm{disc}_\mu(G) = n^{-\Omega(1)}$*, then*

$$\mathsf{Q}(f) = O(n),$$
$$\mathsf{Q}^{cc,*}(f \circ G) = \Omega(n \log n).$$

We also show that the function constructed in the last chapter admits (see Section 6.4, Definition 6.4.1 and Figure 6.1) $\log n$ overhead in query-to-communication simulation when composed with AND_2.

Organization. In Section 7.2 we give notations and preliminaries, recalling some facts that we have previously discussed for ease of reading. In Section 7.3 we prove Theorem 7.3 and Theorem 7.4. In Section 7.4 we show that $\log n$ overhead is required in BCW Simulation for the function from Definition 7.15 even when composed with AND_2. In Section 7.5 we give a separation between log-approximate-spectral norm and approximate degree for a transitive function. In Section 7.6 we present a lower bounds quantum communication complexity via the generalized discrepancy method. This is implicit in [105, Theorem 7] and we present it here for sake of completeness.

7.2 Notation and preliminaries

Without loss of generality, we assume n to be a power of 2 in this paper, unless explicitly stated otherwise. All logarithms in this paper are base 2. Let S_n denote the symmetric group over the set $[n] = \{1,\ldots,n\}$. For a string $x \in \{-1,1\}^n$ and $\sigma \in S_n$, let $\sigma(x)$ denote the string $x_{\sigma(1)},\ldots,x_{\sigma(n)} \in \{-1,1\}^n$. Consider an arbitrary but fixed bijection between subsets of $[\log n]$ and elements of $[n]$. For a string $s \in \{-1,1\}^{\log n}$, we abuse notation and also use s to denote the equivalent element of $[n]$. The view we take will be clear from context. For a string $x \in \{-1,1\}^n$ and set $S \subseteq [n]$, define the string $x_S \in \{-1,1\}^S$ to be the restriction of x to the coordinates in S. Let 1^n and $(-1)^n$ denote the n-bit string $(1,1,\ldots,1)$ and $(-1,-1,\ldots,-1)$, respectively.

7.2.1 Boolean functions

For strings $x, y \in \{-1,1\}^n$, let $\langle x, y \rangle$ denote the inner product (mod 2) of x and y. That is,

$$\langle x, y \rangle = \prod_{i=1}^{n} (x_i \mathsf{AND}_2 y_i).$$

For every positive integer n, let $\mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ be defined as:

$$\mathsf{PARITY}_n(x_1, \ldots, x_n) = \prod_{i \in [n]} x_i.$$

Definition 7.5 (Symmetric functions)**.** *A function $f : \{-1,1\}^n \to \{-1,1\}$ is symmetric if for all $\sigma \in S_n$ and for all $x \in \{-1,1\}^n$ we have $f(x) = f(\sigma(x))$.*

Definition 7.6 (Transitive functions)**.** *A function $f : \{-1,1\}^n \to \{-1,1\}$ is transitive if for all $i, j \in [n]$ there exists a permutation $\sigma \in S_n$ such that*

- *$\sigma(i) = j$, and*
- *$f(x) = f(\sigma(x))$ for all $x \in \{-1,1\}^n$.*

We next discuss function composition. For total functions f, g, let $f \circ g$ denote the standard composition of the functions f and g. We also require the following notion of composition of a total function f with a partial function g.

Definition 7.7 (Composition with partial functions)**.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a total function and let $g : \{-1,1\}^m \to \{-1,1,\star\}$ be a partial function. Let $f \mathbin{\widetilde{\circ}} g : \{-1,1\}^{nm} \to \{-1,1\}$ denote the total function that is defined as follows on input $(X_1, \ldots, X_n) \in \{-1,1\}^{nm}$, where $X_i \in \{-1,1\}^m$ for all $i \in [n]$.*

$$f \mathbin{\widetilde{\circ}} g(X_1, \ldots, X_n) = \begin{cases} f(g(X_1), \ldots, g(X_n)) & \text{if } g(X_i) \in \{-1,1\} \text{ for all } i \in [n], \\ -1 & \text{otherwise.} \end{cases}$$

That is, we use $f \mathbin{\widetilde{\circ}} g$ to denote the total function that equals $f \circ g$ on inputs when each copy of g outputs a value in $\{-1,1\}$, and equals -1 otherwise.

Definition 7.8 (Approximate degree)**.** *For every $\varepsilon \geq 0$, the ε-approximate degree of a function $f : \{-1,1\}^n \to \{-1,1\}$ is defined to be the minimum degree of a real polynomial*

$p : \{-1,1\}^n \to \mathbb{R}$ *that uniformly approximates* f *to error* ε*. That is,*

$$\widetilde{\deg}_\varepsilon(f) = \min\{\deg(p) : |p(x) - f(x)| \leq \varepsilon \text{ for all } x \in \{-1,1\}^n\}.$$

Unless specified otherwise, we drop ε *from the subscript and assume* $\varepsilon = 1/3$*.*

We assume familiarity with quantum computing [120], and use $\mathsf{Q}_\varepsilon(f)$ to denote the ε-error query complexity of f. Unless specified otherwise, we drop ε from the subscript and assume $\varepsilon = 1/3$.

Theorem 7.9 ([20])**.** *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be a function. Then* $\mathsf{Q}(f) \geq \widetilde{\deg}(f)/2$*.*

7.2.2 Communication complexity

The following terminology of two party functions will be helpful to distinguish between functions with domain $\{-1,1\}^n \times \{-1,1\}^n$ and communications problems with the same domain.

Definition 7.10 (Two-party function)**.** *We call a function* $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ *a* two-party function *to indicate that it corresponds to a communication problem in which Alice is given input* $x \in \{-1,1\}^j$*, Bob is given input* $y \in \{-1,1\}^k$*, and their task is to compute* $G(x,y)$*.*

Remark 2. Throughout this paper, we use uppercase letters to denote two-party functions, and lowercase letters to denote functions which are not two-party functions.

Definition 7.11 (Composition with two-party functions)**.** *Let* $f : \{-1,1\}^n \to \{-1,1\}$ *be a function and let* $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ *be a two-party function. Then* $F = f \circ G : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ *denotes the two-party function corresponding to the communication problem in which Alice is given input* $X = (X_1, \ldots, X_n) \in \{-1,1\}^{nj}$*, Bob is given* $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{nk}$*, and their task is compute* $F(X,Y) = f(G(X_1,Y_1), \ldots, G(X_n,Y_n))$*.*

Definition 7.12 (Inner Product function)**.** *For every positive integer n, define the function* $\mathsf{IP}_n : \{-1,1\}^n \times \{-1,1\}^n \to \{-1,1\}$ *by*

$$\mathsf{IP}_n(x,y) = \langle x, y \rangle.$$

In other words, $\mathsf{IP}_n = \mathsf{PARITY}_n \circ \mathsf{AND}_2$*.*

Observation 7.13. *For all positive integers* k,t*,* $\mathsf{PARITY}_k \circ \mathsf{IP}_t = \mathsf{IP}_{kt}$*.*

We also assume familiarity with quantum communication complexity [151]. We use $\mathrm{Q}^{cc}_{\varepsilon}(G)$ and $\mathrm{Q}^{cc,*}_{\varepsilon}(G)$ to represent the ε-error quantum communication complexity of a two-party function G in the models without and with unlimited shared entanglement, respectively. Unless specified otherwise, we drop ε from the subscript and assume $\varepsilon = 1/3$.

Definition 7.14 (Balanced probability distribution)**.** *We call a probability distribution $\mu : \{-1,1\}^n \to \mathbb{R}$ balanced with respect to a function $f : \{-1,1\}^n \to \{-1,1\}$ if $\sum_{x\in\{-1,1\}^n} f(x)\mu(x) = 0$.*

Definition 7.15 (Discrepancy)**.** *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a function and λ be a distribution on $\{-1,1\}^j \times \{-1,1\}^k$. For every $S \subseteq \{-1,1\}^j$ and $T \subseteq \{-1,1\}^k$, define*

$$\mathrm{disc}_{\lambda}(S\times T, G) = \Big|\sum_{x,y\in S\times T} G(x,y)\lambda(x,y)\Big|.$$

The discrepancy of G under the distribution λ is defined to be

$$\mathrm{disc}_{\lambda}(G) = \max_{S\subseteq\{-1,1\}^j, T\subseteq\{-1,1\}^k} \mathrm{disc}_{\lambda}(S\times T, G),$$

and the discrepancy of f is defined to be

$$\mathrm{disc}(G) = \min_{\lambda} \mathrm{disc}_{\lambda}(G).$$

Definition 7.16 (Balanced-discrepancy)**.** *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a function and Λ be the set of all balanced distributions on $\{-1,1\}^j \times \{-1,1\}^k$ with respect to G. The balanced-discrepancy of G is defined to be*

$$\mathrm{bdisc}(G) = \min_{\lambda\in\Lambda} \mathrm{disc}_{\lambda}(G).$$

The following theorem is implicit in [105, Theorem 7]. However, we prove it in Section 7.44 for completeness.

Theorem 7.17. *Let $r : \{-1,1\}^n \to \{-1,1\}$ and $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be functions. Let $\mu : \{-1,1\}^j \times \{-1,1\}^k \to \mathbb{R}$ be a balanced distribution with respect to G and $\mathrm{disc}_{\mu}(G) = o(1)$. If $\frac{8en}{\widetilde{\deg}(r)} \leq \left(\frac{1}{\mathrm{disc}_{\mu}(G)}\right)^{1-\beta}$ for some constant $\beta \in (0,1)$, then*

$$\mathrm{Q}^{cc,*}(r\circ G) = \Omega\left(\widetilde{\deg}(r)\log\left(\frac{1}{\mathrm{disc}_{\mu}(G)}\right)\right).$$

In particular,

$$\mathsf{Q}^{cc,*}(r \circ G) = \Omega\left(\widetilde{\deg}(r)\log\left(\frac{1}{\mathrm{bdisc}(G)}\right)\right).$$

7.2.3 Hadamard encoding

Recall that we index coordinates of n-bit strings by integers in $[n]$, and also interchangeably by strings in $\{-1,1\}^{\log n}$ via the natural correspondence. For $x \in \{-1,1\}^n$, let $-x \in \{-1,1\}^n$ be defined as $(-x)_i = -x_i$ for all $i \in [n]$. We use the notation $\pm x$ to denote the set $\{x,-x\}$.

Definition 7.18 (Hadamard Codewords)**.** *For every positive integer n and $s \in \{-1,1\}^{\log n}$, let $H(s) \in \{-1,1\}^n$ be defined as*

$$(H(s))_t = \prod_{i:s_i=-1} t_i \text{ for all } t \in \{-1,1\}^{\log n}.$$

If $x \in \{-1,1\}^n$ is such that $x = H(s)$ for some $s \in \{-1,1\}^{\log n}$, we say x is a Hadamard codeword corresponding to s.

That is, for every $s \in \{-1,1\}^{\log n}$, there is an n-bit Hadamard codeword corresponding to s. This represents the enumeration of all parities of s.

We now define how to encode a two-party total function G on $(\log j + \log k)$ input bits to a partial function h_G on $(j+k)$ input bits, using Hadamard encoding.

Definition 7.19 (Hadamardization of functions)**.** *Let $j,k \geq 1$ be powers of 2, and let $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$ be a function. Define a partial function $h_G : \{-1,1\}^{j+k} \to \{-1,1,\star\}$ by*

$$h_G(x,y) = \begin{cases} G(s,t) & \text{if } x \in \pm H(s), y \in \pm H(t) \text{ for some } s \in \{-1,1\}^{\log j}, t \in \{-1,1\}^{\log k} \\ \star & \text{otherwise.} \end{cases}$$

7.3 Necessity of the log-factor overhead in the BCW simulation

In this section we prove Theorem 7.3 and Theorem 7.4. For Theorem 7.3 we exhibit a function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ for which $\mathsf{Q}(f) = O(n)$ and $\mathsf{UPP}(f \circ G) = \Omega(n \log n)$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. In Theorem 7.4 we show a general recipe for constructing total

functions $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ such that $\mathsf{Q}(f) = O(n)$ and $\mathsf{Q}^{cc,*}(f \circ G) = \Omega(n \log n)$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. We first give a formal statement of Theorem 7.4.

Theorem 7.20 (Restatement of Theorem 7.4)**.** *Let $r : \{-1,1\}^n \to \{-1,1\}$ be a total function with $\widetilde{\deg}(r) = \Omega(n)$, $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a total function and $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. Define $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$. If there exists $\mu : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \mathbb{R}$ that is a balanced probability distribution with respect to G and $\mathrm{disc}_\mu(G) = n^{-\Omega(1)}$, then*

$$\mathsf{Q}(f) = O(n),$$
$$\mathsf{Q}^{cc,*}(f \circ G) = \Omega(n \log n).$$

The proofs of Theorem 7.3 and Theorem 7.4 each involve proving a query complexity upper bound and a communication complexity lower bound. The proofs of the query complexity upper bounds are along similar lines and follow from Theorem 7.21 and Corollary 7.22 (see Section 7.3.1). The proofs of the communication complexity lower bounds each involve a reduction from a problem whose communication complexity is easier to analyze (see Lemma 7.23 in Section 7.3.2). Finally, we complete the proofs of Theorem 7.3 and Theorem 7.4 in Section 7.3.3.

7.3.1 Quantum query complexity upper bound

We start by stating the main theorem in this section.

Theorem 7.21. *Let $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$ and $r : \{-1,1\}^n \to \{-1,1\}$. Then the quantum query complexity of the function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{n(j+k)} \to \{-1,1\}$ is given by*

$$\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n + \sqrt{n(j+k)}).$$

As a corollary we obtain the following on instantiating $j = k = n$ and r as a Boolean function with quantum query complexity $\Theta(n)$ in Theorem 7.21.

Corollary 7.22. *Let $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ be a non-constant function and let $r : \{-1,1\}^n \to \{-1,1\}$ be a total function with $\mathsf{Q}(r) = \Theta(n)$. Then the quantum query complexity of the total function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is*

$$\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = \Theta(n).$$

Proof. The upper bound $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n)$ follows by plugging in parameters in Theorem 7.21.

For the lower bound, we show that $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) \geq \mathsf{Q}(r)$. Since G is non-constant, there exist $x_1, y_1, x_2, y_2 \in \{-1,1\}^{\log n}$ such that $G(x_1,y_1) = -1$ and $G(x_2,y_2) = 1$. Let $X_1 = H(x_1), Y_1 = H(y_1), X_2 = H(x_2)$ and $X_2 = H(y_2)$. Consider $r \mathbin{\widetilde{\circ}} h_G$ only restricted to inputs where the inputs to each copy of h_G are either (X_1,Y_1) or (X_2,Y_2). Under this restriction, $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is the same as $r : \{-1,1\}^n \to \{-1,1\}$. Thus $\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) \geq \mathsf{Q}(r) = \Omega(n)$. □

We now prove Theorem 7.21.

Proof of Theorem 7.21. Recall from Definition 7.19 that the function $h_G : \{-1,1\}^{j+k} \to \{-1,1\}$ is defined as $h_G(x,y) = G(s,t)$ if $x \in \pm H(s)$ and $y \in \pm H(t)$ for some $s \in \{-1,1\}^{\log j}$ and $t \in \{-1,1\}^{\log k}$, and $h_G(x,y) = \star$ otherwise.

Also recall from Definition 7.7 that the function $r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{n(j+k)} \to \{-1,1\}$ is defined as $r \mathbin{\widetilde{\circ}} h_G((X_1,Y_1),\ldots,(X_n,Y_n)) = r \circ h_G((X_1,Y_1),\ldots,(X_n,Y_n))$ if $h_G((X_i,Y_i)) \in \{-1,1\}$ for all $i \in [n]$, and -1 otherwise.

Quantum query algorithm: View inputs to $r \mathbin{\widetilde{\circ}} h_G$ as $(X_1,Y_1,\ldots,X_n,Y_n)$, where $X_i \in \{-1,1\}^j$ for all $i \in [n]$ and $Y_i \in \{-1,1\}^k$ for all $i \in [n]$. We give a quantum algorithm and its analysis below.

1. Run $2n$ instances of the Bernstein-Vazirani algorithm: 1 instance on each X_i and 1 instance on each Y_i, to obtain $2n$ strings $x_1,\ldots,x_n,y_1,\ldots,y_n$, where each x_i is a $(\log j)$-bit string and each y_i is a $(\log k)$-bit string.
2. For each X_i and Y_i, query $(X_i)_{1^{\log j}}$ and $(Y_i)_{1^{\log k}}$ to obtain bits $b_i, c_i \in \{-1,1\}$ for all $i \in [n]$.
3. Run Grover's search [76, 28] to check equality of the following two $(nj+nk)$-bit strings: $(b_1 H(x_1),\ldots,b_n H(x_n), c_1 H(y_1),\ldots,c_n H(y_n))$ and $(X_1,\ldots,X_n,Y_1,\ldots,Y_n)$.
4. If the step above outputs that the strings are equal, then output $r(G(x_1,y_1),\ldots,G(x_n,y_n))$. Else, output -1.

Analysis of the algorithm:

- If the input is indeed of the form $(X_1,Y_1),\ldots,(X_n,Y_n)$ where each $X_i \in \pm H(x_i)$ and $Y_i \in \pm H(y_i)$ for some $x_i \in \{-1,1\}^{\log j}$ and $y_i \in \{-1,1\}^{\log k}$, then Step 1 outputs the

correct strings $x_1,\ldots,x_n,y_1,\ldots,y_n$ with probability 1 by the properties of the Bernstein-Vazirani algorithm. Step 2 then implies that $X_i = b_i H(x_i)$ and $Y_i = c_i H(y_i)$ for all $i \in [n]$. Next, Step 3 outputs that the strings are equal with probability 1 (since the strings whose equality are to be checked are equal). Hence the algorithm is correct with probability 1 in this case, since $(r \mathbin{\widetilde{\circ}} h_G)(X_1,Y_1,\ldots,X_n,Y_n) = r(G(x_1,y_1),\ldots,G(x_n,y_n))$.

- If the input is such that there exists an index $i \in [n]$ for which $X_i \notin \pm H(x_i)$ for every $x_i \in \{-1,1\}^{\log j}$ or $Y_i \notin \pm H(y_i)$ for every $y_i \in \{-1,1\}^{\log k}$, then the two strings for which equality is to be checked in the Step 3 are not equal. Grover's search catches a discrepancy with probability at least $2/3$. Hence, the algorithm outputs -1 (as does $r \mathbin{\widetilde{\circ}} h_G$), and is correct with probability at least $2/3$ in this case.

Cost of the algorithm: Step 1 accounts for $2n$ quantum queries. Step 2 accounts for $2n$ quantum queries. Step 3 accounts for $O(\sqrt{n(j+k)})$ quantum queries.

Thus,

$$\mathsf{Q}(r \mathbin{\widetilde{\circ}} h_G) = O(n + \sqrt{n(j+k)}).$$

□

7.3.2 Quantum communication complexity lower bound

In this section we first show a communication lower bound (under some model) on $(r \mathbin{\widetilde{\circ}} h_G) \circ G$ in terms of the communication complexity of $r \circ G$ (in the same model of communication) using a simple reduction. We state the lemma below (Lemma 7.23) for the case where the models under consideration are the bounded-error and unbounded-error quantum models, since these are the models of interest to us.

Lemma 7.23. *Let* $r : \{-1,1\}^n \to \{-1,1\}$, $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$, $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ *and* $CC \in \{\mathsf{Q}^{cc,*}, \mathsf{UPP}^{cc}\}$. *Then,*

$$CC((r \mathbin{\widetilde{\circ}} h_G) \circ G) \geq CC(r \circ G).$$

Proof. We first consider the case $G = \mathsf{AND}_2$. Consider a protocol Π of cost ℓ that solves $(r \mathbin{\widetilde{\circ}} h_G) \circ G$ in the CC-model. We exhibit below a protocol of cost ℓ that solves $r \circ G$ in the same model.

Suppose Alice is given input $x = (x_1,\ldots,x_n) \in \{-1,1\}^{n \log j}$ and Bob is given input $y = (y_1,\ldots,y_n) \in \{-1,1\}^{n \log k}$, where $x_i \in \{-1,1\}^{\log j}, y_i \in \{-1,1\}^{\log k}$ for each $i \in [n]$.

Preprocessing step: Alice constructs the $(n(j+k))$-bit string

$$X = ((H(x_1),(-1)^k),\ldots,(H(x_n),(-1)^k)) \in \{-1,1\}^{n(j+k)}, \tag{7.1}$$

and Bob constructs the $(n(j+k))$-bit string

$$Y = (((-1)^j,H(y_1)),\ldots,((-1)^j,H(y_n))) \in \{-1,1\}^{n(j+k)}. \tag{7.2}$$

Protocol: Alice and Bob run the protocol Π with input (X,Y) and output $\Pi(X,Y)$.

Cost: The preprocessing of the inputs to obtain X from x and Y from y takes no communication. Hence the total amount of communication is at most the cost of Π.

Correctness: For X and Y constructed in Equation (7.1) and Equation (7.2), respectively, we now argue that $(r \circ G)(x,y) = ((r \mathbin{\widetilde{\circ}} h_G) \circ \mathsf{AND}_2)(X,Y)$, which would conclude the proof for $G = \mathsf{AND}_2$.

$$\begin{aligned}((r \mathbin{\widetilde{\circ}} h_G) \circ \mathsf{AND}_2)(X,Y) &= (r \mathbin{\widetilde{\circ}} h_G)((H(x_1),H(y_1)),\ldots,(H(x_n),H(y_n))) \\ &= r(G(x_1,y_1),\ldots,G(x_n,y_n)) \\ &\qquad\qquad \text{by Definition 7.19 and Definition 7.7} \\ &= (r \circ G)(x,y).\end{aligned}$$

Thus,

$$CC((r \mathbin{\widetilde{\circ}} h_G) \circ \mathsf{AND}_2) \geq CC(r \circ G).$$

The argument for $G = \mathsf{XOR}_2$ follows along the same lines, with the strings $(-1)^j$ and $(-1)^k$ replaced by 1^j and 1^k, respectively, in the preprocessing step. □

7.3.3 On the tightness of the BCW simulation

We prove Theorem 7.4 in Section 7.3.3 and Theorem 8.2 in Section 7.3.3.

Proof of Theorem 7.4

Towards proving Theorem 7.4, we first observe how to obtain bounded-error quantum communication complexity lower bounds using Theorem 7.17 and Lemma 7.23.

Lemma 7.24. *Let* $r : \{-1,1\}^n \to \{-1,1\}$ *and* $G : \{-1,1\}^{\log j} \times \{-1,1\}^{\log k} \to \{-1,1\}$ *be functions such that* $\mathrm{bdisc}(G) = o(1)$ *and* $\frac{8en}{\widetilde{\deg}(r)} \leq \left(\frac{1}{\mathrm{bdisc}(G)}\right)^{1-\beta}$ *for some constant* $\beta \in (0,1)$. *Let* $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. *Then,*

$$\mathrm{Q}^{cc,*}((r \mathbin{\widetilde{\circ}} h_G) \circ G) = \Omega\left(\widetilde{\deg}(r) \log\left(\frac{1}{\mathrm{bdisc}(G)}\right)\right).$$

Proof. By Lemma 7.23 we have $\mathrm{Q}^{cc,*}((r \mathbin{\widetilde{\circ}} h_G) \circ G) \geq \mathrm{Q}^{cc,*}(r \circ G)$. By Theorem 7.17, $\mathrm{Q}^{cc,*}(r \circ G) = \Omega\left(\widetilde{\deg}(r) \log\left(\frac{1}{\mathrm{bdisc}(G)}\right)\right)$. □

We now prove Theorem 7.4.

Proof of Theorem 7.4. Let $r : \{-1,1\}^n \to \{-1,1\}$, $G : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$ and $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ be as in the statement of the theorem. We have $\mathrm{Q}(r) \geq \widetilde{\deg}(r)/2 = \Omega(n)$, where the first inequality follows by Theorem 7.9 and the second equality follows from the assumption that $\widetilde{\deg}(r) = \Omega(n)$. Moreover, $\mathrm{Q}(r) \leq n$ since r is a function on n input variables. Hence $\mathrm{Q}(r) = \Theta(n)$. Thus, Corollary 7.22 is applicable, and we have

$$\mathrm{Q}(f) = \Theta(n).$$

For the lower bound, $\widetilde{\deg}(r) = \Omega(n)$ by assumption. Thus

$$\frac{2en}{\widetilde{\deg}(r)} = O(1).$$

Also, since by assumption $\frac{1}{\mathrm{bdisc}(G)} = n^{\Omega(1)} = \omega(1)$, we have

$$\frac{2en}{\widetilde{\deg}(r)} \leq \left(\frac{1}{\mathrm{bdisc}(G)}\right)^{1-\beta}$$

for every constant $\beta \in (0,1)$. Lemma 7.24 implies

$$\mathrm{Q}^{cc,*}(f \circ G) = \Omega\left(\widetilde{\deg}(r) \log\left(\frac{1}{\mathrm{bdisc}(G)}\right)\right) = \Omega(n \log n).$$

□

Proof of Theorem 7.3

The total function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ that we use to prove Theorem 7.3 is $f = r \mathbin{\widetilde{\circ}} h_G$, where $r = \mathsf{PARITY}_n$ and $G = \mathsf{IP}_{\log n}$. Note that Theorem 7.4 implies $\mathsf{Q}(f) = O(n)$. For the quantum communication complexity lower bound in Theorem 7.3, we are able to show not only $\mathsf{Q}^{cc,*}(f \circ G) = \Omega(n \log n)$, but $\mathsf{UPP}^{cc}(f \circ G) = \Omega(n \log n)$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

We now prove Theorem 7.3.

Proof of Theorem 7.3. Let $n > 0$ be a power of 2. Let $r = \mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$. Let $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$. By Claim 8.13, f is transitive. By Corollary 7.22 we have

$$\mathsf{Q}(f) = \Theta(n).$$

For the communication lower bound we have

$$\begin{aligned}
\mathsf{UPP}^{cc}(f \circ G) &= \mathsf{UPP}^{cc}((r \mathbin{\widetilde{\circ}} h_G) \circ G) \\
&\geq \mathsf{UPP}^{cc}(\mathsf{PARITY}_n \circ \mathsf{IP}_{\log n}) && \text{by Lemma 7.23} \\
&= \mathsf{UPP}^{cc}(\mathsf{IP}_{n \log n}) && \text{Observation 7.13} \\
&= \Omega(n \log n). && \text{by Theorem 8.7}
\end{aligned}$$

□

7.4 Hardness of composing the function from Section 6.4 with AND_2

In Section 6.4, we exhibited a total function $f : \{-1,1\}^n \to \{-1,1\}$ for which $\mathsf{Q}^{cc,*}(f \circ \mathsf{XOR}_2) = \Omega(\mathsf{Q}(f) \log n)$, that is, the log-factor overhead in the BCW simulation is necessary for f when the inner function is XOR_2. In this section we show that the log-factor overhead is necessary for f even when the inner function is AND_2. We redefine that function here and recall some facts about that function for convenience. First, we view addressing function as a two-party function in the following definition.

Definition 7.25 (Addressing function)**.** *For an integer $n > 0$ that is a power of 2, let the Addressing function, denoted $\mathsf{ADDR}_n : \{-1,1\}^{\log n} \times \{-1,1\}^n \to \{-1,1\}$, be a two-party*

function defined as follows.

$$\mathsf{ADDR}_n(x_1,\ldots,x_{\log n},y_1,\ldots,y_n) = y_{\mathrm{bin}(i)},$$

where $\mathrm{bin}(i)$ *represents the integer in* $[n]$ *whose binary representation is i.*

In the corresponding communication problem, Alice holds the inputs $x_1,\ldots,x_{\log n}$*, and Bob holds the inputs* $y_1,\ldots,y_n$*.*

In the last chapter, we defined the following function $f:\{-1,1\}^{2n^2}\to\{-1,1\}$. Let $X_i,Y_i\in\{-1,1\}^n$ for all $i\in[n]$.

$$f(X_1,Y_1,\ldots,X_n,Y_n) = \begin{cases} \mathsf{PARITY}(\mathsf{ADDR}_n(x_1,Y_1),\ldots,\mathsf{ADDR}_n(x_n,Y_n)) \text{ if } \forall i\in[n], X_i\in\pm H(x_i) \\ -1 \text{ otherwise,} \end{cases} \tag{7.3}$$

That is, if there exist $x_i\in\{-1,1\}^{\log n}$ for all $i\in[n]$ such that $X_i\in\pm H(x_i)$, then $f(X_1,Y_1,\ldots,X_n,Y_n)$ equals the parity of $\mathsf{ADDR}_n(x_1,Y_1),\ldots,\mathsf{ADDR}_n(x_n,Y_n)$, and f equals -1 otherwise.

We proved the following quantum query upper bound on f in the last chapter.

Theorem 7.26. *For* $f:\{-1,1\}^{2n^2}\to\{-1,1\}$ *as defined in Equation* (7.3)*,* $\mathsf{Q}(f)=O(n)$.

Remark 3. The function considered in Section 6.4 is not exactly as in Equation (7.3) because the former one did not consider negations of Hadamard codewords as done in Equation (7.3). However, Theorem 7.26 can still be seen to hold by techniques identical to the last chapter.

In the last chapter, we showed the BCW simulation theorem is tight for f when the inner function is XOR_2, that is $\mathsf{Q}^{cc,*}(f\circ\mathsf{XOR}_2)=\Omega(n\log n)$. However, we left open the question whether the BCW simulation theorem is tight for f even when the inner function is AND_2. We now show that this is indeed the case. The quantum query upper bound holds by Theorem 7.26. For the communication complexity lower bound we use techniques similar to those in the proof of Lemma 7.23.

Lemma 7.27. *Let* $f:\{-1,1\}^{2n^2}\to\{-1,1\}$ *be as in Equation* (7.3)*, and* $CC\in\{\mathsf{Q}^{cc,*},\mathsf{UPP}^{cc}\}$*. Then,*

$$CC(r\circ\mathsf{AND}_2)\geq CC(\mathsf{PARITY}_n\circ\mathsf{ADDR}_n).$$

We require the following lemma.

Lemma 7.28. *Let $n > 0$ be a power of 2, and let U be the uniform distribution on $\{-1,1\}^{\log n} \times \{-1,1\}^n$. Then,*

$$\mathrm{disc}_U(\mathsf{ADDR}_n) \leq \frac{1}{\sqrt{n}}.$$

We provide a proof for completeness.

Proof. Consider the uniform distribution $U : \{-1,1\}^{\log n} \times \{-1,1\}^n \to \mathbb{R}$ defined as $U(x,y) = \frac{1}{n2^n}$ for all $x \in \{-1,1\}^{\log n}$ and $y \in \{-1,1\}^n$. Let $A_{n\times 2^n}$ be the communication matrix of $\mathsf{ADDR}_n : \{-1,1\}^{\log n} \times \{-1,1\}^n \to \{-1,1\}$. Observe that for every $i, j \in [n]$,

$$(AA^T)_{ij} = \sum_{x\in\{-1,1\}^n} x_i x_j = \begin{cases} 2^n \text{ if } i = j \\ 0 \text{ otherwise.} \end{cases}$$

Hence $AA^T = 2^n I_n$, where I_n denotes the $n \times n$ identity matrix. Thus, $\|A\| = \sqrt{2^n}$. For $S \subseteq [n]$ and $T \subseteq [2^n]$, let $1_S : [n] \to \{0,1\}$ and $1_T : [2^n] \to \{0,1\}$ be the indicator function of corresponding rows and columns of A, respectively. We now upper bound the discrepancy with respect to the uniform distribution:

$$\begin{aligned} \mathrm{disc}_U(\mathsf{ADDR}_n) &= \max_{S\subseteq[n],T\subseteq[2^n]} \left|(1_S)^T \cdot (U \circ A) \cdot 1_T\right| \\ &\qquad \text{where } (U \circ A) \text{ denotes the entry-wise product matrix of } U \text{ and } A \\ &\leq \sqrt{n} \cdot \|U \circ A\| \cdot \sqrt{2^n} \\ &= \frac{1}{\sqrt{n2^n}} \|A\| \\ &= \frac{1}{\sqrt{n}}. \end{aligned}$$

□

We now prove Lemma 7.27 using Lemma 7.28.

Proof of Lemma 7.27. Consider a protocol Π of cost ℓ that solves $f \circ \mathsf{AND}_2$ in the *CC*-model. We exhibit below a protocol of cost ℓ that solves $\mathsf{PARITY}_n \circ \mathsf{ADDR}_n$ in the same model. Recall from Definitions 7.11 and 7.25 that in the communication problem $\mathsf{PARITY}_n \circ \mathsf{ADDR}_n$, Alice is given the address bits and Bob is given the target bits for each copy of the inner gadget ADDR_n. Suppose Alice is given input $x = (x_1, \ldots, x_n) \in \{-1,1\}^{n\log n}$ and Bob is given input $y = (y_1, \ldots, y_n) \in \{-1,1\}^{n^2}$, where $x_i \in \{-1,1\}^{\log n}, y_i \in \{-1,1\}^n$ for each $i \in [n]$.

Preprocessing step: Alice constructs the following $(2n^2)$-bit string

$$X = (H(x_1),(-1)^n),\ldots,(H(x_n),(-1)^n)) \in \{-1,1\}^{2n^2}, \tag{7.4}$$

and Bob constructs the $(2n^2)$-bit string

$$Y = (((-1)^n,y_1),\ldots,((-1)^n,y_n)) \in \{-1,1\}^{2n^2}. \tag{7.5}$$

Protocol: Alice and Bob run the protocol Π with input (X,Y), and output $\Pi(X,Y)$.

Cost: The preprocessing of the inputs to obtain X from x and Y from y takes no communication. Hence the total amount of communication is at most the cost of Π.

Correctness: We now argue that $\mathsf{PARITY}_n \circ \mathsf{ADDR}_n(x,y) = (f \circ \mathsf{AND}_2)(X,Y)$, which would conclude the proof. We have from Equation (7.4) and Equation (7.5) that

$$\begin{aligned}(f \circ \mathsf{AND}_2)(X,Y) &= f((H(x_1),y_1),\ldots,(H(x_n),y_n)) \\ &= \mathsf{PARITY}_n(\mathsf{ADDR}_n(x_1,y_1),\ldots,\mathsf{ADDR}_n(x_n,y_n)) \quad \text{by Equation (7.3)} \\ &= \mathsf{PARITY}_n \circ \mathsf{ADDR}_n(x,y).\end{aligned}$$

Thus,

$$CC(f \circ \mathsf{AND}_2) \geq CC(\mathsf{PARITY}_n \circ \mathsf{ADDR}_n),$$

which proves the lemma. □

Theorem 7.29. *Let $n > 0$ be a sufficiently large power of 2. Let $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ be as in Equation* (7.3)*. Then,*

$$\begin{aligned}\mathsf{Q}(f) &= O(n), \\ \mathsf{Q}^{cc,*}(f \circ \mathsf{AND}_2) &= \Omega(n \log n).\end{aligned}$$

Proof. The upper bound follows from Theorem 7.26. For the lower bound, we first note that for sufficiently large n and every constant $\beta \in (0,1)$, we have

$$\frac{2en}{\widetilde{\deg}(\mathsf{PARITY}_n)} = 2e < (\sqrt{n})^{1-\beta} = \left(\frac{1}{\mathrm{disc}_U(\mathsf{ADDR}_n)}\right)^{1-\beta}, \tag{7.6}$$

where U denotes the uniform distribution over $\{-1,1\}^{\log n} \times \{-1,1\}^n$. The first equality above holds since $\widetilde{\deg}(\mathsf{PARITY}_n) = n$, and the last equality holds by Lemma 7.28. It is easy to verify that U is a balanced distribution w.r.t. ADDR_n. Hence,

$$\begin{aligned}
\mathsf{Q}^{cc,*}(f \circ \mathsf{AND}_2) &\geq \mathsf{Q}^{cc,*}(\mathsf{PARITY}_n \circ \mathsf{ADDR}_n) && \text{by Lemma 7.27} \\
&= \Omega\left(\widetilde{\deg}(\mathsf{PARITY}_n) \log\left(\frac{1}{\mathrm{disc}_U(\mathsf{ADDR}_n)}\right)\right) \\
&\text{by Theorem 7.17 (which is applicable by Equation (7.6) and since } U \text{ is balanced w.r.t. } \mathsf{ADDR}_n) \\
&= \Omega\left(n \log\left(\frac{1}{\mathrm{disc}_U(\mathsf{ADDR}_n)}\right)\right) && \text{since } \widetilde{\deg}(\mathsf{PARITY}_n) = n \\
&= \Omega(n \log n). && \text{by Lemma 7.28}
\end{aligned}$$

□

7.5 A separation between log-approximate-spectral norm and approximate degree for a transitive function

In this section, we exhibit a transitive function $f : \{-1,1\}^n \to \{-1,1\}$ for which the logarithm of its approximate spectral norm ($\log(\|\widehat{f}\|_{1,1/3})$) is at least $\Omega(\widetilde{\deg}(f) \log n)$.

Definition 7.30 (Spectral norm)**.** *Let $p : \{-1,1\}^n \to \mathbb{R}$ be a function, and let $p = \sum_{S \subseteq [n]} \widehat{p}(S)\chi_S$ denote its Fourier expansion. The spectral norm of p is define by*

$$\|\widehat{p}\|_1 := \sum_{S \subseteq [n]} \left|\widehat{p}(S)\right|.$$

Definition 7.31 (Approximate Spectral Norm)**.** *The approximate spectral norm of a function $f : \{-1,1\}^n \to \{-1,1\}$, denoted by $\|\widehat{f}\|_{1,\varepsilon}$ is defined to be the minimum spectral norm of a real polynomial $p : \{-1,1\}^n \to \mathbb{R}$ that satisfies $\left|p(x) - f(x)\right| \leq \varepsilon$ for all $x \in \{-1,1\}^n$. That is,*

$$\|\widehat{f}\|_{1,\varepsilon} := \min\left\{\|\widehat{p}\|_1 : \left|p(x) - f(x)\right| \leq \varepsilon \text{ for all } x \in \{-1,1\}^n\right\}.$$

As we discussed in the last chapter's summary, the following question raised in [13] was shown to have a negative answer, using the function f of Equation (7.3) to witness this.

Question 7.32 ([13, Section 4]). *Is it true that for all Boolean functions $f : \{-1,1\}^n \to \{-1,1\}$,*

$$\log(\|\widehat{f}\|_{1,\varepsilon}) = O(\widetilde{\deg}(f))?$$

In this section we show that Question 7.32 is false even for the special class of transitive Boolean functions. This is in stark contrast with the class of symmetric functions, for which Question 7.32 is true.

Claim 7.33. *There exists a transitive function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ such that* $\log\left(\|\widehat{f}\|_{1,1/3}\right) = \Omega(\widetilde{\deg}(f)\log n)$.

We first state some required preliminaries.

Definition 7.34 (Monomial projection). *We call a function $g : \{-1,1\}^m \to \{-1,1\}$ a monomial projection of a function $f : \{-1,1\}^n \to \{-1,1\}$ if g can be expressed as $g(x_1,\ldots,x_m) = f(M_1,\ldots,M_n)$, where each M_i is a monomial in the variables $x_1,\ldots,x_m$.*

It is known that the approximate spectral norm of a function can only decrease upon monomial projections (see, for example, [42, Observation 25]).

Observation 7.35 ([42, Observation 25]). *For $f : \{-1,1\}^n \to \{-1,1\}$ and $g : \{-1,1\}^m \to \{-1,1\}$ such that g is a monomial projection of f,*

$$\|\widehat{g}\|_{1,1/3} \leq \|\widehat{f}\|_{1,1/3}.$$

Fact 7.36 (Fourier coefficients of IP_n). *Let $\mathsf{IP}_n : \{-1,1\}^{2n} \to \{-1,1\}$ be as in Definition 7.12. Then for all $S \subseteq [2n]$ we have,*

$$\left|\widehat{\mathsf{IP}_n}(S)\right| = \frac{1}{2^n}.$$

Fact 7.37 (Plancherel's Theorem). *Let $f, g : \{-1,1\}^n \to \mathbb{R}$ be functions. Then,*

$$\sum_{x\in\{-1,1\}^n} f(x)g(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\widehat{g}(S).$$

Proof of Claim 7.33. Let $n > 0$ be a power of 2. Let $r = \mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$. From Claim 8.13, the function

$f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is transitive. By Corollary 7.22 we have

$$\mathsf{Q}(f) = \Theta(n). \tag{7.7}$$

By Theorem 7.9 we have $\mathsf{Q}(f) \geq \widetilde{\deg}(f)/2$ and together with Equation (7.7), this implies $\widetilde{\deg}(f) = O(n)$. Thus to complete the proof of the claim, it suffices to show $\log\left(\|\widehat{f}\|_{1,1/3}\right) = \Omega(n\log n)$.

We first note that $\mathsf{IP}_{n\log n}$ is a monomial projection of f. Consider the function f acting on the input variables $x^{(1)},\ldots,x^{(n)},y^{(1)},\ldots,y^{(n)}$, where $x^{(i)},y^{(i)} \in \{-1,1\}^n$ for all $i \in [n]$.

For $i \in [n]$, set $x^{(i)}_{1^{\log n}} = y^{(i)}_{1^{\log n}} = 1$. For $i \in [n]$ and string $s \in \{-1,1\}^{\log n} \setminus \left\{1^{\log n}\right\}$, set $x^{(i)}_s = \prod_{j:s_j=-1} x^{(i)}_j$ and $y^{(i)}_s = \prod_{j:s_j=-1} y^{(i)}_j$. That is, in each block of inputs $x^{(i)}$ and $y^{(i)}$, the coordinate corresponding to $1^{\log n}$ equals 1, the coordinates corresponding to $j \in \{-1,1\}^{\log n}$ with $|j| = 1$ are free variables, and all other variables are replaced by monomials in these variables. Under the above monomial projection there are $2n\log n$ free variables, namely $\left\{x^{(i)}_s, y^{(i)}_s : i \in [n], s \in \{-1,1\}^{\log n}, |s| = 1\right\}$. Also note that under this projection and every setting of the free variables, the blocks $x^{(i)}$ and $y^{(i)}$, for $i \in [n]$, are always Hadamard codewords. Let f' be the monomial projection (see Definition 7.34) of f under the projection defined in this paragraph. For the purpose of the next equality we abbreviate strings $s \in \{-1,1\}^{\log n}$ of Hamming weight 1 by the set $\{i\}$, where $i \in [\log n]$ is such that $s_i = -1$. On the free variables, the projected function f' equals

$$\mathsf{PARITY}_n\left(\langle(x^{(1)}_{\{1\}},\ldots,x^{(1)}_{\{\log n\}}),(y^{(1)}_{\{1\}},\ldots,y^{(1)}_{\{\log n\}})\rangle,\ldots,\langle(x^{(n)}_{\{1\}},\ldots,x^{(n)}_{\{\log n\}}),(y^{(n)}_{\{1\}},\ldots,y^{(n)}_{\{\log n\}})\rangle\right).$$

Thus,

$$f' = \mathsf{IP}_{n\log n}.$$

It follows from earlier works [31, 32] that every polynomial that approximates $\mathsf{IP}_{n\log n}$ to error $1/3$ must have spectral norm $2^{\Omega(n\log n)}$. We include a short proof below for completeness.

Let P be a polynomial that approximates $\mathsf{IP}_{n\log n}$ to error $1/3$. Since we have $P(x,y)\mathsf{IP}_{n\log n}(x,y) \geq 2/3$ for all $(x,y) \in \{-1,1\}^{2n\log n}$,

$$\begin{aligned}
2/3 &\leq \mathbb{E}_{x,y\in\{-1,1\}^{2n\log n}}[P(x,y)\mathsf{IP}_{n\log n}(x,y)] \\
&= \sum_{S\subseteq[2n\log n]} \widehat{P}(S)\widehat{\mathsf{IP}_{n\log n}}(S) && \text{by Plancherel's theorem} \\
&\leq \sum_{S\subseteq[2n\log n]} \frac{|\widehat{P}(S)|}{2^{n\log n}} && \text{by Fact 7.36} \\
&= \frac{\|\widehat{P}\|_1}{2^{n\log n}} \\
&\implies \log(\|\widehat{P}\|_1) = \Omega(n\log n) \\
&\implies \log(\|\widehat{\mathsf{IP}_{n\log n}}\|_{1,1/3}) = \Omega(n\log n).
\end{aligned}$$

This yields the desired contradiction by Observation 7.35.

□

7.6 Quantum communication lower bound via the generalized discrepancy method

In this section we prove Theorem 7.17, which gives a lower bound on the quantum communication complexity of a composed function in terms of the approximate degree of the outer function and the discrepancy of the inner function. This result is implicit in [105, Theorem 7]. Their result is stated in the more general setting of matrix norms, and the log factor we require on the right-hand side of Theorem 7.17 is not included in their statement. Theorem 7.17 follows implicitly from the proof of [105, Theorem 7] along with the fact that γ_2^*-norm characterizes discrepancy [108].

For completeness and clarity, we prove Theorem 7.17 below from first principles.

Definition 7.38. *For functions $f,g:\{-1,1\}^n \to \mathbb{R}$ and a probability distribution $\mu:\{-1,1\}^n \to \mathbb{R}$ the correlation between f and g with respect to μ is defined to be*

$$\operatorname{corr}_\mu(f,g) = \sum_{x\in\{-1,1\}^n} f(x)g(x)\mu(x).$$

For a Boolean function $f : \{-1,1\}^n \to \{-1,1\}$, its approximate degree can be captured by a certain linear program. Writing out the dual of this program and analyzing its optimum yields the following theorem (see, for example, [36, Theorem 1]).

Theorem 7.39 (Dual witness for ε-approximate degree)**.** *For every $\varepsilon \geq 0$ and $f : \{-1,1\}^n \to \{-1,1\}$, $\widetilde{\deg}_\varepsilon(f) \geq d$ if and only if there exists a polynomial $\psi : \{-1,1\}^n \to \mathbb{R}$ such that*

1. $\sum_{x\in\{-1,1\}^n} f(x)\psi(x) > \varepsilon$,
2. $\sum_{x\in\{-1,1\}^n} |\psi(x)| = 1$ *and*
3. $\widehat{\psi}(S) = 0$ *for all* $|S| < d$.

We require the following fact, which follows immediately from the fact that $\widehat{f}(S)$ is a uniform average of different signed values of $f(x)$.

Fact 7.40 (Folklore)**.** *For every function $f : \{-1,1\}^n \to \mathbb{R}$,*

$$2^n \max_{S\subseteq[n]} \left|\widehat{f}(S)\right| \leq \sum_{x\in\{-1,1\}^n} \left|f(x)\right|.$$

The following theorem shows that if a two-party function F correlates well with a two-party function G under some distribution λ, and the discrepancy of G under λ is small, then the bounded-error quantum communication complexity of F must be large. This is referred to as the generalized discrepancy method, and was first proposed by Klauck [99]. The following version can be found in [41, page 173], for example, stated as a lower bound on randomized communication complexity. However, the generalized discrepancy method is also known to give lower bounds on bounded-error quantum communication complexity, even in the model where the parties can share an arbitrary prior entangled state for free.

Theorem 7.41 (Generalized Discrepancy Bound)**.** *Consider functions $E, F : \{-1,1\}^m \times \{-1,1\}^n \to \{-1,1\}$. If there exists a distribution $\lambda : \{-1,1\}^m \times \{-1,1\}^n \to \mathbb{R}$ such that $\mathrm{corr}_\lambda(E,F) \geq \delta$, then*

$$\mathrm{Q}^{cc,*}_{\frac{1}{2}-\varepsilon}(E) = \Omega\left(\log\left(\frac{\delta + 2\varepsilon - 1}{\mathrm{disc}_\lambda(F)}\right)\right).$$

Definition 7.42. *For a distribution $\mu : \{-1,1\}^m \times \{-1,1\}^n \to \mathbb{R}$ and integer $k > 0$, define the distribution $\mu^{\otimes k} : \{-1,1\}^{mk} \times \{-1,1\}^{nk} \to \mathbb{R}$ by*

$$\mu^{\otimes k}((X_1,Y_1)\ldots,(X_k,Y_k)) = \prod_{i\in[k]} \mu(X_i,Y_i),$$

where $X_i \in \{-1,1\}^m$ and $Y_i \in \{-1,1\}^n$ for all $i \in [k]$.

We require the following XOR lemma for discrepancy due to Lee, Shraibman and Špalek [106].

Theorem 7.43 ([106, Theorem 19])**.** *Let $P : \{-1,1\}^m \times \{-1,1\}^n \to \{-1,1\}$ be a two-party function and $\mu : \{-1,1\}^m \times \{-1,1\}^n \to \mathbb{R}$ be a distribution. For every integer $k > 0$,*

$$\mathrm{disc}_{\mu^{\otimes k}}(\mathsf{PARITY}_k \circ P) \leq (8\mathrm{disc}_\mu(P))^k.$$

We recall Theorem 7.17 below.

Theorem 7.44 (Restatement of Theorem 7.17)**.** *Let $r : \{-1,1\}^n \to \{-1,1\}$ and $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be functions. Let $\mu : \{-1,1\}^j \times \{-1,1\}^k \to \mathbb{R}$ be a balanced distribution with respect to G and $\mathrm{disc}_\mu(G) = o(1)$. If $\frac{8en}{\widetilde{\deg}(r)} \leq \left(\frac{1}{\mathrm{disc}_\mu(G)}\right)^{1-\beta}$ for some constant $\beta \in (0,1)$, then*

$$\mathrm{Q}^{cc,*}(r \circ G) = \Omega\left(\widetilde{\deg}(r) \log\left(\frac{1}{\mathrm{disc}_\mu(G)}\right)\right).$$

In particular,

$$\mathrm{Q}^{cc,*}(r \circ G) = \Omega\left(\widetilde{\deg}(r) \log\left(\frac{1}{\mathrm{bdisc}(G)}\right)\right).$$

Proof. For simplicity we assume $j = k = m$. The general proof follows along similar lines. Let $\widetilde{\deg}(r) = d$, and let $\psi : \{-1,1\}^n \to \mathbb{R}$ be a dual witness for this as given by Theorem 7.39. Let $\nu : \{-1,1\}^n \to \mathbb{R}$ be defined as $\nu(x) = |\psi(x)|$ for all $x \in \{-1,1\}^n$ and also define $h : \{-1,1\}^n \to \{-1,1\}$ as $h(x) = \mathrm{sign}(\psi(x))$ for all $x \in \{-1,1\}^n$. First note that ν is a distribution since $\sum_{x\in\{-1,1\}^n} \nu(x) = \sum_{x\in\{-1,1\}^n} |\psi(x)| = 1$ by Theorem 7.39. From Theorem 7.39 we also have

$$\mathrm{corr}_\nu(r,h) > 1/3 \tag{7.8}$$

$$\widehat{h\nu}(S) = 0 \quad \text{for all } |S| < d, \tag{7.9}$$

where $h\nu(x) := h(x)\nu(x) = \psi(x)$ for all $x \in \{-1,1\}^n$. We will construct a probability distribution $\lambda : \{-1,1\}^{mn} \times \{-1,1\}^{mn} \to \mathbb{R}$ using $\nu : \{-1,1\}^n \to \mathbb{R}$ and $\mu : \{-1,1\}^m \times \{-1,1\}^m \to \mathbb{R}$ such that $r \circ G$ and $h \circ G$ have large correlation under λ. We will also show that $\mathrm{disc}_\lambda(h \circ G)$ is small. The proof of the theorem would then follow from Theorem 7.41.

Let $X,Y \in \{-1,1\}^{mn}$ be such that $X = (X_1,\ldots,X_n)$, $Y = (Y_1,\ldots,Y_n)$ and $X_i, Y_i \in \{-1,1\}^m$ for all $i \in [n]$. Also, let $G(X,Y) = (G(X_1,Y_1),\ldots,G(X_n,Y_n)) \in \{-1,1\}^n$.

Define $\lambda : \{-1,1\}^{mn} \times \{-1,1\}^{mn} \to \mathbb{R}$ as follows.

$$\lambda(X,Y) = 2^n \cdot \nu(G(X,Y)) \cdot \prod_{i\in[n]} \mu(X_i,Y_i). \tag{7.10}$$

Observe that for any $z \in \{-1,1\}^n$,

$$\begin{aligned}
\sum_{X,Y:G(X,Y)=z} \lambda(X,Y) &= \sum_{X,Y:G(X,Y)=z} 2^n \cdot \nu(G(X,Y)) \cdot \prod_{i\in[n]} \mu(X_i,Y_i) \qquad \text{by Equation (7.10)} \\
&= 2^n \nu(z) \prod_{i\in[n]} \left(\sum_{X_i,Y_i:G(X_i,Y_i)=z_i} \mu(X_i,Y_i) \right) \\
&= \nu(z), \qquad (7.11)
\end{aligned}$$

where the last equality follows since μ is balanced w.r.t. G by assumption. Thus

$$\sum_{X,Y} \lambda(X,Y) = \sum_{z\in\{-1,1\}^n} \nu(z) = 1.$$

We next observe that $\mathrm{corr}_\lambda(r\circ G, h\circ G)$ is large.

$$\begin{aligned}
\mathrm{corr}_\lambda(r\circ G, h\circ G) &= \sum_{X,Y} r\circ G(X,Y)\cdot h\circ G(X,Y)\cdot\lambda(X,Y) \\
&= \sum_{z\in\{-1,1\}^n} \sum_{X,Y:G(X,Y)=z} r(z)h(z)\cdot\lambda(X,Y) \\
&= \sum_{z\in\{-1,1\}^n} r(z)h(z)\nu(z) \qquad \text{by Equation (7.11)} \\
&= \mathrm{corr}_\nu(r,h) > 1/3. \qquad (7.12)
\end{aligned}$$

where the last equality follows from Equation (7.8).

We now upper bound the discrepancy of $h\circ G$ with respect to a rectangle R, under the distribution λ. Let $R \subseteq \{-1,1\}^{mn} \times \{-1,1\}^{mn}$ be any rectangle of the form $R(X,Y) =$

$A(X)B(Y)$ for $A : \{-1,1\}^{mn} \to \{0,1\}$ and $B : \{-1,1\}^{mn} \to \{0,1\}$.

$$\begin{aligned}
\mathrm{disc}_\lambda(h \circ G, R) &= \Big| \sum_{X,Y} h \circ G(X,Y) \cdot R(X,Y) \cdot \lambda(X,Y) \Big| \\
&= \Big| \sum_{z \in \{-1,1\}^n} \sum_{X,Y : G(X,Y)=z} h \circ G(X,Y) \cdot R(X,Y) \cdot \lambda(X,Y) \Big| \\
&= 2^n \Big| \sum_{z \in \{-1,1\}^n} h(z)\nu(z) \sum_{X,Y : G(X,Y)=z} R(X,Y) \prod_{i \in [n]} \mu(X_i,Y_i) \Big| \\
&\qquad \text{by Equation (7.10)} \\
&= 2^n \Big| \sum_{z \in \{-1,1\}^n} \left(\sum_{S \subseteq [n]} \widehat{h\nu}(S) \chi_S(z) \right) \sum_{X,Y : G(X,Y)=z} R(X,Y) \prod_{i \in [n]} \mu(X_i,Y_i) \Big| \\
&= 2^n \Big| \sum_{z \in \{-1,1\}^n} \sum_{S \subseteq [n]} \widehat{h\nu}(S) \sum_{X,Y : G(X,Y)=z} R(X,Y) \cdot \chi_S(z) \cdot \prod_{i \in [n]} \mu(X_i,Y_i) \Big| \\
&= 2^n \Big| \sum_{z \in \{-1,1\}^n} \sum_{S \subseteq [n]} \widehat{h\nu}(S) \sum_{X,Y : G(X,Y)=z} R(X,Y) \cdot \prod_{i \in S} G(X_i,Y_i) \cdot \prod_{i \in [n]} \mu(X_i,Y_i) \Big| \\
&\le 2^n \sum_{S \subseteq [n], |S| \ge d} \big| \widehat{h\nu}(S) \big| \cdot \Big| \sum_{X,Y} R(X,Y) \cdot \prod_{i \in S} G(X_i,Y_i)\mu(X_i,Y_i) \cdot \prod_{j \notin S} \mu(X_j,Y_j) \Big| \\
&\qquad \text{since } \widehat{h\nu}(S) = 0 \text{ for all } |S| < d \text{ by Equation (7.9)} \\
&\le 2^n \sum_{S \subseteq [n], |S| \ge d} \big| \widehat{h\nu}(S) \big| \cdot \Big| \sum_{X_j, Y_j : j \notin S} \prod_{j \notin S} \mu(X_j,Y_j) \left(\sum_{X_i, Y_i : i \in S} A(X)B(Y) \cdot \prod_{i \in S} G\mu(X_i,Y_i) \right) \Big|.
\end{aligned}$$

For any $X = (X_1, \ldots, X_n) \in \{-1,1\}^{mn}$ (respectively, $Y = (Y_1, \ldots, Y_n) \in \{-1,1\}^{mn}$) and set $S \subseteq [n]$ define the string $X_S \in \{-1,1\}^{m|S|}$ (respectively, $Y_S \in \{-1,1\}^{m|S|}$) by $X_S = (\ldots, X_i, \ldots)$ (respectively, $Y_S = (\ldots, Y_i, \ldots)$), where i ranges over all elements of S. For any S and fixed $\{X_j : j \notin S\}$ (respectively, $\{Y_j : j \notin S\}$) note that $A(X) = A'(X_S)$ (respectively, $B(X) = B'(X_S)$) for some $A' : \{-1,1\}^{m|S|} \to \{0,1\}$ (respectively, $B' : \{-1,1\}^{m|S|} \to \{0,1\}$) which is a function of the fixed values $\{X_j : j \notin S\}$ (respectively, $\{Y_j : j \notin S\}$). Let $R'(X_S, Y_S) = A'(X_S)B'(Y_S)$.

Continuing from the above we obtain,

$$\begin{aligned}
\mathrm{disc}_\lambda(h\circ G,R) &\le 2^n \sum_{S\subseteq[n],|S|\ge d} \left|\widehat{h\nu}(S)\right| \Big| \sum_{X_j,Y_j:j\notin S} \prod_{j\notin S} \mu(X_j,Y_j) \left(\sum_{X_i,Y_i:i\in S} R'(X_S,Y_S) \prod_{i\in S} G\mu(X_i,Y_i) \right) \Big| \\
&\le 2^n \sum_{S\subseteq[n],|S|\ge d} \left|\widehat{h\nu}(S)\right| \Big| \sum_{X_j,Y_j:j\notin S} \prod_{j\notin S} \mu_j(X_j,Y_j) \left(\mathrm{disc}_{\mu^{\otimes|S|}}(\mathsf{XOR}_{|S|}\circ G) \right) \Big| \\
&= 2^n \sum_{S\subseteq[n],|S|\ge d} \left|\widehat{h\nu}(S)\right| \cdot \mathrm{disc}_{\mu^{\otimes|S|}}(\mathsf{XOR}_{|S|}\circ G) \\
&\le \sum_{S\subseteq[n],|S|\ge d} \left(8\mathrm{disc}_\mu(G)\right)^{|S|} \qquad \text{from Fact 7.40 and Theorem 7.43} \\
&= \sum_{k=d}^{n} \binom{n}{k} \left(8\mathrm{disc}_\mu(G)\right)^k \le \sum_{k=d}^{n} \left(\frac{8en}{k}\mathrm{disc}_\mu(G)\right)^k \\
&\le \sum_{k=d}^{n} \left(\mathrm{disc}_\mu(G)^\beta\right)^k \qquad \text{since } \tfrac{8en}{d}\mathrm{disc}_\mu(G) \le \left(\mathrm{disc}_\mu(G)\right)^\beta \text{ by assumption} \\
&\le \frac{\mathrm{disc}_\mu(G)^{d\cdot\beta}}{1-\mathrm{disc}_\mu(G)^\beta}
\end{aligned}$$

Since $\mathrm{disc}_\mu(G) = o(1)$ by assumption ($\mathrm{disc}_\mu(G) = 1-\Omega(1)$ suffices, but we assume $\mathrm{disc}_\mu(G) = o(1)$ in the statement for readability) and $\beta \in (0,1)$ is a constant, we have $1-\mathrm{disc}_\mu(G)^\beta = \Omega(1)$, and hence

$$\mathrm{disc}_\lambda(h\circ G) = O(\mathrm{disc}_\mu(G)^{d\beta}). \tag{7.13}$$

Hence,

$$\begin{aligned}
\mathrm{Q}^{cc}_{1/2-2/5}(r\circ G) &\ge \log\left(\frac{1/3+4/5-1}{\mathrm{disc}_\lambda(h\circ G)}\right) && \text{by Theorem 7.41} \\
&= \Omega\left(\log\left(\frac{1}{\mathrm{disc}_\mu(G)^{d\beta}}\right)\right) && \text{by Equation (7.13)} \\
&= \Omega\left(\widetilde{\deg}(r)\log\left(\frac{1}{\mathrm{disc}_\mu(g)}\right)\right). \\
& && \text{since } \beta=\Omega(1) \text{ by assumption, and } d=\widetilde{\deg}(r)
\end{aligned}$$

The theorem follows since $\mathrm{Q}^{cc}(F) = \Theta(\mathrm{Q}^{cc}_\varepsilon(F))$ for all constants $\varepsilon \in (0,1/2)$. □

7.7 Summary

In this chapter we built upon the last chapter to construct functions for which logarithmic overhead is needed in query to communication simulation. We constructed functions for which for which BCW query-to-communication simulation is tight when composed with AND_2. Along with the result in the last chapter, this shows functions for which BCW query-to-communication simulation is tight for all non-constant two-bit inner functions. We then gave a general recipe to construct such functions which gives a wide range of functions for which an overhead is required in query-to-communication simulation.

Chapter 8

Symmetry and Quantum Query-to-Communication Simulation

8.1 Introduction

In the last two chapters we have constructed functions, that when composed with two-party functions AND_2 or XOR_2 witness a logarithmic overhead in query-to-communication simulation. It is natural to ask, like we did in the last chapter, whether there is an interesting class of functions for which there is no gap between query complexity and communication complexity when composed with AND_2 and XOR_2.

This gives rise to the following basic question: is there a natural class of functions for which the $\log n$ overhead in the BCW simulation is *not* required? Improving upon Høyer and de Wolf [87], Aaronson and Ambainis [1] showed that for the canonical problem of Set-Disjointness, the $\log n$ overhead in the BCW simulation can be avoided. Since the outer function NOR_n is symmetric (i.e., it only depends on the Hamming weight of its input, its number of -1s), a natural question is whether the $\log n$ overhead can be avoided whenever the outer function is symmetric. In this chapter we give a positive answer to this question.

Theorem 8.1. *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{0,1\}$, we have*

$$\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}_E^{cc}(G)).$$

Here $\mathsf{Q}_E^{cc}(G)$ denotes the *exact* quantum communication complexity of G, where the error probability is 0. In particular, if $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$ then $\mathsf{Q}_E^{cc}(G) = 1$ and hence $\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f))$.

Remark 4. If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$, then our protocol in the proof of Theorem 8.1 starts from a shared entangled state of $O(t \log n)$ EPR-pairs. Note that if $t \leq n\mathsf{Q}_E^{cc}(G)^2/(\log n)^2$ (this condition holds for instance if $\mathsf{Q}_E^{cc}(G) \geq \log n$) then this number of EPR-pairs is no more than the amount of communication and hence might as well be established in the first message, giving asymptotically the same upper bound $\mathsf{Q}^{cc}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}_E^{cc}(G))$ for the model without prior entanglement.

The next question one might ask is whether one can weaken the notion of symmetry required in Theorem 8.1. A natural generalization of the class of symmetric functions is the class of *transitive-symmetric* functions. A function $f : \{-1,1\}^n \to \{-1,1\}$ is said to be transitive-symmetric if for all $i, j \in [n]$, there exists $\sigma \in S_n$ such that $\sigma(i) = j$, and $f(x) = f(\sigma(x))$ for all $x \in \{-1,1\}^n$. Henceforth we refer to transitive-symmetric functions as simply transitive functions. Can the $\log n$ overhead in the BCW simulation be avoided whenever the outer function is transitive? We give a negative answer to this question in a strong sense: the $\log n$ overhead is still necessary even when we allow the quantum communication protocol an error probability that can be arbitrarily close to $1/2$. In fact, we have already constructed such a transitive function in the last chapter.

Theorem 8.2. *There exists a transitive and total function $f : \{-1,1\}^n \to \{-1,1\}$, such that*

$$\mathsf{UPP}^{cc}(f \circ G) = \Omega(\mathsf{Q}(f) \log n)$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Here $\mathsf{UPP}^{cc}(f \circ G)$ denotes the unbounded-error quantum communication complexity of $f \circ G$ (adding "quantum" here only changes the communication complexity by a constant factor). We note here that we have already seen the function of interest in the last theorem (see Figure 7.1 and Section 7.3.3). In this chapter we prve that this function is in fact transitive.

Theorem 8.1 and Theorem 8.2 clearly demonstrate the role of symmetry in determining the presence of the $\log n$ overhead in the BCW query-to-communication simulation: this overhead is absent for symmetric functions (Theorem 8.1), but present for a transitive function even when the model of communication under consideration is as strong as the unbounded-error model (Theorem 8.2).

Overview of our approach and techniques

In this section we discuss the ideas that go into the proofs of Theorem 8.1 and Theorem 8.2.

Communication complexity upper bound for symmetric functions

To prove Theorem 8.1 we use the well-known fact that every symmetric function f has an interval around Hamming weight $n/2$ where the function is constant; for NOR_n the length of this interval would be essentially n, while for PARITY_n it would be 1. To compute f, it suffices to either determine that the Hamming weight of the input lies in that interval (because the function value is the same throughout that interval) or to count the Hamming weight exactly.

For two-party functions of the form $f \circ G$, we want to do this type of counting on the n-bit string $z = (G(X_1,Y_1),\ldots,G(X_n,Y_n)) \in \{-1,1\}^n$. We show how this can be done with $O(\mathsf{Q}(f)\,\mathsf{Q}_E^{cc}(G))$ qubits of communication if we had a quantum protocol that can find -1s in the string z at a cost of $O(\sqrt{n}\,\mathsf{Q}_E^{cc}(G))$ qubits. Such a protocol was already given by Aaronson and Ambainis for the special case where $G = \mathsf{AND}_2$ for their optimal quantum protocol for Set-Disjointness, as a corollary of their quantum walk algorithm for search on a grid [1]. In this paper we give an alternative $O(\sqrt{n}\,\mathsf{Q}_E^{cc}(G))$-qubit protocol. This implies the result of Aaronson and Ambainis as a special case, but it is arguably simpler and may be of independent interest.

Our protocol can be viewed as an efficient distributed implementation of amplitude amplification with faulty components. In particular, we replace the usual reflection about the uniform superposition by an imperfect reflection about the n-dimensional maximally entangled state (= $\log n$ EPR-pairs if n is a power of 2). Such a reflection would require $O(\log n)$ qubits of communication to implement perfectly, but can be implemented with small error using only $O(1)$ qubits of communication, by invoking the efficient protocol of Aharonov et al. [4, Theorem 1] that tests whether a given bipartite state equals the n-dimensional maximally entangled state. Still, at the start of this protocol we need to assume (or establish by means of quantum communication) a shared state of $\log n$ EPR-pairs. If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$ then our protocol for $f \circ G$ will run the -1-finding protocol $O(t)$ times, which accounts for our assumption that we share $O(t \log n)$ EPR-pairs at the start of the protocol.

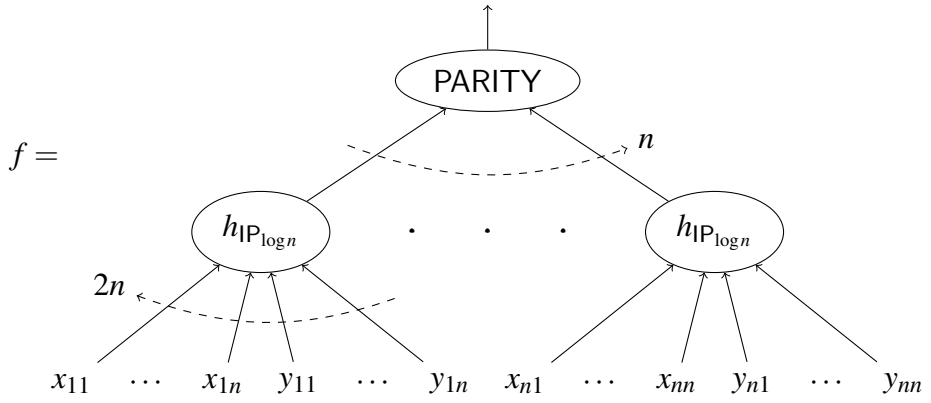


Fig. 8.1 If the inputs to the j-th $h_{\mathsf{IP}_{\log n}}$ are the Hadamard codewords $H(s_j)$ and $H(t_j)$ for all $j \in [n]$ and some $s_j, t_j \in \{-1,1\}^{\log n}$, then $f = \mathsf{PARITY}(\mathsf{IP}_{\log n}(s_1,t_1),\ldots,\mathsf{IP}_{\log n}(s_n,t_n))$. If there exists at least one $j \in [n]$ for which either $x_{j1},\ldots,x_{jn}$ or $y_{j1},\ldots,y_{jn}$ is not a Hadamard codeword, then f outputs -1.

Communication complexity lower bound for transitive functions

For proving Theorem 8.2, use the function $f : \{-1,1\}^{2n^2} \to \{-1,1\}$ that we constructed in Section 7.3. We reproduce the function in Figure 8.1 for convenience. The bounded-error quantum query complexity of this function is $O(n)$ and the unbounded-error communication complexity of $f \circ G$ is $\Omega(n \log n)$ for $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. We reproduce the figure of this function for ease of reading, see Figure 8.1. Using properties of IP and Hadamard codewords, and the symmetry of PARITY_n, we are able to show that f is transitive (see Claim 8.13).

8.1.1 Results and Organization

Section 8.2 introduces some notation and preliminaries. In Section 8.3 we construct our new one-sided error protocol for finding solutions in the string $z = (G(X_1,Y_1),\ldots,G(X_n,Y_n)) \in \{-1,1\}^n$, as a corollary of our distributed version of amplitude amplification. In Section 8.4 we prove Theorem 8.1, which shows that the $\log n$ overhead in the BCW simulation can be avoided when the outer function is symmetric.

Theorem 8.3 (Restatement of Theorem 8.1)**.** *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{0,1\}$, we have*

$$\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}^{cc}_E(G)).$$

The above theorem relies on the protocol from Section 8.3. In Section 8.5 we prove that $\log n$ overhead in required in query-to-communication simulation for a transitive function in a string sense.

Theorem 8.4 (Restatement of Theorem 8.2)**.** *There exists a transitive and total function* $f : \{-1,1\}^n \to \{-1,1\}$*, such that*

$$\mathsf{UPP}^{cc}(f \circ G) = \Omega(\mathsf{Q}(f)\log n)$$

for every $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$.

Recall that $\mathsf{UPP}^{cc}(f \circ G)$ denotes the unbounded-error quantum communication complexity of $f \circ G$. The fact the $\log n$ overhead ia required is already proven in Section 7.3, Theorem 7.2. In Section 8.5 we prove that the function from Theorem 7.2 (Figure 8.1) is transitive.

8.2 Preliminaries

In this section we recall some concepts from quantum computing. We also use several facts from the preliminaries of the previous chapter (see Section 7.2) but we do not state them again here.

The Bernstein-Vazirani algorithm [24] is a quantum query algorithm that takes an n-bit string as input and outputs a $(\log n)$-bit string. The algorithm has the following properties:

- the algorithm makes one quantum query to the input and
- if the input $x \in \{-1,1\}^n$ satisfies $x \in \pm H(s)$ for some $s \in \{-1,1\}^{\log n}$, then the algorithm returns s with probability 1.

Consider a symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$. Define the quantity

$$\Gamma(f) = \min\{|2k-n+1| : f(x) \neq f(y) \text{ if } |x| = k \text{ and } |y| = k+1\}$$

from [123]. One can think of $\Gamma(f)$ as essentially the length of the interval of Hamming weights around $n/2$ where f is constant (for example, for the majority and parity functions this would be 1, and for OR_n this would be $n-1$).

Theorem 8.5 ([20, Theorem 4.10])**.** *For every symmetric function* $f : \{-1,1\}^n \to \{-1,1\}$*, we have*

$$\mathsf{Q}(f) = \Theta(\sqrt{(n-\Gamma(f))n}).$$

The upper bound follows from a quantum algorithm that exactly counts the Hamming weight $|x|$ of the input if $|x| \leq t$ or $|x| \geq n-t$ for $t = \lceil (n-\Gamma(f))/2 \rceil$, and that otherwise learns $|x|$ is in the interval $[t+1, n-t-1]$ (which is an interval around $n/2$ where $f(x)$ is constant). By the definition of $\Gamma(f)$, this information about $|x|$ suffices to compute $f(x)$. In Section 8.4 we use this observation to give an efficient quantum communication protocol for a two-party function $f \circ G$.

We will need a unitary protocol that allows Alice and Bob to implement an approximate reflection about the n-dimensional maximally entangled state

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{i \in \{0,1\}^{\log n}} |i\rangle|i\rangle.$$

Ideally, such a reflection would map $|\psi\rangle$ to itself, and put a minus sign in front of all states orthogonal to $|\psi\rangle$. Doing this perfectly would requires $O(\log n)$ qubits of communication. Fortunately we can derive a cheaper protocol from a test that Aharonov et al. [4, Theorem 1] designed, which uses $O(\log(1/\varepsilon))$ qubits of communication and checks whether a given bipartite state equals $|\psi\rangle$, with one-sided error probability ε. By the usual trick of running this protocol, applying a Z-gate to the answer qubit, and then reversing the protocol, we can implement the desired reflection approximately. This requires possibly some auxiliary qubits on Alice and Bob's side which start in $|0\rangle$ and end in $|0\rangle$, except in a part of the final state that has norm at most ε. A bit more precisely:

Theorem 8.6. *Let $R_\psi = 2|\psi\rangle\langle\psi| - I$ be the reflection about the maximally entangled state shared between Alice and Bob. There exists a protocol that uses $O(\log(1/\varepsilon))$ qubits of communication and that implements a unitary R^ε_ψ such that $\| R^\varepsilon_\psi - R_\psi \| \leq \varepsilon$ and $R^\varepsilon_\psi|\psi\rangle = |\psi\rangle$.*

We use $\mathsf{UPP}^{cc}(F)$ to denote unbounded-error quantum communication complexity of two-party function F. It is folklore (see for example [91]) that the unbounded-error quantum communication complexity of F equals its classical counterpart up to a factor of at most 2 so it does not really matter much whether we use UPP^{cc} for classical unbounded-error communication complexity (as it is commonly used) or for quantum unbounded-error complexity. The unbounded-error model does not allow shared randomness or prior shared entanglement (which yields shared randomness by measuring) between Alice and Bob, since any two-party function F would have constant communication complexity in that setting. Crucially, for both the complexity of IP_n is linear in n:

Theorem 8.7 ([61])**.** *Let n be a positive integer. Then,*

$$\mathsf{UPP}^{cc}(\mathsf{IP}_n) = \Omega(n).$$

8.3 Noisy amplitude amplification and a new distributed-search protocol

In this section we present a version of quantum amplitude amplification that still works if the reflections involved are not perfectly implemented. In particular, the usual reflection about the uniform superposition will be replaced in the communication setting by an imperfect reflection about the n-dimensional maximally entangled state, based on the communication-efficient protocol of Aharonov et al. [4, Theorem 1] for testing whether Alice and Bob share that state. This allows us to avoid the $\log n$ factor that would be incurred if we instead used a BCW-style distributed implementation of standard amplitude amplification, with $O(\log n)$ qubits of communication to implement each query. Our main result in this section is the following general theorem, which allows us to search among a sequence of two-party instances $(X_1,Y_1),\ldots,(X_n,Y_n)$ for an index $i \in [n]$ where $G(X_i,Y_i) = -1$, for any two-party function G.

Theorem 8.8. *Let $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$ be a two-party function, $X = (X_1,\ldots,X_n) \in \{-1,1\}^{nj}$ and $Y = (Y_1,\ldots,Y_n) \in \{-1,1\}^{nk}$. Define $z = (G(X_1,Y_1),\ldots,G(X_n,Y_n)) \in \{-1,1\}^n$. Assume Alice and Bob start with $\lceil \log n \rceil$ shared EPR-pairs.*

- *There exists a quantum protocol using $O(\sqrt{n}\,\mathrm{Q}_E^{cc}(G))$ qubits of communication that finds (with success probability ≥ 0.99) an $i \in [n]$ such that $z_i = -1$ if such an i exists, and says "no" with probability 1 if no such i exists.*
- *If the number of -1s in z is within a factor of 2 from a known integer t, then the communication can be reduced to $O(\sqrt{n/t}\,\mathrm{Q}_E^{cc}(G))$ qubits.*

Remark 5. The $\log n$ shared EPR-pairs that we assume Alice and Bob share at the start could also be established by means of $\log n$ qubits of communication at the start of the protocol. For the result in the first bullet, this additional communication does not change the asymptotic bound. For the result of the second bullet, if $t \leq n\mathrm{Q}_E^{cc}(G)^2/(\log n)^2$, then this additional communication does not change the asymptotic bound either. However, if $t = \omega(n/(\log n)^2)$ and $\mathrm{Q}_E^{cc}(G) = O(1)$ then the quantum communication $O(\sqrt{n/t}\,\mathrm{Q}_E^{cc}(G))$ is $o(\log n)$ and establishing the $\log n$ EPR-pairs by means of a first message makes a difference.

As a corollary, we obtain a new $O(\sqrt{n})$-qubit protocol for the distributed search problem composed with $G = \mathsf{AND}_2$ (whose decision version is the Set-Disjointness problem).

8.3.1 Amplitude amplification with perfect reflections

We first describe basic amplitude amplification in a slightly unusual recursive manner, similar to [88]. We are dealing with a search problem where some set $\mathscr{G}$ of basis states are deemed "good" and the other basis states are deemed "bad." Let $P_{\mathscr{G}} = \sum_{g\in\mathscr{G}} |g\rangle\langle g|$ be the projector onto the span of the good basis states, and $O_{\mathscr{G}} = I - 2P_{\mathscr{G}}$ be the reflection that puts a '$-$' in front of the good basis states: $O_{\mathscr{G}}|g\rangle = -|g\rangle$ for all basis states $g \in \mathscr{G}$, and $O_{\mathscr{G}}|b\rangle = |b\rangle$ for all basis states $b \notin \mathscr{G}$.

Suppose we have an initial state $|\psi\rangle$ which is a superposition of a good state and a bad state:

$$|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle,$$

where $|G\rangle = P_{\mathscr{G}}|\psi\rangle / \| P_{\mathscr{G}}|\psi\rangle \|$ and $|B\rangle = (I - P_{\mathscr{G}})|\psi\rangle / \| (I - P_{\mathscr{G}})|\psi\rangle \|$. For example in Grover's algorithm, with a search space of size n containing t solutions, the initial state $|\psi\rangle$ would be the uniform superposition, and its overlap (inner product) with the good subspace spanned by the t "good" (sometimes called "marked") basis states would be $\sin(\theta) = \sqrt{t/n}$.

We'd like to increase the weight of the good state, i.e., move the angle θ closer to $\pi/2$. Let R_ψ denote the reflection about the state $|\psi\rangle$, i.e., $R_\psi|\psi\rangle = |\psi\rangle$ and $R_\psi|\phi\rangle = -|\phi\rangle$ for every $|\phi\rangle$ that is orthogonal to $|\psi\rangle$. Then the algorithm $A_1 = R_\psi \cdot O_{\mathscr{G}}$ is the product of two reflections, which (in the 2-dimensional space spanned by $|G\rangle$ and $|B\rangle$) corresponds to a rotation by an angle 2θ, thus increasing our angle from θ to 3θ. This is the basic amplitude amplification step. It maps

$$|\psi\rangle \mapsto A_1|\psi\rangle = \sin(3\theta)|G\rangle + \cos(3\theta)|B\rangle.$$

We can now repeat this step recursively, defining

$$A_2 = A_1 R_\psi A_1^* \cdot O_{\mathscr{G}} \cdot A_1.$$

Note that $A_1 R_\psi A_1^*$ is a reflection about the state $A_1|\psi\rangle$. Thus A_2 triples the angle between $A_1|\psi\rangle$ and $|B\rangle$, mapping

$$|\psi\rangle \mapsto A_2|\psi\rangle = \sin(9\theta)|G\rangle + \cos(9\theta)|B\rangle.$$

Continuing recursively in this fashion, define the algorithm

$$A_{j+1} = A_j R_\psi A_j^* \cdot O_{\mathscr{G}} \cdot A_j. \tag{8.1}$$

The last algorithm A_k will map

$$|\psi\rangle \mapsto A_k|\psi\rangle = \sin(3^k\theta)|G\rangle + \cos(3^k\theta)|B\rangle.$$

Hence after k recursive amplitude amplification steps, we have angle $3^k\theta$. Since we want to end up with angle $\approx \pi/2$, if we know θ then we can choose

$$k = \lfloor \log_3(\pi/(2\theta)) \rfloor. \tag{8.2}$$

This gives us an angle $3^k\theta \in (\pi/6, \pi/2]$, so the final state $A_k|\psi\rangle$ has overlap $\sin(\theta_k) > 1/2$ with the good state $|G\rangle$.

Let C_k denote the "cost" (in whatever measure, for example query complexity, or communication complexity, or circuit size) of algorithm A_k. Looking at its recursive definition (Equation (8.1)), C_k is 3 times C_{k-1}, plus the cost of R_ψ plus the cost of $O_{\mathscr{G}}$. If we just count applications of $O_{\mathscr{G}}$ ("queries"), considering R_ψ to be free, then $C_{k+1} = 3C_k + 1$. This recursion has the closed form $C_k = \sum_{i=0}^{k-1} 3^i < 3^k$. With the above choice of k we get $C_k = O(1/\theta)$. In the case of Grover's algorithm, where $\theta = \arcsin(\sqrt{t/n}) \approx \sqrt{t/n}$, the cost is $C_k = O(\sqrt{n/t})$.

8.3.2 Amplitude amplification with imperfect reflections

Now we consider the situation where we do not implement the reflections R_ψ perfectly, but instead implement another unitary R_ψ^ε at operator-norm distance $\| R_\psi^\varepsilon - R_\psi \| \leq \varepsilon$ from R_ψ, with the additional property that $R_\psi^\varepsilon|\psi\rangle = |\psi\rangle$ (this one-sided error property will be important for the proof). We can control this error ε, but smaller ε will typically correspond to higher cost of R_ψ^ε. The reflection $O_{\mathscr{G}}$ will still be implemented perfectly below.

We again start with the initial state

$$|\psi\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle.$$

For errors $\varepsilon_1, \ldots, \varepsilon_k$ that we will specify later, recursively define the following algorithms.

$$A_1 = R_\psi^{\varepsilon_1} \cdot O_{\mathscr{G}} \text{ and } A_{j+1} = A_j R_\psi^{\varepsilon_{j+1}} A_j^* \cdot O_{\mathscr{G}} \cdot A_j.$$

These algorithms will map the initial state as follows:

$$|\psi\rangle \mapsto |\psi_j\rangle = A_j|\psi\rangle = \sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle + |E_j\rangle, \tag{8.3}$$

where $|E_j\rangle$ is some unnormalized error state defined by the above equation; its norm η_j quantifies the extent to which we deviate from perfect amplitude amplification. Our goal here is to upper bound this η_j. In order to see how η_j can grow, let us see how $A_j R_\psi^{\varepsilon_{j+1}} A_j^* \cdot O_{\mathscr{G}}$ acts on $\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle$ (we'll take into account the effects of the error term $|E_j\rangle$ later). If $R_\psi^{\varepsilon_{j+1}}$ were equal to R_ψ, then we would have one perfect round of amplitude amplification and obtain $\sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle$; but since $R_\psi^{\varepsilon_{j+1}}$ is only ε_{j+1}-close to R_ψ, additional errors can appear. First we apply $O_{\mathscr{G}}$, which negates $|G\rangle$ and hence changes the state to

$$-\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle = |\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle.$$

Second we apply $V = A_j R_\psi^{\varepsilon_{j+1}} A_j^*$. Let $V' = A_j R_\psi A_j^*$, and note that $V|\psi_j\rangle = V'|\psi_j\rangle = |\psi_j\rangle$ and $\| V' - V \| = \| R_\psi - R_\psi^{\varepsilon_{j+1}} \| \leq \varepsilon_{j+1}$. The new state is

$$\begin{aligned} V(|\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle) &= V'(|\psi_j\rangle - |E_j\rangle - 2\sin(3^j\theta)|G\rangle) + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle) \\ &= V'(-\sin(3^j\theta)|G\rangle + \cos(3^j\theta)|B\rangle) + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle) \\ &= \sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle). \end{aligned}$$

Putting back also the earlier error term $|E_j\rangle$ from Equation (8.3) (to which the unitary $VO_{\mathscr{G}}$ is applied as well), it follows that the new error state is

$$|E_{j+1}\rangle = |\psi_{j+1}\rangle - (\sin(3^{j+1}\theta)|G\rangle + \cos(3^{j+1}\theta)|B\rangle) = VO_{\mathscr{G}}|E_j\rangle + (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle).$$

Its norm is

$$\begin{aligned} \eta_{j+1} &\leq \| VO_{\mathscr{G}}|E_j\rangle \| + \| (V'-V)(|E_j\rangle + 2\sin(3^j\theta)|G\rangle) \| \\ &\leq \eta_j + \varepsilon_{j+1}(\eta_j + 2\sin(3^j\theta)) = (1+\varepsilon_{j+1})\eta_j + 2\varepsilon_{j+1}\sin(3^j\theta). \end{aligned}$$

Since $\eta_0 = 0$, we can "unfold" the above recursive upper bound to the following, which is easy to verify by induction on k:

$$\eta_k \leq \sum_{j=1}^{k} \prod_{\ell=j+1}^{k} (1+\varepsilon_\ell) 2\varepsilon_j \sin(3^{j-1}\theta).$$

For each $1 \leq j \leq k$, choose

$$\varepsilon_j = \frac{1}{100 \cdot 4^j}. \tag{8.4}$$

Note that $\sigma = \sum_{j=1}^{k} \varepsilon_k \leq 1/300$. With this choice of ε_j's, and the inequalities $1+x \leq e^x$, $e^\sigma \leq 1.5$ and $\sin(x) \leq x$ for $x \leq \pi/2$ (which is the case here), we can upper bound the norm of the error term $|E_k\rangle$ after k iterations (see Equation (8.3)) as

$$\eta_k \leq \sum_{j=1}^{k} e^\sigma 2\varepsilon_j 3^{j-1}\theta \leq \frac{3\theta}{400} \sum_{j=1}^{k} (3/4)^{j-1} \leq \frac{3\theta}{100}. \tag{8.5}$$

Accordingly, up to very small error we have done perfect amplitude amplification.

8.3.3 Distributed amplitude amplification with imperfect reflection

We will now instantiate the above scheme to the case of *distributed* search, where our measure of cost is communication, that is, the number of qubits sent between Alice and Bob. Specifically, consider the *intersection problem* where Alice and Bob have inputs $x \in \{-1,1\}^n$ and $y \in \{-1,1\}^n$, respectively. Assume for simplicity that n is a power of 2, so $\log n$ is an integer. Alice and Bob want to find an $i \in \{0,\ldots,n-1\} = \{0,1\}^{\log n}$ such that $x_i = y_i = -1$, if such an i exists.

The basis states in this distributed problem are $|i\rangle|j\rangle$, and we define the set of "good" basis states as

$$\mathcal{G} = \{|i\rangle|j\rangle \mid x_i = y_j = -1\},$$

even though we are only looking for i, j where $i = j$ (it's easier to implement $O_{\mathcal{G}}$ with this more liberal definition of $\mathcal{G}$). Our protocol will start with the maximally entangled initial state $|\psi\rangle$ in n dimensions, which corresponds to $\log n$ EPR-pairs:

$$|\psi\rangle = \frac{1}{\sqrt{n}} \sum_{i \in \{0,1\}^{\log n}} |i\rangle|i\rangle = \sin(\theta)|G\rangle + \cos(\theta)|B\rangle,$$

where we assume there are t i's where $x_i = y_i = -1$, i.e., t solutions to the intersection problem, so

$$\theta = \arcsin(\sqrt{t/n}). \tag{8.6}$$

and

$$|G\rangle = \frac{1}{\sqrt{t}} \sum_{(i,i) \in \mathcal{G}} |i\rangle|i\rangle.$$

It costs $\lceil \log n \rceil$ qubits of communication between Alice and Bob to establish this initial shared state, or it costs nothing if we assume pre-shared entanglement. Our goal is to end up with a state that has large inner product with $|G\rangle$.

In order to be able to use amplitude amplification, we would like to be able to reflect about the above state $|\psi\rangle$. However, in general this perfect reflection R_ψ costs a lot of communication: Alice would send her $\log n$ qubits to Bob, who would unitarily put a -1 in front of all states orthogonal to $|\psi\rangle$, and then sends back Alice's qubits. This has a communication cost of $O(\log n)$ qubits, which is too much for our purposes. Fortunately, Theorem 8.6 gives us a way to implement a one-sided ε-error reflection protocol R_ψ^ε that only costs $O(\log(1/\varepsilon))$ qubits of communication.

The reflection $O_{\mathscr{G}}$ puts a '$-$' in front of the basis states $|i\rangle|j\rangle$ in $\mathscr{G}$. This can be implemented perfectly using only 2 qubits of communication, as follows. To implement this reflection on her basis state $|i\rangle$, Alice XORs $|x_i\rangle$ into a fresh auxiliary $|0\rangle$-qubit and sends this qubit to Bob. Bob receives this qubit and applies the following unitary map:

$$|b\rangle|j\rangle \mapsto y_j^b|b\rangle|j\rangle, \quad b \in \{0,1\}, j \in [n].$$

He sends back the auxiliary qubit. Alice sets the auxiliary qubit back to $|0\rangle$ by XORing x_i into it. Ignoring the auxiliary qubit (which starts and ends in state $|0\rangle$), this maps $|i\rangle|j\rangle \mapsto (-1)^{[x_i=y_j=-1]}|i\rangle|j\rangle$. Hence we have implemented $O_{\mathscr{G}}$ correctly: a minus sign is applied exactly for the good basis states, the ones where $x_i = y_j = -1$.

Now consider the algorithms (more precisely, communication protocols):

$$A_1 = R_\psi^{\varepsilon_1} \cdot O_{\mathscr{G}} \text{ and } A_{j+1} = A_j R_\psi^{\varepsilon_{j+1}} A_j^* \cdot O_{\mathscr{G}} \cdot A_j$$

with the choice of ε_j's from Equation (8.4). If we pick $k = \lfloor \log_3(\pi/(2\theta)) \rfloor$, like in Equation (8.2), then $3^k\theta \in (\pi/6, \pi/2]$. Hence by Equation (8.3) and Equation (8.5), the inner product of our final state with $|G\rangle$ will be between $\sin(3^k\theta) - 3\theta/100 \geq 0.4$ and 1.

At this point Alice and Bob can measure, and with probability $\geq 0.4^2$ they will each see the same i, with the property that $x_i = y_i = -1$.

From Equation (8.1) and Theorem 8.6, the recursion for the communication costs of these algorithms is

$$C_{j+1} = 3C_j + O(\log(1/\varepsilon_{j+1})) + 2.$$

Solving this recurrence with our ε_j's from Equation (8.4) and the value of θ from Equation (8.6) we obtain

$$C_k = \sum_{j=1}^{k} 3^{k-j}(O(\log(1/\varepsilon_j))+2) = \sum_{j=1}^{k} 3^{k-j}O(j) = O(3^k) = O(\sqrt{n/t}).$$

Thus, using $O(\sqrt{n/t})$ qubits of communication we can find (with constant success probability) an intersection point i. This also allows us to solve the Set-Disjointness problem (the decision problem whose output is 1 if there is no intersection between x and y). Note that if the t we used equals the actual number of solutions only up to a factor of 2, the above protocol still has $\Omega(1)$ probability to find a solution, and $O(1)$ repetitions will boost this success probability to 0.99. In case we do not even know t approximately, we can use the standard technique of trying exponentially decreasing guesses for t to find an intersection point with communication $O(\sqrt{n})$.

Note that there is no log-factor in the communication complexity, in contrast to the original $O(\sqrt{n}\log n)$-qubit Grover-based quantum protocol for the intersection problem of Buhrman et al. [34]. Aaronson and Ambainis [1] earlier already managed to remove the log-factor, giving an $O(\sqrt{n})$-qubit protocol for Set-Disjointness as a consequence of their local version of quantum search on a grid graph (which is optimal [126]). We have just reproved this result of [1] in a different and arguably simpler way.

The above description is geared towards the intersection problem, where the "inner" function is $G = \mathsf{AND}_2$: we called a basis state $|i\rangle|j\rangle$ "good" if $x_i = y_j = -1$. However, this can easily be generalized to the situation where Alice and Bob's respective inputs are $X = (X_1,\ldots,X_n)$ and $Y = (Y_1,\ldots,Y_n)$ and we want to find an $i \in [n]$ where $G(X_i,Y_i) = -1$ for some two-party function G, and define the set of "good" basis states as $\mathscr{G} = \{|i\rangle|j\rangle \mid G(X_i,Y_j) = -1\}$.[1] The only thing that changes in the above is the implementation of the reflection $O_{\mathscr{G}}$, which would now be computed by means of an exact quantum communication protocol for $G(X_i,Y_j)$, at a cost of $2\mathsf{Q}_E^{cc}(G)$ qubits of communication.[2] Note that because we can check (at the expense of another $\mathsf{Q}_E^{cc}(G)$ qubits of communication) whether the output index i actually satisfies $G(X_i,Y_i) = -1$, we may assume the protocol has one-sided error: it always outputs "no" if there is no such i. This concludes the proof of Theorem 8.8.

[1] We intentionally use the letter 'G' to mean "good" in $\mathscr{G}$ and and to refer to the two-party function G, since G determines which basis states $|i\rangle|j\rangle$ are "good."

[2] The factor of 2 is to reverse the protocol after the phase $G(X_i,Y_j)$ has been added to basis state $|i\rangle|j\rangle$, in order to set any workspace qubits back to $|0\rangle$.

8.4 No log-factor needed for symmetric functions

In this section we prove Theorem 8.1 from the introduction. Consider a symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$. As explained in Section 8.2, there is an integer $t = \lceil (n - \Gamma(f))/2 \rceil$ such that we can compute f if we learn the Hamming weight $|z|$ of the input $z \in \{-1,1\}^n$ or learn that $|z| \in [t+1, n-t-1]$. The bounded-error quantum query complexity is $\mathrm{Q}(f) = \Theta(\sqrt{tn})$ (Theorem 8.5).

For a given two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{-1,1\}$, we have an induced two-party function $F : \{-1,1\}^{nj} \times \{-1,1\}^{nk} \to \{-1,1\}$ defined as $F(X_1, \ldots, X_n, Y_1, \ldots, Y_n) = f(G(X_1,Y_1), \ldots, G(X_n,Y_n))$. Define

$$z = (G(X_1,Y_1), \ldots, G(X_n,Y_n)) \in \{-1,1\}^n.$$

Then $F(X,Y) = f(z)$ only depends on the number of -1s in z. The following theorem allows us to count this number using $O(\mathrm{Q}(f)\,\mathrm{Q}_E^{cc}(G))$ qubits of communication.

Theorem 8.9. *For every t between 1 and $n/2$, there exists a quantum protocol that starts from $O(t \log n)$ EPR-pairs, communicates $O(\sqrt{tn}\,\mathrm{Q}_E^{cc}(G))$ qubits, and tells us $|z|$ or tells us that $|z| > t$, with error probability $\leq 1/8$.*

Proof. Abbreviate $q = \mathrm{Q}_E^{cc}(G)$. Our protocol has two parts: the first filters out the case $|z| \geq 2t$, while the second finds all solutions if $|z| < 2t$.

Part 1. First Alice and Bob decide between the case (1) $|z| \geq 2t$ and (2) $|z| \leq t$ (even though $|z|$ might also lie in $\{t+1, \ldots, 2t-1\}$) using $O(\sqrt{nq})$ qubits of communication, as follows. They use shared randomness to choose a uniformly random subset $S \subseteq [n]$ of $\lceil n/(2t) \rceil$ elements. Let E be the event that $z_i = -1$ for at least one $i \in S$. By standard calculations there exist $p_1, p_2 \in [0,1]$ with $p_1 = p_2 + \Omega(1)$ such that $\Pr[E] \geq p_1$ in case (1) and $\Pr[E] \leq p_2$ in case (2). Alice and Bob use the distributed-search protocol from the first bullet of Theorem 8.8 to decide E, with $O(\sqrt{|S|}\,q) = O(\sqrt{n}\,q)$ qubits of communication (plus a negligible $O(\log n)$ EPR-pairs) and error probability much smaller than $p_1 - p_2$. By repeating this a sufficiently large constant number of times and seeing whether the fraction of successes was larger or smaller than $(p_1 + p_2)/2$, they can distinguish between cases (1) and (2) with success probability $\geq 15/16$. If they conclude they're in case (1) then they output "$|z| > t$" and otherwise they proceed to the second part of the protocol.

Note that if $|z| \in \{t+1, \ldots, 2t-1\}$ (the "grey zone" in between cases (1) and (2)), then we can't give high-probability guarantees for one output or the other, but concluding (1)

leads to the correct output "$|z| > t$" in this case, while concluding (2) means the protocol proceeds to Part 2. So either course of action is fine if $|z| \in \{t+1, \ldots, 2t-1\}$.

By Newman's theorem [119] the shared randomness used for choosing S can be replaced by $O(\log n)$ bits of private randomness on Alice's part, which she can send to Bob in her first message, so Part 1 communicates $O(\sqrt{n}\,q)$ qubits in total.

Part 2. We condition on Part 1 successfully filtering out case (1), so from now on assume $|z| < 2t$. Our goal in this second part of the protocol is to find all indices i such that $z_i = -1$ (we call such i "solutions"), with probability $\geq 15/16$, using $O(\sqrt{tn}\,q)$ qubits of communication. This will imply that the overall protocol is correct with probability $1 - 1/16 - 1/16 = 7/8$, and uses $O(\sqrt{tn}\,q)$ qubits of communication in total. For an integer $k \geq 1$, consider the following protocol P_k.

Algorithm 1: Protocol P_k

Input: An integer $k \geq 1$

repeat

1. Run the protocol from the last bullet of Theorem 8.8 with $t = 2^{k-1}$. (suppressing some constant factors, assume for simplicity that this uses $\sqrt{n/2^k}\,q$ qubits of communication, $\log n$ shared EPR-pairs at the start, and has probability $\geq 1/100$ to find a solution if the actual number of solutions is in $[t/2, 2t]$).
2. Alice measures and gets outcome $i \in [n]$ and Bob measures and gets outcome $j \in [n]$, respectively.
3. Alice sends i to Bob, Bob sends j to Alice.
4. If $i = j$ then they verify that $G(X_i, Y_i) = -1$ by one run of the protocol for G, and if so then they replace X_i, Y_i by some pre-agreed inputs X_i', Y_i', respectively, such that $G(X_i', Y_i') = 1$ (this reduces the number of -1s in z by 1)

until $200\sqrt{2^k n}\,q$ *qubits have been sent*;

Claim 8.10. *Suppose $|z| \in [2^{k-1}, 2^k)$. Then protocol P_k uses $O(\sqrt{2^k n}\,q)$ qubits of communication, assumes $O(2^k \log n)$ EPR-pairs at the start of the protocol, and finds at least $|z| - 2^{k-1} + 1$ solutions, except with probability $\leq 1/2$.*

Proof. The upper bound on the communication is obvious from the stopping criterion of P_k.

As long as the remaining number of solutions is $\geq 2^{k-1}$, each run of the protocol has probability $\geq 1/100$ to find another solution. Hence the expected number of runs of the protocol of Theorem 8.8 to find at least $|z| - 2^{k-1} + 1$ solutions, is $\leq 100(|z| - 2^{k-1} + 1)$. By

Markov's inequality, the probability that we haven't yet found $|z| - 2^{k-1} + 1$ solutions after $\leq 200(|z| - 2^{k-1} + 1) \leq 100 \cdot 2^k$ runs, is $\leq 1/2$. The communication cost of so many runs is $100 \cdot 2^k(\sqrt{n/2^k}\,q + \log n) \leq 200\sqrt{2^k n}\,q$ qubits. Hence by the time that the number of qubits of the stopping criterion have been communicated, we have probability $\geq 1/2$ of having found at least $|z| - 2^{k-1} + 1$ solutions. The assumed number of EPR-pairs at the start is $\log n$ per run, so $O(2^k \log n)$ in total. □

Note that if we start with a number of solutions $|z| \in [2^{k-1}, 2^k)$, and P_k succeeds in finding at least $|z| - 2^{k-1} + 1$ new solutions, then afterwards we have $< 2^{k-1}$ solutions left. The following protocol runs these P_k in sequence, pushing down the remaining number of solutions to 0.

Algorithm 2: Protocol P

for $k = \lceil \log_2(2t) \rceil$ downto 1 **do**

1. Run P_k a total of $r_k = \lceil \log_2(2t) \rceil - k + 5$ times (replacing all -1s found by $+1$s in z).
2. Output the total number of solutions found.

end

Claim 8.11. *If $|z| < 2t$ then protocol $\mathscr{P}$ uses $O(\sqrt{tn}\,q)$ qubits of communication, assumes $O(t \log n)$ EPR-pairs at the start of the protocol, and outputs $|z|$, except with probability $\leq 1/16$.*

Proof. First, by Claim 8.10, the total number of qubits communicated is

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} r_k \cdot O(\sqrt{2^k n}\,q) = O(\sqrt{tn}\,q) \cdot \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} (\ell + 5)/\sqrt{2^\ell} = O(\sqrt{tn}\,q),$$

where we used a variable substitution $k = \lceil \log_2(2t) \rceil - \ell$. Second, the number of EPR-pairs we're starting from is

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} r_k \cdot O(2^k \log n) = O(t \log n) \cdot \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} (\ell + 5)/2^\ell = O(t \log n).$$

Third, by Claim 8.10 and the fact that we are performing r_k repetitions of P_k, if the kth round of $\mathscr{P}$ starts with a remaining number of solutions that is in the interval $[2^{k-1}, 2^k)$ then that round ends with $< 2^{k-1}$ remaining solutions, except with probability at most $1/2^{r_k}$. By the union bound, the probability that any one of the $\lceil \log_2(2t) \rceil$ rounds does not succeed at this,

is at most

$$\sum_{k=1}^{\lceil \log_2(2t) \rceil} \frac{1}{2^{r_k}} = \sum_{\ell=0}^{\lceil \log_2(2t) \rceil - 1} \frac{1}{2^{\ell+5}} \leq \frac{1}{16}.$$

Since $2^{\lceil \log_2(2t) \rceil} \geq 2t$ and we start with $|z| < 2t$, if each round succeeds, then by the end of $\mathscr{P}$ there are no remaining solutions left. Thus, the protocol $\mathscr{P}$ finds all solutions and learns $|z|$ with probability at least $15/16$. □

Part 1 and Part 2 each have error probability $\leq 1/16$, so by the union bound the protocol succeeds except with probability $1/8$. If $|z| \geq 2t$ then Part 1 outputs the correct answer "$|z| > t$"; if $|z| \leq t$ then all solutions (and hence $|z|$) are found by Part 2; and if $|z| \in \{t+1, \ldots, 2t-1\}$ then either Part 1 already outputs the correct answer "$|z| > t$" or the protocol proceeds to Part 2 which then finds all solutions. □

We can use the above theorem twice: once to count the number of -1s in z (up to t) and once to count the number of 1s in z (up to t). This uses $O(\sqrt{tn}\,\mathsf{Q}_E^{cc}(G)) = O(\mathsf{Q}(f)\,\mathsf{Q}_E^{cc}(G))$ qubits of communication, assumes $O(t \log n)$ shared EPR-pairs at the start of the protocol, and gives us enough information about $|z|$ to compute $f(z) = F(X,Y)$. This concludes the proof of Theorem 8.1 from the introduction, restated below.

Theorem 8.12 (Restatement of Theorem 8.1)**.** *For every symmetric Boolean function $f : \{-1,1\}^n \to \{-1,1\}$ and two-party function $G : \{-1,1\}^j \times \{-1,1\}^k \to \{0,1\}$, we have*

$$\mathsf{Q}^{cc,*}(f \circ G) = O(\mathsf{Q}(f)\mathsf{Q}_E^{cc}(G)).$$

If $\mathsf{Q}(f) = \Theta(\sqrt{tn})$, then our protocol assumes a shared state of $O(t \log n)$ EPR-pairs at the start.

We remark that for the special case where $G = \mathsf{AND}_2$, our upper bound matches the lower bound proved by Razborov [126], except for symmetric functions f where the first switch of function value happens at Hamming weights very close to n. In particular, if $f = \mathsf{AND}_n$ and $G = \mathsf{AND}_2$, then $\mathsf{Q}^{cc}(f \circ G) = 1$ but $\mathsf{Q}(f) = \Theta(\sqrt{n})$.

8.5 Overhead required for transitive functions

The following claim shows that the function in Section 7.3, Theorem 7.2 is in fact transitive.

Claim 8.13. *Let $n > 0$ be a power of 2. Let $r = \mathsf{PARITY}_n : \{-1,1\}^n \to \{-1,1\}$ and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$. The function $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is transitive.*

Proof. We first show that $h_G : \{-1,1\}^{2n} \to \{-1,1\}$ is transitive. We next observe that $s \mathbin{\widetilde{\circ}} t$ is transitive whenever s is symmetric and t is transitive (s can be assumed to just be transitive rather than symmetric, as noted in Remark 6). The theorem then follows since PARITY_n is symmetric.

Towards showing transitivity of h_G, let $\pi \in S_{2n}$, and $(\sigma_\ell, \sigma_\ell) \in S_{2n}$ for $\ell \in \{-1,1\}^{\log n}$ be defined as follows. (Here $\sigma_\ell \in S_n$; the first copy acts on the first n coordinates, and the second copy acts on the next n coordinates.)

- $$\pi(k) = \begin{cases} k+n & k \le n \\ k-n & k > n. \end{cases}$$

  That is, on every string $(x,y) \in \{-1,1\}^{2n}$, the permutation π maps (x,y) to (y,x).

- For every $\ell \in \{-1,1\}^{\log n}$, the permutation $\sigma_\ell \in S_n$ is defined as

  $$\sigma_\ell(i) = i \oplus \ell, \tag{8.7}$$

  where $i \oplus \ell$ denotes the bitwise XOR of the strings i and ℓ. That is, for every input $(x,y) \in \{-1,1\}^{2n}$ and every $k \in \{-1,1\}^{\log n}$, the input bit x_k is mapped to $x_{k\oplus\ell}$ and y_k is mapped to $y_{k\oplus\ell}$.

For every $(x,y) \in \{-1,1\}^{2n}$ and $i,j \in \{-1,1\}^{\log n}$, the permutation $\sigma_{i\oplus j}(x,y)$ swaps x_i and x_j, and also swaps y_i and y_j. If for $i,j \in \{-1,1\}^{\log n}$, our task was to swap the i'th index of the first n variables with the j'th index of the second n variables, then the permutation $\sigma_{i\oplus j} \circ \pi$ does the job. That is, for every $(x,y) \in \{-1,1\}^{2n}$ and $i,j \in \{-1,1\}^{\log n}$, the permutation $\sigma_{i\oplus j} \circ \pi$ maps x_i to y_j. Thus the set of permutations $\{\pi, \{\sigma_\ell : \ell \in \{-1,1\}^{\log n}\}\}$ acts transitively on S_{2n}.

Now we show that for all $x,y \in \{-1,1\}^{2n}$ and all $\ell \in \{-1,1\}^{\log n}$, we have $h_G(\sigma_\ell(x), \sigma_\ell(y)) = h_G(x,y)$. Fix $\ell \in \{-1,1\}^{\log n}$.

- If $x \in \pm H(s)$ and $y \in \pm H(t)$ are Hadamard codewords, then $x_k = \langle k,s\rangle$ and $y_k = \langle k,t\rangle$ for all $k \in \{-1,1\}^{\log n}$, and $G(x,y) = \langle s,t\rangle$. Thus, for every $k \in \{-1,1\}^{\log n}$ we have $\sigma_\ell(x_k) = x_{k\oplus\ell} = \langle k \oplus \ell, s\rangle = \langle \ell, s\rangle \cdot \langle k,s\rangle$. Hence $\sigma_\ell(x) \in \pm H(s)$ (since $\langle \ell, s\rangle$ does

not depend on k, and takes value either 1 or -1). Similarly, $\sigma_\ell(y) \in \pm H(t)$. Thus $h_G(\sigma_\ell(x,y)) = h_G(x,y)$.

- If x (y, respectively) is not a Hadamard codeword, then a similar argument shows that for all $\ell \in [n]$, $\sigma_\ell(x)$ ($\sigma_\ell(y)$, respectively) is also not a Hadamard codeword.

Using the fact that $\langle s,t\rangle = \langle t,s\rangle$ for every $s,t \in \{-1,1\}^{\log n}$, one may verify that $h_G(\pi(x,y)) = h_G(x,y)$ for all $x,y \in \{-1,1\}^{2n}$.

Along with the observation that PARITY_n is a symmetric function, we have that $f = r \mathbin{\widetilde{\circ}} h_G : \{-1,1\}^{2n^2} \to \{-1,1\}$ is transitive under the following permutations:

- S_n acting on the inputs of PARITY_n, and
- The group generated by $\{\pi\} \cup \{(\sigma_\ell, \sigma_\ell) : \ell \in [n]\}$ acting independently on the inputs of each copy of h_G, where σ_ℓ is as in Equation (8.7).

□

Remark 6. The proof of transitivity of f in Theorem 8.2 can also be used to prove that if $r : \{-1,1\}^n$ is transitive and $G = \mathsf{IP}_{\log n} : \{-1,1\}^{\log n} \times \{-1,1\}^{\log n} \to \{-1,1\}$, then $r \mathbin{\widetilde{\circ}} h_G$ is transitive as well. By instantiating r to a transitive function with approximate degree $\Omega(n)$ (e.g. Majority), Theorem 7.20 implies that the BCW simulation is tight w.r.t. the bounded-error communication model for a wide class of transitive functions.

8.6 Summary

This chapter explored the role of symmetry in query-to-communication problem in the setting of quantum communication. We showed that when the outer function is symmetric, no overhead is required to simulate query algorithms to get a communication cost of almost same cost. On the other hand, such a claim does not hold true when we relax the notion of symmetry and talk about transitive outer functions.

Part III

Classical Query and Communication Complexity

Chapter 9

Set Disjointness and VC-Dimension

9.1 Introduction

A striking feature of communication complexity is its interplay with other diverse areas like analysis, combinatorics, and geometry [102, 128]. Vapnik and Chervonenkis [146] introduced the measure *Vapnik-Chervonenkis dimension* or *VC dimension* for set systems in the context of statistical learning theory. As was the case with communication complexity, VC dimension has found numerous connections and applications in many different areas like approximation algorithms, discrete and combinatorial geometry, computational geometry, discrepancy theory and many other areas [111, 44, 122, 112]. In this chapter we study both of them under the same lens: of restricted systems and, for the first time, prove that geometric simplicity does not necessarily imply efficient communication complexity.

Let us start with recollecting some definitions from Vapnik–Chervonenkis theory. Let $\mathscr{S}$ be a collection of subsets of a *universe* $\mathscr{U}$. For a subset y of $\mathscr{U}$, we define

$$\mathscr{S}|_y := \{y \cap x : x \in \mathscr{S}\}.$$

We say a subset y of $\mathscr{U}$ is *shattered* by $\mathscr{S}$ if $\mathscr{S}|_y = 2^y$, where 2^y denotes the power set of y. *Vapnik–Chervonenkis (VC) dimension* of $\mathscr{S}$, denoted as VC-dim$(\mathscr{S})$, is the size of the largest subset y of $\mathscr{U}$ shattered by $\mathscr{S}$. VC dimension has been one of the fundamental measures for quantifying complexity of a collection of subsets.

Now let us revisit the world of communication complexity. Let $f : \Omega_1 \times \Omega_2 \to \Omega$. In *communication complexity*, two players Alice and Bob get as inputs $x \in \Omega_1$ and $y \in \Omega_2$

respectively, and the goal for the players is to devise a protocol to compute $f(x,y)$ by exchanging as few bits of information between themselves as possible.

The *deterministic communication complexity* $D(f)$ of a function f is the minimum number of bits Alice and Bob will exchange in the worst case to deterministically compute the function f. In the randomized setting, both Alice and Bob share an infinite random source[1] and the goal is to give the correct answer with probability at least $2/3$. The randomized communication complexity $R(f)$ of f denotes the minimum number of bits exchanged by the players in the worst case input by the best randomized protocol computing f. In both deterministic and randomized settings, Alice and Bob are allowed to make multiple rounds of interaction. Communication complexity when the number of rounds of interaction is bounded is also often studied. An important special case is when only one round of communication is allowed, that is, only Alice is allowed to send messages to Bob and Bob computes the output. We will denote by $D^{\rightarrow}(f)$ and $R^{\rightarrow}(f)$ the *one way deterministic communication complexity* and *one way randomized communication complexity* respectively, of f.

One of the most well studied functions in communication complexity is the disjointness function. Given a universe $\mathscr{U}$ known to both Alice and Bob, the *disjointness function*, $\mathsf{DISJ}_{\mathscr{U}} : 2^{\mathscr{U}} \times 2^{\mathscr{U}} \rightarrow \{0,1\}$, where $2^{\mathscr{U}}$ denotes the power set of $\mathscr{U}$, is defined as follows:

$$\mathsf{DISJ}_{\mathscr{U}}(x,y) = \begin{cases} 1, & \text{if } x \cap y = \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{9.1}$$

We also define the *intersection function*. Given a universe $\mathscr{U}$ known to both Alice and Bob, the *intersection function*, $\mathsf{INT}_{\mathscr{U}} : 2^{\mathscr{U}} \times 2^{\mathscr{U}} \rightarrow 2^{\mathscr{U}}$ is defined as $\mathsf{INT}_{\mathscr{U}}(x,y) = x \cap y$. It is easy to see that $\mathsf{INT}_{\mathscr{U}}$ is harder function to compute than $\mathsf{DISJ}_{\mathscr{U}}$. The $\mathsf{DISJ}_{\mathscr{U}}$ function and its different variants, like $\mathsf{INT}_{\mathscr{U}}$, have been one of the most important problems in communication complexity and have found numerous applications in areas like streaming algorithms for proving lower bounds [128, 125]. By abuse of the notation, when $\mathscr{U} = [n]$, where $[n]$ denotes the set $\{1,\ldots,n\}$, we will denote the functions $\mathsf{DISJ}_{[n]}$ and $\mathsf{INT}_{[n]}$ by DISJ_n and INT_n respectively.

Using the standard *rank argument* [102, 125] one can show that $D(\mathsf{DISJ}_n) = \Theta(n)$. In a breakthrough paper, Kalyanasundaram and Schnitger [93] proved that $R(\mathsf{DISJ}_n) = \Omega(n)$. Razborov [127] and Bar-Yossef et al. [17] gave alternate proofs for the above result. From the

[1] This is the communication complexity setting with shared random coins. If no random string is shared, it is called the private random coins setting. By [118] we know that the communication complexity in both the setting differs by at most a logarithmic additive factor.

above cited results we can also see the $D(\mathsf{INT}_n) = R(\mathsf{INT}_n) = \Theta(n)$. $R(\mathsf{DISJ}_n) = R(\mathsf{INT}_n) = \Theta(n)$ also follow from a recent result by Braverman et al. [29].

Naturally, one would also like to ask what happens to the deterministic and randomized communication complexity (one way or multiple round) of DISJ_n, when both Alice and Bob know that their inputs have more structure. In particular what can we say if the inputs are guaranteed to be from a subset of $\mathscr{S} \subseteq 2^{\mathscr{U}}$, where $\mathscr{S}$ is known to both players. Let $\mathsf{DISJ}_{\mathscr{U}}$ functions *restricted* to $\mathscr{S} \times \mathscr{S}$ be denoted by $\mathsf{DISJ}_{\mathscr{U}} \mid_{\mathscr{S}\times\mathscr{S}}$. This problem has also been studied extensively, mostly for certain special classes of subsets $\mathscr{S} \subseteq 2^{\mathscr{U}}$. For example, the *sparse set disjointness* function, where the set $\mathscr{S}$ contains all the subsets of $\mathscr{U}$ of size at most d, is an important special case of these works.

We will denote by d-SPARSEDISJ$_n$ and d-SPARSEINT$_n$, the functions $\mathsf{DISJ}_n \mid_{S\times\mathscr{S}}$ and $\mathsf{INT}_n \mid_{S\times S}$ respectively, where S is the collections of all subsets of $[n]$ of size at most d. Using the *rank argument* [102, 125], one can again show that, for all $d \leq n$, the deterministic communication complexity of d-SPARSEDISJ$_n$ is $\Omega\left(d \log \frac{n}{d}\right)$. Håstad and Wigderson [82], and Dasgupta et al. [48] showed that the randomized communication complexity and one way randomized communication complexity of d-SPARSEDISJ$_n$ is $R(d\text{-SPARSEDISJ}_n) = \Theta(d)$ and $R^{\rightarrow}(d\text{-SPARSEDISJ}_n) = \Theta(d \log d)$ respectively. In a follow up work, Saglam and Tardos [131] proved that with $O(\log^* d)$ rounds of communication and $O(d)$ bits of communication it is possible to compute d-SPARSEDISJ$_n$. More recently, Brody et al. [30] proved that $R^{\rightarrow}(d\text{-SPARSEINT}_n) = \Theta(d \log d)$ and $R(d\text{-SPARSEINT}_n) = \Theta(d)$. These results show that in the d-sparse setting, there is a separation between randomized and deterministic communication complexity of DISJ_n and INT_n functions.

One would like to ask what happens to the communication complexity for other restrictions to the disjointness (and intersection) problem. The following are two natural problems, with a geometric flavor, for which one would like to study the communication complexity.

Problem 9.1 (DISCRETE LINE DISJ). Let $G \subset \mathbb{Z}^2$ be a set of n points in $\mathbb{Z}^2$ and L be the set of all lines in $\mathbb{R}^2$. Also, let $\mathscr{L} = L^d$ denote the collection of all d-size subsets of L. The DISCRETE LINE DISJ on G and $\mathscr{L}$ is a function, $\mathsf{DISJ}_G \mid_{\mathscr{L}\times\mathscr{L}}: \mathscr{L} \times \mathscr{L} \to \{0,1\}$ defined as $\mathsf{DISJ}_G \mid_{\mathscr{L}\times\mathscr{L}} \left(\{\ell_1,\ldots,\ell_d\},\{\ell'_1,\ldots,\ell'_d\}\right)$ is 1 if and only if there exists a line in Alice's set[2] that intersects some line in Bob's set at some point in G. Formally,

$$\mathsf{DISJ}_G \mid_{\mathscr{L}\times\mathscr{L}} \left(\{\ell_1,\ldots,\ell_d\},\{\ell'_1,\ldots,\ell'_d\}\right) = \begin{cases} 1, & \text{if } \exists i,j \in [d] \text{ s.t. } \ell_i \cap \ell'_j \cap G \neq \emptyset \\ 0, & \text{otherwise} \end{cases} \tag{9.2}$$

[2]We assume that Alice has the set $\{\ell_1,\ldots,\ell_d\}$ and Bob has the set $\{\ell'_1,\ldots,\ell'_d\}$.

Problem 9.2 (DISCRETE INTERVAL DISJ). Let $X \subset \mathbb{Z}$ be a set of n points in $\mathbb{Z}$ and *Int* be the set of all possible intervals. Also, let $\mathscr{I} = Int^d$ denote the collection of all d-size subsets of *Int*. The DISCRETE INTERVAL DISJ on X and $\mathscr{I}$ is a function, $\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}: \mathscr{I}\times\mathscr{I} \to \{0,1\}$ defined as $\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}\left(\{I_1,\ldots,I_d\},\{I'_1,\ldots,I'_d\}\right)$ is 1 if and only if there exists an interval in Alice's set[3] that intersects some interval in Bob's set at some point in X.

$$\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}\left(\{I_1,\ldots,I_d\},\{I'_1,\ldots,I'_d\}\right) = \begin{cases} 1, & \text{if } \exists i,j\in[d] \text{ s.t. } I_i\cap I'_j\cap X\neq\emptyset \\ 0, & \text{otherwise} \end{cases} \tag{9.3}$$

Note that both the DISCRETE LINE DISJ and DISCRETE INTERVAL DISJ functions are generalizations of sparse set disjointness function.[4] Although it may not be obvious at first look, but both the DISCRETE LINE DISJ function and the DISCRETE INTERVAL DISJ functions are disjointness functions restricted to a suitable subset. In fact, the connection between the *Sparse set disjointness* function (d-SPARSEDISJ$_n$), the DISCRETE LINE DISJ function and the DISCRETE INTERVAL DISJ function run deep - all the three subsets of the domain which help to define the functions as restriction of the disjointness function have VC dimension $\Theta(d)$, see Section 9.4. Naturally one would like to know, if the fact that the collection of subsets $\mathscr{S}$ has VC dimension d has any implication on the communication complexity of $\mathsf{DISJ}_{\mathscr{U}}|_{\mathscr{S}\times\mathscr{S}}$. For example, is the randomized communication complexity of DISCRETE LINE DISJ function and the DISCRETE INTERVAL DISJ function upper bounded by a function of d (independent of n)? And, do the DISCRETE LINE DISJ function and the DISCRETE INTERVAL DISJ function also have a separation between their randomized and deterministic communication complexities similar to that of the *Sparse set disjointness* function (d-SPARSEDISJ$_n$)? We show that these are not necessarily the cases.

9.1.1 Results and Organization

We now state two the main theorems of this chapter.

Theorem 9.3. *For* DISCRETE LINE DISJ*: there exists a $G\subset\mathbb{Z}^2$ with n points such that $D(\mathsf{DISJ}_G|_{\mathscr{L}\times\mathscr{L}}) = D^{\to}(\mathsf{DISJ}_G|_{\mathscr{L}\times\mathscr{L}}) = \Theta\left(d\log\frac{n}{d}\right)$ and, for the randomized setting,*

$$R(\mathsf{DISJ}_G|_{\mathscr{L}\times\mathscr{L}}) = \Omega\left(d\frac{\log(n/d)}{\log\log(n/d)}\right)$$

[3] We assume that Alice has the set $\{I_1,\ldots,I_d\}$ and Bob has the set $\{I'_1,\ldots,I'_d\}$.

[4] Take n integer points on the x-axis. For DISCRETE LINE DISJ setting, restrict only to lines orthogonal to x-axis. For DISCRETE INTERVAL DISJ setting, take n integer points on $\mathbb{Z}$ and only restrict to intervals containing one integer point. Both of these restriction will give the disjointness problem in the d-sparse setting.

Theorem 9.4. *For* DISCRETE INTERVAL DISJ*: there exists a $X \subset \mathbb{Z}$ with n points such that*

$$D(\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}) = D^{\rightarrow}(\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}) = R^{\rightarrow}(\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}) = \Theta\left(d\log\frac{n}{d}\right).$$

DISCRETE LINE INT, that is, the intersection finding version of DISCRETE LINE DISJ is defined as follows : the objective is to compute a function $\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}\colon \mathscr{L}\times\mathscr{L}\to G$ that is defined as

$$\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}(\{\ell_1,\ldots,\ell_d\},\{\ell'_1,\ldots,\ell'_d\}) = \bigcup_{i,j\in[d]}\left(\ell_i\cap\ell'_j\cap G\right).$$

As we have already mentioned, Brody et al. [30] proved that $R(d\text{-SPARSEINT}_n) = \Theta(d)$, whereas $D(d\text{-SPARSEINT}_n) = \Theta\left(d\log\frac{n}{d}\right)$. We show that DISCRETE LINE INT does not demonstrate such a separation between the deterministic and randomized communication complexity.

Theorem 9.5. *For* DISCRETE LINE INT*: there exists a $G\subset\mathbb{Z}^2$ with n points such that*

$$D(\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}) = D^{\rightarrow}(\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}) = R^{\rightarrow}(\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}) = R(\mathsf{INT}_G|_{\mathscr{L}\times\mathscr{L}}) = \Theta\left(d\log\frac{n}{d}\right).$$

The upper bound for all the above three theorems can be obtained from the fact that the corresponding sets have VC dimension $\Theta(d)$, see Section 9.4. *Sauer-Shelah Lemma* [134, 137, 146] states that if $\mathscr{S}\subseteq 2^{[n]}$ and VC-dim$(\mathscr{S})\le d$ then

$$|\mathscr{S}| \le \sum_{i=0}^{d}\binom{n}{i} \le \left(\frac{en}{d}\right)^d.$$

Thus if VC-dim$(\mathscr{S})\le d$, then the Sauer-Shelah Lemma implies that

$$D^{\rightarrow}(\mathsf{INT}_n|_{\mathscr{S}\times\mathscr{S}}) = O\left(d\log\frac{n}{d}\right).$$

So, $O\left(d\log\frac{n}{d}\right)$ is a upper bound to the above questions, both for randomized and deterministic and also for the one-way communication. But can the randomized communication complexity of $\mathsf{DISJ}_{\mathscr{U}}|_{\mathscr{S}\times\mathscr{S}}$ and $\mathsf{INT}_{\mathscr{U}}|_{\mathscr{S}\times\mathscr{S}}$ be even lower when S has VC dimension d? The following result, which is a direct consequence of Theorems 9.4, 9.3 and 9.5, enables us to we answer the question in the negative:

Theorem 9.6. *Let $1\le d\le n$.*

1. *There exists $\mathscr{S}\subseteq 2^{[n]}$ with VC-dim$(\mathscr{S})\le d$ and $R^{\rightarrow}(\mathsf{DISJ}_n|_{\mathscr{S}\times\mathscr{S}}) = \Omega\left(d\log\frac{n}{d}\right)$.*

2. *There exists* $\mathscr{S} \subseteq 2^{[n]}$ *with VC-dim*$(\mathscr{S}) \leq d$ *and* $R(\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}}) = \Omega\left(d\frac{\log(n/d)}{\log\log(n/d)}\right)$.

3. *There exists* $\mathscr{S} \subseteq 2^{[n]}$ *with VC-dim*$(\mathscr{S}) \leq d$ *and* $R(\mathsf{INT}_n\,|_{\mathscr{S}\times\mathscr{S}}) = \Omega\left(d\log\frac{n}{d}\right)$.

The following table compares our result with the previous best known lower bound for $\mathsf{DISJ}_{\mathscr{U}}\,|_{\mathscr{S}\times\mathscr{S}}$ and $\mathsf{INT}_{\mathscr{U}}\,|_{\mathscr{S}\times\mathscr{S}}$ among all sets $S \subset 2^{\mathscr{U}}$ of VC dimension d.

Problems	$R(\mathsf{DISJ}_n\,\|_{\mathscr{S}\times\mathscr{S}})$	$R^{\rightarrow}(\mathsf{DISJ}_n\,\|_{\mathscr{S}\times\mathscr{S}})$	$R(\mathsf{INT}_n\,\|_{\mathscr{S}\times\mathscr{S}})$	$R^{\rightarrow}(\mathsf{INT}_n\,\|_{\mathscr{S}\times\mathscr{S}})$
Known	$\Omega(d)$ [82]	$\Omega(d\log d)$ [48]	$\Omega(d)$ [30]	$\Omega(d\log d)$ [30]
Our Work	$\Omega\left(d\frac{\log(n/d)}{\log\log(n/d)}\right)$	$\Omega\left(d\log\frac{n}{d}\right)$	$\Omega\left(d\log\frac{n}{d}\right)$	$\Omega\left(d\log\frac{n}{d}\right)$

Table 9.1 The largest communication complexity, for the functions $\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}}$ and $\mathsf{INT}_n\,|_{\mathscr{S}\times\mathscr{S}}$, among all $S \subseteq 2^{[n]}$ of VC dimension d, that was previously known and what we prove in this chapter. Tight bounds of $\Omega\left(d\log\frac{n}{d}\right)$ for the largest $D(\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}})$, $D^{\rightarrow}(\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}})$, $D(\mathsf{INT}_n\,|_{\mathscr{S}\times\mathscr{S}})$ and $D^{\rightarrow}(\mathsf{INT}_n\,|_{\mathscr{S}\times\mathscr{S}})$, among all $S \subset 2^{[n]}$ of VC dimension d, follows directly from the fact that if $\mathscr{S}$ is a collection of all subsets of $[n]$ of size at most d then $D(\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}}) = D(\mathsf{INT}_n\,|_{\mathscr{S}\times\mathscr{S}}) = \Omega\left(d\log\left(\frac{n}{d}\right)\right)$, see [102, 125].

9.1.2 Preliminaries

We denote the set $\{1,\ldots,n\}$ by $[n]$. For a binary number $\mathbf{x}$, *decimal*$(\mathbf{x})$ denotes the decimal value of $\mathbf{x}$. For two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\{0,1\}^n$, $\mathbf{x}\cap\mathbf{y} = \{i \in [n] : \mathbf{x}_i = \mathbf{y}_i = 1\}$, and $\mathbf{x} \subseteq \mathbf{y}$ when $\mathbf{x}_i \leq \mathbf{y}_i$ for each $i \in [n]$. For a finite set X, 2^X denotes the power set of X. For $x, y \in \mathbb{R}$ with $x < y$, $[x,y]$ denotes the closed interval is the set of all real numbers that lie between x and y.

9.2 One way communication complexity (Theorems 9.4 and 9.6 (1))

In this section we will prove the following result.

Theorem 9.7. *For all* $n \geq d$, *there exists* $X \subset \mathbb{Z}$ *with* $|X| = n$ *and* $\mathscr{R} \subseteq 2^X$ *with VC-dim*$(\mathscr{R}) = 2d$, *such that*

$$\mathscr{R} \subseteq \left\{ X \cap \left(\bigcup_{1\leq j\leq d} I_j \right) \mid \{I_1,\ldots,I_d\} \in \mathscr{I} \right\} \text{ and } R^{\rightarrow}(\textsc{Disj}_X\,|_{\mathscr{R}\times\mathscr{R}}) = \Omega\left(d\log\frac{n}{d}\right).$$

Note that the set $\mathscr{I}$ *is defined in Problem 9.2.*

Fig. 9.1 Let us consider $d = 3, n = 18$ and $m = 4$. J_1, J_2 and J_3 are the intervals of length 4 starting from p_1, p_2 and p_3, respectively. The ground set X is the set of all 18 points present in three intervals.

Remark 7. The above result takes care of the proofs of Theorem 9.4 and Theorem 9.6 (1).

The *hard* instance, for the proof of the above theorem, is inspired by the *interval* set systems in combinatorial geometry and is constructed in Section 9.2.1. In Section 9.2.2, we prove Theorem 9.7 by using a reduction from AUGMENTED INDEXING, which we denote by AUGINDEX$_\ell$. Formally the problem AUGINDEX$_\ell$ is defined as follows: Alice gets a string $\mathbf{x} \in \{0,1\}^\ell$ and Bob gets an index $j \in [\ell]$ and $x_{j'}$ for all $j' < j$. Bob wants to report x_j as the output.

Proposition 9.8. (Miltersen et al. [114]) $R^{\rightarrow}(\text{AUGINDEX}_\ell) = \Omega(\ell)$.

9.2.1 Construction of a hard instance

We construct a set $X \subset \mathbb{Z}$ with $|X| = n$ and $\mathscr{R} \subseteq 2^X$ with VC-dim$(\mathscr{R}) = 2d$. Informally, X is the union of the set of points present in the union of d pairwise disjoint intervals, in $\mathbb{Z}$, each containing $\frac{n}{d}$ points. Each set in $\mathscr{R}$ is the union of the set of points in the subintervals anchored either at the left or the right end point of each of the above d intervals. Formally, the description of X and $\mathscr{R}$ are given below along with some of its properties that are desired to show Theorem 9.7.

The ground set X: Let $m = \frac{n}{d} - 2$. Without loss of generality we can assume that $m = 2^k$, where $k \in \mathbb{N}$. Let $J_0 = \{0, \ldots, m+1\}$ be the set of $m+2$ consecutive integers that starts from the origin and ends at $m+1$. Similarly, let J_p be the set of $m+2$ consecutive integers that starts at $p \in \mathbb{Z}$ and ends at $p+m+1$. Let $p_1, \ldots, p_d$ be d points in $\mathbb{Z}$ such that the sets $J_{p_1}, \ldots, J_{p_d}$ are pairwise disjoint. Let the *ground* set X be

$$X = \bigcup_{i=1}^{d} J_{p_i}.$$

Note that $X \subset \mathbb{Z}$ and $|X| = (m+2)d = n$. See Figure 9.1 for an illustration.

The subsets of X in $\mathscr{R}$: $\mathscr{R} \subseteq 2^X$ contains two types of sets $\mathscr{R}_0$ and $\mathscr{R}_{m+1}$, where

- Take any d intervals $R_1, \ldots, R_d$ of integer lengths such that, for all $i \in [d]$, length of R_i is at most $m+1$, $R_i \subseteq [p_i, p_i+m+1]$, and R_i starts at p_i. Note that R_i does not intersect with any $X \setminus J_{p_i}$. The set $A = \bigcup_{i=1}^{d} (R_i \cap X)$ is an element in $\mathscr{R}_0$. We say that A is *generated* by $R_1, \ldots, R_d$.
- Take any d intervals $R'_1, \ldots, R'_d$ of integer lengths such that, for all $i \in [d]$, length of R'_i is at most $m+1$, $R'_i \subseteq [p_i, p_i+m+1]$ and R'_i ends at p_i+m+1. Note that R'_i does not intersect with any $X \setminus J_{p_i}$. The set $B = \bigcup_{i=1}^{d} (R'_i \cap X)$ is an element in $\mathscr{R}_{m+1}$. We say that B is *generated* by $R'_1, \ldots, R'_d$.

See Figure 9.2 for an illustration.

(a) A and B *intersect*.

(b) A and B does not *intersect*.

Fig. 9.2 Consider n, d, m and X as in Figure 9.3. A is the set of points in X that are present in the three blue intervals. Similarly, B is the set of points in X that are present in the three red intervals.

The following claim bounds the VC dimension of $\mathscr{R}$, constructed as above.

Claim 9.9. *For $X \subset \mathbb{Z}$ with $|X| = n$ and $\mathscr{R} \subset 2^X$ as described above, VC-dim$(\mathscr{R}) = 2d$,*

Proof. The proof follows from the fact that any subset of of X containing $2d+1$ points will contain at least three points from some J_{p_i}, where $i \in [d]$. These points in J_{p_i} can not be shattered by the sets in $\mathscr{R}$. Also, observe that there exists $2d$ points, with two from each J_{p_j}, that can be shattered by the sets in $\mathscr{R}$. □

Now, we give a claim about X and $\mathscr{R}$ constructed above that will be required for our proof of Theorem 9.7.

Claim 9.10. *Let $A \in \mathscr{R}_0$ and $B \in \mathscr{R}_{m+1}$ be such that A is generated by $R_1, \ldots, R_d$ and B is generated by $R'_1, \ldots, R'_d$. Then A and B intersects if and only if there exists an $i \in [d]$ such that R_i intersects R'_i at a point in J_{p_i}.*

The proof of Claim 9.10 follows directly from our construction of $X \subset \mathbb{Z}$ and $\mathscr{R} \subseteq 2^X$, as $J_{p_1}, \ldots, J_{p_d}$ are pairwise disjoint.

9.2.2 Reduction from $\text{AUGINDEX}_{d\log m}$ to $\text{DISJ}_X\,|_{\mathscr{R}\times\mathscr{R}}$

Before presenting the reduction we recall the definitions of $\text{AUGINDEX}_{d\log m}$ and $\text{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$. In $\text{AUGINDEX}_{d\log m}$, Alice gets $\mathbf{x}\in\{0,1\}^{d\log m}$ and Bob gets an index j and $x_{j'}$ for each $j'<j$. The objective of Bob is to report x_j as the output. In $\text{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$, Alice gets $A\in\mathscr{R}_0$ and Bob gets $B\in\mathscr{R}_{m+1}$. The objective of Bob is to determine whether $A\cap B=\emptyset$. Note that $X,\mathscr{R},\mathscr{R}_0$ and $\mathscr{R}_{m+1}$ are as discussed in the Section 9.2.1.

Let $\mathscr{P}$ be a one-way protocol that solves $\text{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$ with $o\left(d\log\frac{n}{d}\right)=o(d\log m)$ bits of communication. Now, we consider the following protocol $\mathscr{P}'$ for $\text{AUGINDEX}_{d\log m}$ that has the same one way communication cost as that of $\text{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$. Then we will be done with the proof of Theorem 9.7.

Protocol $\mathscr{P}'$ for $\text{AUGINDEX}_{d\log m}$ problem

Step-1 Let $\mathbf{x}\in\{0,1\}^{d\log m}$ be the input of Alice. Bob gets an index $j\in[d\log m]$ and bits $x_{j'}$ for each $j'<j$.

Step-2 Alice will form d strings $\mathbf{a}_1,\ldots,\mathbf{a}_d\in\{0,1\}^{\log m}$ by partitioning the string $\mathbf{x}$ into d parts such that, $\forall i\in[d]$, we have

$$\mathbf{a}_i=x_{(i-1)\log m+1}\ldots x_{i\log m}.$$

Bob first forms a string $\mathbf{y}\in\{0,1\}^{d\log m}$, where $y_{j'}=x_{j'}$ for each $j'<j$, $y_j=1$, and $y_{j'}=0$ for each $j'>j$. Then Bob finds $\mathbf{b}_1,\ldots,\mathbf{b}_d\in\{0,1\}^{\log m}$ by partitioning the string $\mathbf{y}$ into d parts such that, $\forall i\in[d]$, we have

$$\mathbf{b}_i=y_{(i-1)\log m+1}\cdots y_{i\log m}.$$

Step-3 For each $i\in[d]$, let R_i and R'_i be the intervals that starts at p_i and ends at p_i+m+1, respectively, where

$$R_i=[p_i,m+p_i-\mathit{decimal}(\mathbf{a}_i)]$$

and

$$R'_i=[p_i+m+1-\mathit{decimal}(\mathbf{b}_i),p_i+m+1].$$

Alice finds the set $A \in \mathscr{R}_0$ generated by $R_1, \ldots, R_d$ and Bob finds the set $B \in \mathscr{R}_{m+1}$ generated by $R'_1, \ldots, R'_d$, i.e.,

$$A = \bigcup_{i \in [d]} (R_i \cap X) \text{ and } B = \bigcup_{i \in [d]} (R'_i \cap X).$$

Step-4 Alice and Bob solves $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$ on inputs A and B, and report $x_j = 0$ if and only if $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}(A,B) = 0$. Note that x_j is the output of $\text{AUGINDEX}_{d\log m}$ problem.

The following observation follows from the description of the protocol $\mathscr{P}'$ and from the construction of $X \subset \mathbb{Z}$ and $\mathscr{R} \subseteq 2^X$.

Observation 9.11. *Let $i^* \in [d]$ such that $j \in \{(i^*-1)\log m + 1, i^* \log m\}$. Then*

(i) $R_i \cap R'_i = \emptyset$ *for all* $i \neq i^*$.

(ii) $R_{i^*} \cap R'_{i^*} = \emptyset$ *if and only if* $decimal(\mathbf{b}_{i^*}) \leq decimal(\mathbf{a}_{i^*})$.

(iii) $decimal(\mathbf{b}_{i^*}) > decimal(\mathbf{a}_{i^*})$ *if and only if* $x_j = 0$.

We will use the above observation to show the correctness of the protocol $\mathscr{P}'$.

First consider the case $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}(A,B) = 0$. Then, by Claim 9.10, there exists an $i \in [d]$ such that R_i and R'_i intersects at a point in J_{p_i}. From Observation 9.11 (i), we can say $R_{i*} \cap R'_{i^*} \neq \emptyset$. Combining $R_{i*} \cap R'_{i^*} \neq \emptyset$ with Observations 9.11 (ii) and (iii), we have $x_j = 0$. Hence, $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}(A,B) = 0$ implies $x_j = 0$. The converse part, i.e., $x_j = 0$ implies $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}(A,B) = 0$, can be shown in the similar fashion.

The one-way communication complexity of protocol $\mathscr{P}'$ for $\text{AUGINDEX}_{d\log m}$ is the same as that of $\mathscr{P}$ for $\mathsf{DISJ}_X|_{\mathscr{R}\times\mathscr{R}}$, that is, $o(d\log m)$. However, this is impossible as the one-way communication complexity of AUGMENTED INDEXING, over $d\log m$ bits, is $\Omega(d\log m) = \Omega\left(d\log\frac{n}{d}\right)$ bits. This completes the proof of Theorem 9.7.

9.3 Two way communication complexity (Theorems 9.3, 9.5, 9.6(2) and 9.6(3))

In this section, we prove the following theorems.

Theorem 9.12. *For all $n \geq d$, there exists a $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$ with VC-dim$(\mathscr{T}) = 2d$, such that*

$$\mathscr{T} \subseteq \left\{ G \cap \left(\bigcup_{1 \leq j \leq d} \ell_j \right) \mid \{\ell_1, \ldots, \ell_d\} \in \mathscr{L} \right\} \text{ and } R(\mathsf{DISJ}_G |_{\mathscr{T} \times \mathscr{T}}) = \Omega\left(d \frac{\log(n/d)}{\log\log(n/d)} \right).$$

The set $\mathscr{L}$ is as defined in Problem 1.

Theorem 9.13. *For all $n \geq d$, there exists a $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$ with VC-dim$(\mathscr{T}) = 2d$, such that*

$$\mathscr{T} \subseteq \left\{ G \cap \left(\bigcup_{1 \leq j \leq d} \ell_j \right) \mid \{\ell_1, \ldots, \ell_d\} \in \mathscr{L} \right\} \text{ and } R(\mathsf{INT}_G |_{\mathscr{T} \times \mathscr{T}}) = \Omega\left(d \log \frac{n}{d} \right).$$

The set $\mathscr{L}$ is as defined in problem 1.

Remark 8. Theorem 9.12 takes care of Theorem 9.3 and 9.6(2). Theorem 9.13 takes care of Theorem 9.5 and 9.6(3).

Note that the same set system will be used for the proofs of the above theorems. The *hard* instance, for the proof of the above theorems, is inspired by *point line incidence* set systems in computational geometry and is constructed in Section 9.3.1. We prove Theorems 9.12 and 9.13 in Sections 9.3.2 and 9.3.3, respectively, using reductions.

9.3.1 The hard instance for the proofs of Theorems 9.12 and 9.13

In this subsection, we give the description of $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$, with VC-Dim$(\mathscr{T}) = 2d$. The same G and $\mathscr{T}$ will be our *hard* instance for the proofs of Theorems 9.12 and 9.13. In this subsection, without loss of generality, we can assume that d divides n and n/d is a perfect square.

Informally, G is the set of points present in the union of d many pairwise disjoint square grids each containing $\frac{n}{d}$ points and the grids are taken in such a way that any straight line of non-negative slope can intersects with at most one grid. Also, each set in $\mathscr{T}$ is the union of the set of points present in d many lines of non-negative slope such that one line intersects with exactly one grid. Moreover, all of the d lines have slopes either zero or positive. Formally, the description of G and $\mathscr{T}$ are given below along with some of its properties that are desired to show Theorems 9.12 and 9.13.

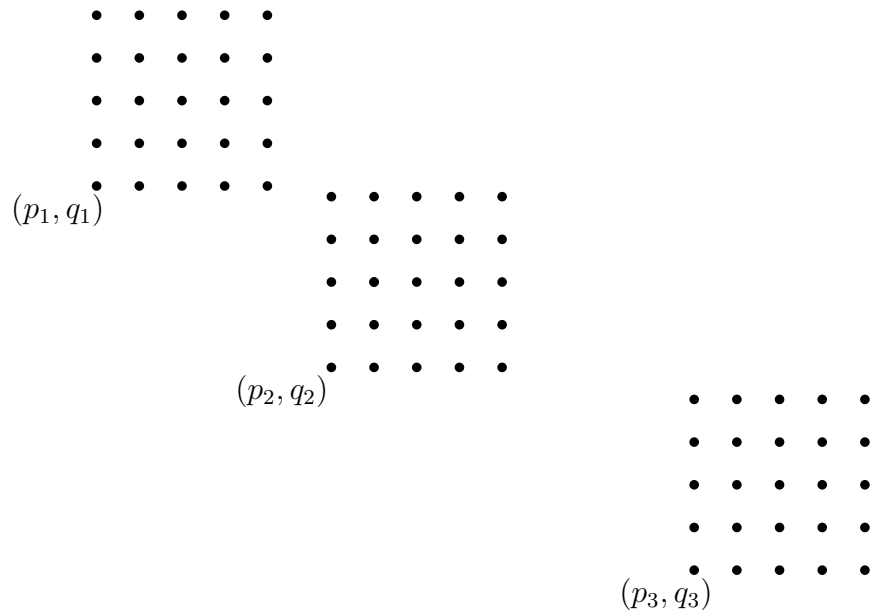


Fig. 9.3 Let us take $n = 75, d = 3$ and $m = 5$. The 5×5 grids centered at $(p_1, q_1), (p_2, q_2)$ and (p_3, q_3) are $G_{(p_1,q_1)}$, $G_{(p_2,q_2)}$ and $G_{(p_3,q_3)}$; respectively. The ground set G is the set of all 75 points present in three grids.

The ground set G: Let $m = \sqrt{\frac{n}{d}}$, and

$$G_{(0,0)} = \left\{(x,y) \in \mathbb{Z}^2 : 0 \leq x, y \leq m-1\right\}$$

be the grid of size $m \times m$ anchored at the origin $(0,0)$. For any $p, q \in \mathbb{Z}$, the $m \times m$ grid anchored at (p,q) will be denoted by $G_{(p,q)}$, i.e.,

$$G_{(p,q)} = \left\{(i+p, j+q) : (i,j) \in G_{(0,0)}\right\}.$$

For $d \in \mathbb{N}$, consider $G_{(p_1,q_1)}, \ldots, G_{(p_d,q_d)}$ satisfying the following property:

PROPERTY *For any $i, j \in [d]$, with $i \neq j$, let L_1 and L_2 be lines of non-negative slopes that pass through at least two points of $G_{(p_i,q_i)}$ and $G_{(p_j,q_j)}$, respectively. Then L_1 and L_2 does not intersect at any point* inside $\bigcup_{\ell=1}^{d} G_{(p_\ell,q_\ell)}$.

Observe that there exists $G_{(p_1,q_1)}, \ldots, G_{(p_d,q_d)}$ satisfying PROPERTY. See Figure 9.3 for an illustration. We will take the ground set G as

$$G = \bigcup_{\ell=1}^{d} G_{(p_\ell,q_\ell)}.$$

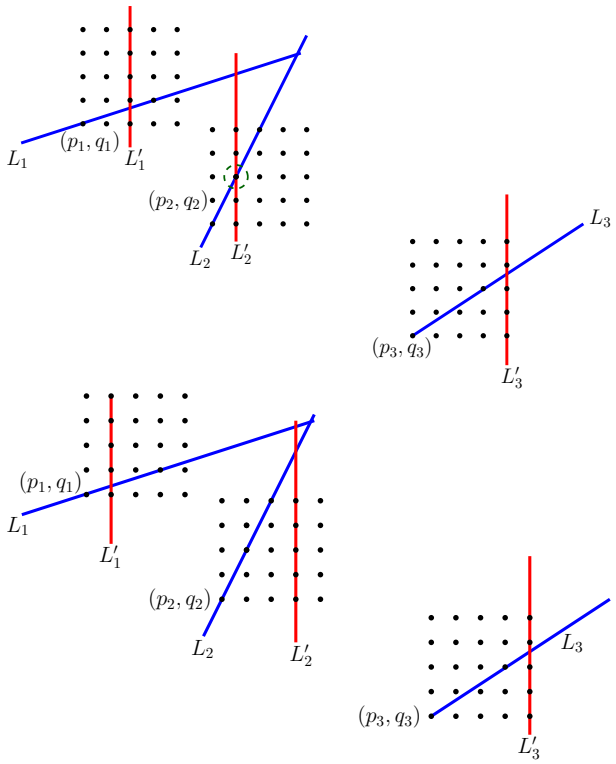


Fig. 9.4 Consider n,d,m and G as in Figure 9.3. A is the set of points in G that are present in three blue lines, that is, $L_1 \cup L_2 \cup L_3$. Similarly, B is the set of points in G that are present in three red line $L'_1 \cup L'_2 \cup L'_3$. First figure shows the instance where A and B *intersect* at a grid point, and the second figure shows an instance where A and B does not *intersect* at a grid point.

Without loss of generality, we can assume that $(p_1,q_1) = (0,0)$. Note that $G \subset \mathbb{Z}^2$ and $|G| = dm^2 = n$.

The subsets of G in $\mathscr{T}$: $\mathscr{T}$ contains two types of subsets $\mathscr{T}_1$ and $\mathscr{T}_2$ of G, and they are generated by the following ways:

- Take any d lines $L_1,\ldots,L_d$ of non negative slope such that, $\forall i \in [d]$, L_i passes through $(p_i,q_i) \in G_{(p_i,q_i)}$ and (at least) another point in $G_{(p_i,q_i)}$. Note that L_i does not contain any point from $G \setminus G_{(p_i,q_i)}$. The set $A = \bigcup_{i=1}^{d} \left(L_i \cap G_{(p_i,q_i)}\right)$ is in $\mathscr{T}_1$, and we say A is *generated* by the lines $L_1,\ldots,L_d$.
- Take any d vertical lines $L'_1,\ldots,L'_d$ such that, $\forall i \in [d]$, L'_i contains at least one point from $G_{(p_i,q_i)}$. Note that L'_i does not contain any point from $G \setminus G_{(p_i,q_i)}$. The set $B = \bigcup_{i=1}^{d} (L'_i \cap G_{(p_i,q_i)})$ is in $\mathscr{T}_2$, and we say B is generated by the lines $L'_1,\ldots,L'_d$.

See Figure 9.4 for an illustration.

The following claim bounds the VC dimension of $\mathscr{T}$, which as described above.

Claim 9.14. *For $G \subset \mathbb{Z}^2$ and $\mathscr{T} \subseteq 2^G$ as described above, VC-dim$(\mathscr{T}) = 2d$.*

Proof. The proof follows from the fact that any subset of X containing $2d+1$ points will contain at least three points from some $G_{(p_j,q_j)}, j \in [d]$. These points in $G_{(p_j,q_j)}$ can not be shattered by the sets in $\mathscr{T}$. Also, observe that there exists $2d$ points two from each $G_{(p_j,q_j)}$ that can be shattered by the sets in $\mathscr{T}$. □

Now, we give two claims about G and $\mathscr{T}$, constructed above, that follow directly from our construction of $G \subset \mathbb{Z}^2$ and $\mathscr{T} \subseteq 2^G$.

Claim 9.15. *Let $A \in \mathscr{T}_1$ and $B \in \mathscr{T}_2$ such that A is generated by lines $L_1, \ldots, L_d$ and A is generated by lines $L'_1, \ldots, L'_d$. Then A and B intersect if and only if there exists $i \in [d]$ such that L_i and L'_i intersect at a point in $G_{(p_i,q_i)}$.*

Claim 9.16. *Let $A \in \mathscr{T}_1$ and $B \in \mathscr{T}_2$ such that A is generated by lines $L_1, \ldots, L_d$ and B is generated by lines $L'_1, \ldots, L'_d$. Also let $|A \cap B| = d$. Then for each $i \in [d]$, L_i and L'_i intersect at a point in $G_{(p_i,q_i)}$. Moreover, A (B) can be determined if we know B (A) and $A \cap B$.*

The above claims will be used in the proofs of Theorems 9.12 and 9.13.

9.3.2 Proof of Theorem 9.12

Let us consider a problem in communication complexity denoted by OR-DISJ$^t_{\{0,1\}^\ell}$ that will be used in our proof. In OR-DISJ$^t_{\{0,1\}^\ell}$, Alice gets t strings $\mathbf{x}_1, \ldots, \mathbf{x}_t \in \{0,1\}^\ell$ and Bob also gets t strings $\mathbf{y}_1, \ldots, \mathbf{y}_t \in \{0,1\}^\ell$. The objective is to compute

$$\text{OR-DISJ}^t_{\{0,1\}^\ell}((\mathbf{x}_1, \ldots, \mathbf{x}_t), (\mathbf{y}_1, \ldots, \mathbf{y}_t)) = \bigvee_{i=1}^{t} \text{DISJ}_{\{0,1\}^\ell}(\mathbf{x}_i, \mathbf{y}_i).$$

Note that $\mathsf{DISJ}_{\{0,1\}^\ell}(\mathbf{x}_i, \mathbf{y}_i)$ is a binary variable that takes value 1 if and only if $\mathbf{x}_i \cap \mathbf{y}_i = \emptyset$.

Proposition 9.17 (Jayram et al. [92]). $R\left(\text{OR-DISJ}^t_{\{0,1\}^\ell}\right) = \Omega(\ell t)$.

Note that Proposition 9.17 directly implies the following result.

Proposition 9.18. $R\left(\text{OR-DISJ}^t_{\{0,1\}^\ell}|_{S_\ell \times S_\ell}\right) = \Omega(\ell t)$, where $S_\ell = \{0,1\}^\ell \setminus \{0^\ell\}$.

Let $k \in \mathbb{N}$ be the largest integer such that first k consecutive primes $\pi_1, \ldots, \pi_k$ satisfy the following inequalty:

$$\Pi_{i=1}^{k} \pi_i \leq \sqrt{\frac{n}{d}}. \tag{9.4}$$

Using the fact that $\Pi_{i=1}^{k} \pi_i = e^{(1+o(1))k\log k}$, we get $k = \Theta\left(\frac{\log(n/d)}{\log\log(n/d)}\right)$.

We prove the theorem by a reduction from OR-DISJ$^{d}_{\{0,1\}^k}|_{S_k \times S_k}$ to DISJ$_G|_{\mathscr{T}\times\mathscr{T}}$, where

$$S_k := \{0,1\}^k \setminus \{0^k\}.$$

Note that $G \subset \mathbb{Z}^2$ with $|G| = n$, and $\mathscr{T} \subseteq 2^G$, with VC-dim$(\mathscr{T}) = 2d$, are the same as that we constructed in Section 9.3.1. To reach a contradiction, assume that there exists a two way protocol $\mathscr{P}$ that solves DISJ$_G$ $|_{\mathscr{T}\times\mathscr{T}}$ with communication cost of

$$o\left(d\frac{\log m}{\log\log m}\right) = o\left(d\frac{\log(n/d)}{\log\log(n/d)}\right).$$

Now, we give protocol $\mathscr{P}'$ that solves OR-DISJ$^{d}_{\{0,1\}^k}|_{S_k\times S_k}$, as described below.

Protocol $\mathscr{P}'$ for OR-DISJ$^{d}_{\{0,1\}^k}|_{S_k\times S_k}$

Step-1 Let $A = (\mathbf{x}_1, \ldots, \mathbf{x}_d) \in [S_k]^d$ [5] and $B = (\mathbf{y}_1, \ldots, \mathbf{y}_d) \in [S_k]^d$ be the inputs of Alice and Bob for OR-DISJ$^{d}_{\{0,1\}^k}|_{S_k\times S_k}$. Recall that $S_k = \{0,1\}^k \setminus \{0^k\}$. Bob finds $\bar{B} = (\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_d) \in \left[\{0,1\}^k\right]^d$, where $\bar{\mathbf{y}}_i$ is obtained by complementing each bit of $\mathbf{y}_i$.

Step-2 Both Alice and Bob privately determine the first k prime numbers $\pi_1, \ldots, \pi_k$ without any communication.

Step-3 Let $\Phi : \{0,1\}^k \to \{0,1\}^{\lceil\log(\sqrt{\frac{n}{d}})\rceil}$ be the function such that $\phi(\mathbf{x})$ is the bit representation of the number $\prod_{i=1}^{k} \pi_i^{x_i}$, where $\mathbf{x} = (x_1, \ldots, x_k) \in \{0,1\}^k$. Alice finds $A' = (\mathbf{a}_1, \ldots, \mathbf{a}_d) \in \left[\{0,1\}^{\lceil\log(\sqrt{\frac{n}{d}})\rceil}\right]^d$ and Bob finds $B' = (\mathbf{b_1}, \ldots, \mathbf{b_1} \in \left[\{0,1\}^{\lceil\log(\sqrt{\frac{n}{d}})\rceil}\right]^d$ privately without any communication, where $\mathbf{a_i} = \phi(\mathbf{x_i})$ and $\mathbf{b_i} = \phi(\bar{\mathbf{y}}_i)$ for each $i \in [d]$.

Step-4 For each $i \in [d]$, let L_i and L_i' be the lines having equation

$$L_i : y - q_i = \frac{decimal(\mathbf{a}_i) - 1}{decimal(\mathbf{a}_i)}(x - p_i)$$

[5] For a set W, $[W]^d = W \times \ldots \times W$ (d times).

and

$$L'_i : x - p_i = decimal(\mathbf{b}_i).$$

Here p_i's and q_i's are selected to satisfy PROPERTY. Alice finds $A'' \in \mathscr{T}$ that is generated by the lines $L_1, \ldots, L_d$, and Bob finds $B'' \in \mathscr{T}$ which is generated by the lines $L'_1, \ldots, L'_d$, i.e.,

$$A'' = \bigcup_{i \in [d]} (L_i \cap G_{(p_i,q_i)}) \text{ and } B'' = \bigcup_{i \in [d]} (L'_i \cap G_{(p_i,q_i)}).$$

Step-5 Then Alice and Bob solve $\mathsf{DISJ}_G \mid_{\mathscr{T} \times \mathscr{T}} (A'', B'')$, and report $\bigvee_{i=1}^{d} \textsc{Disj}_{\{0,1\}^k}(\mathbf{x}_i, \mathbf{y}_i) = 1$ if and only if $\mathsf{DISJ}_G \mid_{\mathscr{T} \times \mathscr{T}} (A'', B'') = 0$.

Now we argue for the correctness of the protocol $\mathscr{P}'$. Let $\mathsf{DISJ}_G \mid_{\mathscr{T} \times \mathscr{T}} (A'', B'') = 0$, that is, $A'' \cap B'' \neq \emptyset$. By Claim 9.15 and from the description of $\mathscr{P}'$, there exists $i \in [d]$ such that the lines $L_i : y - q_i = \frac{decimal(\mathbf{a}_i)-1}{decimal(\mathbf{a}_i)}(x - p_i)$ and $L'_i : x - p_i = decimal(\mathbf{b_i})$ intersect at a point in $G_{(p_i,q_i)}$, that is, the lines $y = \frac{decimal(\mathbf{a}_i)-1}{decimal(\mathbf{a}_i)}x$ and $x = decimal(\mathbf{b}_i)$ intersect at a point in $G_{(0,0)}$. Now, we can say that, there exists $i \in [d]$ such that $decimal(\mathbf{a}_i)$ divides $decimal(\mathbf{b_i})$, equivalently, $\phi(\mathbf{x}_i)$ divides $\phi(\bar{\mathbf{y}}_i)$. This implies $\mathbf{x}_i$ is a subset of $\bar{\mathbf{y}}_i$ (or $\mathbf{x}_i \cap \mathbf{y_i} = \emptyset$) for some $i \in [d]$. Hence, $\bigvee_{i=1}^{d} \textsc{Disj}_{\{0,1\}^k}(\mathbf{x}_i, \mathbf{y}_i) = 1$. The converse part, that is, $\bigvee_{i=1}^{d} \textsc{Disj}_{\{0,1\}^k}(\mathbf{x}_i, \mathbf{y}_i) = 1$ implies $\mathsf{DISJ}_G \mid_{\mathscr{T} \times \mathscr{T}} (A'', B'') = 0$ can be shown in the similar fashion.

Observe that the communication cost of protocol $\mathscr{P}'$ for $\textsc{Or-Disj}^d_{\{0,1\}^k} \mid_{S_k \times S_k}$ is same as that of protocol $\mathscr{P}$ for $\mathsf{DISJ}_G \mid_{\mathscr{T} \times \mathscr{T}}$, is

$$o\left(d\frac{\log m}{\log\log m}\right) = o\left(d\frac{\log(n/d)}{\log\log(n/d)}\right) = o(dk).$$

The above two equalities follows from the facts that $m = \sqrt{\frac{n}{d}}$ and $k = \Theta\left(\frac{\log(n/d)}{\log\log(n/d)}\right)$. This contradicts Proposition 9.18 which says that

$$R\left(\textsc{Or-Disj}^d_{\{0,1\}^k} \mid_{S_k \times S_k}\right) = \Omega(dk).$$

9.3.3 Proof of Theorem 9.13

With out loss of generality, we also assume that d divides n and, more over, n/d is a perfect square.

First, consider the problem $\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}$, where the objective of Alice and Bob is to learn each other's set. Note that $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$ with VC-Dim$(\mathscr{T}) = 2d$ are same as that constructed in Section 9.3.1. In $\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}$, Alice and Bob get two sets A and B, respectively, from $\mathscr{T}$ with a promise $|A \cap B| = d$. The objective of Alice (Bob) is to learn B (A). Observe that $R(\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}) = \Omega(d \log n)$ as there are $\Omega(m^d) = \Omega\left(\left(\sqrt{\frac{n}{d}}\right)^d\right)$ many candidate sets for the inputs of Alice and Bob. We prove the theorem by a reduction from $\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}$ to $\mathsf{INT}_G|_{\mathscr{T}\times\mathscr{T}}$.

Let by contradiction consider a protocol $\mathscr{P}$ that solves $\mathsf{INT}_G|_{\mathscr{T}\times\mathscr{T}}$ by using $o(d \log n)$ bits of communication. To solve $\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}$, Alice and Bob first run a protocol $\mathscr{P}$ and finds $A \cap B$. Now by Claim 9.15, it is possible for Alice (Bob) to determine B (A) by combining A (B) along with $A \cap B$, with out any communication with Bob (Alice). Now, we have a protocol $\mathscr{P}'$ that solves $\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}$ with $o(d \log n)$ bits of communication. However, this is impossible as $R(\textsc{Learn}_G|_{\mathscr{T}\times\mathscr{T}}) = \Omega(d \log n)$. Hence, we are done with the proof of Theorem 9.13.

9.4 VC dimension, and Problems 9.1 and 9.2

VC dimension, and collection of d lines. Let $G \subset \mathbb{Z}^2$ be a set of n points in $\mathbb{Z}^2$. Observe, that the communication functions $\mathsf{DISJ}_G|_{\mathscr{L}\times\mathscr{L}}$ (defined in Problem 9.1) and $\mathsf{DISJ}_G|_{\mathscr{G}\times\mathscr{G}}$, where

$$\mathscr{G} = \left\{ G \cap \left(\bigcup_{1 \le j \le d} \ell_j \right) \mid \{\ell_1, \ldots, \ell_d\} \in \mathscr{L} \right\},$$

are equivalent problems. Note that the set $\mathscr{L}$ is defined in Problem 9.1. Using standard geometric arguments, see [112, Chap. 10] and [79, Chap. 5], we can show that VC-dim$(\mathscr{G}) = 2d$.

VC dimension, and collection of d intervals. Let $X \subset \mathbb{Z}$ be a set of n points in $\mathbb{Z}$. Observe, that the communication functions $\mathsf{DISJ}_X|_{\mathscr{I}\times\mathscr{I}}$ (defined in Problem 9.2) and $\mathsf{DISJ}_X|_{\mathscr{F}\times\mathscr{F}}$, where

$$\mathscr{F} = \left\{ X \cap \left(\bigcup_{1 \le j \le d} I_j \right) \mid \{I_1, \ldots, I_d\} \in \mathscr{I} \right\},$$

are equivalent problems. Note that the set $\mathscr{I}$ is defined in Problem 9.2. Using standard geometric arguments, as in the above case, we can show that VC-dim$(\mathscr{F}) = 2d$.

9.5 Summary

In this chapter, we studied $\mathsf{DISJ}_n|_{\mathscr{S}\times\mathscr{S}}$ and $\mathsf{INT}_n|_{\mathscr{S}\times\mathscr{S}}$ when $\mathscr{S}$ is a subset of $2^{[n]}$ and VC-dim$(\mathscr{S}) \leq d$. One of the main contributions of our work is the result (Theorem 9.6) showing that unlike in the case of d-SPARSEDISJ$_n$ and d-SPARSEINT$_n$ functions, there is no separation between randomized and deterministic communication complexity of $\mathsf{DISJ}_n|_{\mathscr{S}\times\mathscr{S}}$ and $\mathsf{INT}_n|_{\mathscr{S}\times\mathscr{S}}$ functions when VC-dim$(S) \leq d$. Note that we have settled both the one-way and two-way (randomized) communication complexities of $\mathsf{INT}_n|_{\mathscr{S}\times\mathscr{S}}$ when VC-dim$(\mathscr{S}) \leq d$ (Theorem 9.6 (1) and (3)). In the context of $\mathsf{DISJ}_n|_{\mathscr{S}\times\mathscr{S}}$, we have settled the one-way (randomized) communication complexity. The two-way communication complexity for $\mathsf{DISJ}_n|_{\mathscr{S}\times\mathscr{S}}$ is tight up to factor $\log\log\frac{n}{d}$ (See Theorem 9.6 (2)).

Chapter 10

Query Complexity of Global Minimum Cut

10.1 Introduction

The global minimum cut (denoted MINCUT) of a connected, unweighted, undirected and simple graph $G = (V,E)$, $|V| = n$ and $|E| = m$, is a partition of the vertex set V into two sets S and $V \setminus S$ such that the number of edges between S and $V \setminus S$ is minimized. Let CUT(G) denote this edge set corresponding to a minimum cut in G, and t denote $|\text{CUT}(G)|$. The problem is so fundamental that researchers keep coming back to it again and again across different models [96, 95, 113, 98, 117, 5, 129, 66, 64, 65, 94]. The algorithmic landscape for the minimum cut problem has been heavily influenced by Karger and Stein's work [95, 96] and algorithmic solutions for minimum cut across different models [113, 98, 117, 5, 129, 66, 64, 65, 94] [1] have revisited their approach [95, 96]. Fundamental graph parameter estimation problems, like estimation of the number of edges [58, 68], triangles [56], cliques [57], stars [70], etc. have been solved in the local and bounded query models [69, 68, 97]. Estimation of the size of MINCUT is also in the league of such fundamental problems to be solved in the model of local queries.

In property testing [67], a graph can be accessed at different granularities — the query oracle can answer properties about graph that are local or global in nature. Local queries involve the relation of a vertex with its immediate neighborhood, whereas, global queries involve the relation between sets of vertices. Recently using a global query, named CUT QUERY [129], the problem of estimating and finding MINCUT was solved, but the problem

[1] The list is to name a few and it is not exhaustive.

of estimating or finding MINCUT using local queries has not been solved. The fundamental contribution of our work is to resolve the query complexity of MINCUT using local queries. We resolve both the estimation and finding variants of the problem. To start with, we formally define the query oracle models we would be needing for discussions that follow.

The query oracle models. We start with the most studied local queries and the random edge query for a graph $G = (V, E)$ where the vertex set V is known but the edge set E is unknown.

- Local Query
  - DEGREE query: given $u \in V$, the oracle reports the degree of u in V;
  - NEIGHBOR query: given $u \in V$ and a positive integer i, the oracle reports the i-th neighbor of u, if it exists; otherwise, the oracle reports $\perp$;
  - ADJACENCY query: given $u, v \in V$, the oracle reports whether $\{u, v\} \in E$.
- RANDOM EDGE query: The query outputs an uniformly random edge of G.

Apart from the local queries mentioned, in the last few years, researchers have also used the RANDOM EDGE query [6, 14]. Notice that the randomness will be over the probability space of all edges, and hence, a random edge query is not a local query. This fact is also evident from the work of Eden and Rosenbaum [55]. We use RANDOM EDGE query in conjunction with local queries only for lower bound purposes. The other query oracle relevant for our discussion will be a *global query* called the CUT QUERY proposed by Rubinstein et al. [129] that was motivated by submodular function minimization. The query takes as input a subset S of the vertex set V and returns the size of the cut between S and $V \setminus S$ in the graph G.

Prologue. Our motivation for this work is twofold — MINCUT is a fundamental graph estimation problem that needs to be solved in the local query oracle model and the lower bound of Eden and Rosenbaum [54] who extended the seminal work of Blais et al. [26] to develop a technique for proving query complexity lower bounds for graph properties via reductions from communication complexity. Using those techniques, for graphs that can be accessed by only local queries like DEGREE, NEIGHBOR, ADJACENCY and RANDOM EDGE, Eden and Rosenbaum [54] showed that MINCUT admits a lower bound of $\Omega(m/t)$ in general graphs, where m and t are the number of edges and the size of the minimum cut, respectively, in the graph. However, the query complexity of estimating MINCUT (in general graphs) has remained elusive as there is no matching upper bound. It is surprising that the query complexity of a fundamental graph problem like MINCUT has not been addressed before Eden and Rosenbaum [54].

In this work, we prove an upper bound of $\min\{m+n, \frac{m}{t}\}\text{poly}\left(\log n, \frac{1}{\varepsilon}\right)$ for estimating MINCUT using local queries only (and not RANDOM EDGE query). Observe that for $t \geq 1$, our upper bound almost matches with the lower bound given by Eden and Rosenbaum [54] if we ignore poly $\left(\log n, \frac{1}{\varepsilon}\right)$ term. Note that the case of $t = 0$ is the CONNECTIVITY problem, where the objective is to decide whether the graph is connected. We build on the lower bound result for *estimating* MINCUT by Eden et al. [54] to show that $\Omega(m)$ local queries are required if we want to determine the exact size of a MINCUT or find a MINCUT. This result implies that there is a separation between the problem of estimating the size of MINCUT and the problem of finding a MINCUT using local queries. On the other hand, Babai et al. [16] showed that CONNECTIVITY testing has a randomized communication complexity of $\Omega(n)$ [2]. This implies that any algorithm that solves CONNECTIVITY requires $\Omega(n/\log n)$ local queries. This is because Alice and Bob can deterministically simulate each local query by communicating at most $\log n$ bits. We have already discussed that $\Omega(m)$ local queries are needed to solve CONNECTIVITY. From the lower bounds of $\Omega(m)$ and $\Omega(n/\log n)$ for CONNECTIVITY along with the lower bound result of Eden and Rosenbaum [54] for estimating MINCUT, our upper bound result on estimating MINCUT (ignoring poly $\left(\log n, \frac{1}{\varepsilon}\right)$ term) is tight - this settles the query complexity of MINCUT using local queries.

Prior to our work, no local query based algorithm existed for MINCUT. But it was Rubinstein et al. [129] who studied MINCUT for the first time using CUT QUERY, a global query. They showed that there exists a randomized algorithm for finding a MINCUT in G using $\widetilde{O}(n)$[3] CUT QUERY. The deterministic lower bound of $\Omega(n\log n)$ by Hajnal et al. [78], for CONNECTIVITY in communication complexity, implies that $\Omega(n)$ CUT QUERY are required by any deterministic algorithm to estimate MINCUT. The randomized lower bound of $\Omega(n)$ by Babai et al. [16], for CONNECTIVITY in communication complexity, says that $\Omega(n/\log n)$ CUT QUERY are necessary for any randomized algorithm to estimate MINCUT [4]. So, if we ignore polylogarithmic factors, the upper bound result by Rubinstein et al. for finding a MINCUT along with the above discussed lower bound result for finding a MINCUT, imply that there is no separation between the problem of estimating size of MINCUT and the problem of finding a MINCUT using CUT QUERY. On a different note, Graur et al. [72] showed a deterministic lower bound of $3n/2$ on the number of CUT QUERY for estimating MINCUT.

[2]Here the edge set of the graph is partitioned among Alice and Bob. The objective of Alice and Bob is to determine whether the graph is connected by communicating.

[3]$\widetilde{O}(n)$ hides polylogarithmic terms in n.

[4]We would like to thank Troy Lee for pointing us to the papers of Hajnal et al. [78] and Babai et al. [16]

Problem statements and results. We focus on two problems in this work.

Minimum Cut Estimation **Input:** A parameter $\varepsilon \in (0,1)$, and access to an unknown graph G via local queries **Output:** A $(1 \pm \varepsilon)$-approximation to $\lvert \textsc{Cut}(G) \rvert$.

Minimum Cut Finding **Input:** Access to an unknown graph G via local queries **Output:** Find a set CUT(G) .

10.1.1 Results and Organization

Our results are the following.

Theorem 10.1. (Minimum cut estimation using local queries) *There exists an algorithm, with* DEGREE *and* NEIGHBOR *query access to an unknown graph $G = (V,E)$, that solves the minimum cut estimation problem with high probability. The expected number of queries used by the algorithm is* $\min\left\{m+n, \frac{m}{t}\right\} \operatorname{poly}\left(\log n, \frac{1}{\varepsilon}\right)$.

Building on the lower bound construction of Eden and Rosenbaum [54], we show that no nontrivial query algorithm exists for finding a minimum cut or even estimating the exact size of a minimum cut in graphs.

Theorem 10.2. (Lower bound for minimum cut finding, i.e., CUT(G)) *Let $m,n,t \in \mathbb{N}$ with $t \leq n-1$ and $2nt \leq m \leq \binom{n}{2}$* [5]*. Any algorithm that has access to* DEGREE, NEIGHBOR, ADJACENCY *and* RANDOM EDGE *queries to an unknown graph $G = (V,E)$ must make at least $\Omega(m)$ queries in order to find all the edges in a minimum cut of G with probability $2/3$.*

Theorem 10.3. (Lower bound for finding the exact size of the minimum cut, i.e., $\lvert \textsc{Cut}(G) \rvert$) *Let $m,n,t \in \mathbb{N}$ with $2 \leq t \leq n-2$ and $2nt \leq m \leq \binom{n}{2}$. Any algorithm that has access to* DEGREE, NEIGHBOR, ADJACENCY *and* RANDOM EDGE *queries to an unknown graph $G = (V,E)$ must make at least $\Omega(m)$ queries in order to decide whether $\lvert \textsc{Cut}(G) \rvert = t$ or $\lvert \textsc{Cut}(G) \rvert = t-2$ with probability $2/3$.*

Remark 9. Local queries show a clear separation in its power in finding MINCUT as opposed to the estimation problem. This is established by using the tight lower bound of minimum cut estimation (viz. $\Omega(m/t)$ lower bound of Eden and Rosenbaum and our Theorem 10.1) vis-a-vis minimum cut finding as mentioned in our Theorems 10.2 and 10.3 on lower bound

[5] As mentioned at the beginning of the introduction, $n.m$ and t denote the number of vertices, number of edges and the size of MINCUT in G, respectively.

for finding CUT(G) . As noted earlier, there is no such separation between estimating and finding MINCUT when CUT QUERY is used.

Organization. Section 10.3 discusses the query algorithm for estimating the MINCUT while Section 10.4 proves lower bounds on finding the MINCUT. Section 10.5 concludes with a few observations.

10.2 Preliminaries

Notations. In this chapter, we denote the set $\{1,\ldots,n\}$ by $[n]$. For ease of notation, we sometimes use $[n]$ to denote the set of vertices of a graph. We say $x \geq 0$ is an $(1\pm\varepsilon)$-approximation to $y \geq 0$ if $|x-y| \leq \varepsilon y$. $V(G)$ and $E(G)$ would denote the vertex and edge sets when we want to make the graph G explicit, else we use V and E. For a graph G, $\textsc{Cut}(G)$ denotes the set of edges in a minimum cut of G. Let A_1, A_2 be a partition of V, i.e., $V = A_1 \cup A_2$ with $A_1 \cap A_2 = \emptyset$. Then, $\mathscr{C}_G(A_1,A_2) = \{\{u,v\} \in E \,:\, u \in A_1 \text{ and } v \in A_2\}$. The statement *with high probability* means that the probability of success is at least $1-\frac{1}{n^c}$, where c is a positive constant. $\widetilde{\Theta}(\cdot)$ and $\widetilde{\mathscr{O}}(\cdot)$ hides a poly $\left(\log n, \frac{1}{\varepsilon}\right)$ term in the upper bound.

10.2.1 Probability Results

Lemma 10.4 (See [52]). *Let $X = \sum_{i\in[n]} X_i$ where X_i, $i \in [n]$, are independent random variables, $X_i \in [0,1]$ and $\mathbb{E}[X]$ is the expected value of X. Then*

(i) For $\varepsilon > 0$

$$\Pr[|X - \mathbb{E}[X]| > \varepsilon\mathbb{E}[X]] \leq \exp\left(-\frac{\varepsilon^2}{3}\mathbb{E}[X]\right).$$

(ii) Suppose $\mu_L \leq \mathbb{E}[X] \leq \mu_H$, then for $0 < \varepsilon < 1$

(a) $\Pr[X > (1+\varepsilon)\mu_H] \leq \exp\left(-\frac{\varepsilon^2}{3}\mu_H\right).$

(b) $\Pr[X < (1-\varepsilon)\mu_L] \leq \exp\left(-\frac{\varepsilon^2}{2}\mu_L\right).$

10.3 Estimation algorithm

In this Section, we will prove Theorem 10.1. In Section 10.3.1, we discuss about the intuitions and give the overview of our algorithm. We formalize the intuitions in Section 10.3.2.

10.3.1 Overview of our algorithm

We start by assuming that a lower bound $\hat{t}$ on $t = |\textsc{Cut}(G)|$ is known. Later, we discuss how to remove this assumption.

We generate a random subgraph H of G by sampling each edge of the graph G independently with probability $p = \Theta\left(\log n/\varepsilon^2\hat{t}\right)$ [6]. Using Chernoff bound, we can show that any particular cut of size k, $k \geq t$, in G is *well approximated* in H with probability at least $n^{-\Omega(k/\hat{t})}$. With this idea, consider the following Algorithm, stated informally, for minimum cut estimation.

Algorithm-Sketch (works with $\hat{t} \leq t$)

Step-1: Generate a random subgraph H of G by sampling each edge in G independently with probability $p = \Theta\left(\log n/\varepsilon^2\hat{t}\right)$. Note that H can be generated by using $\widetilde{O}(m/\hat{t})$ DEGREE and NEIGHBOR queries in expectation. We will discuss it in Algorithm 3 in Section 10.3.2.

Step-2 Determine $|\textsc{Cut}(H)|$ and report $\widetilde{t} = \frac{|\textsc{Cut}(H)|}{p}$ as a $(1 \pm \varepsilon)$-approximation of $|\textsc{Cut}(G)|$.

The number of queries made by the above algorithm is $\widetilde{O}(m/\hat{t})$ in expectation. But it produces correct output only when the vertex partition corresponding to $\textsc{Cut}(G)$ and $\textsc{Cut}(H)$ are the same. This is not the case always. If we can show that all cuts in G are approximately preserved in H, then Algorithm-Sketch produces correct output with high probability. The main bottleneck to prove it is that the total number of cuts in G can be exponential. A result of Karger (stated in the following lemma) will help us to make Algorithm-Sketch work.

Lemma 10.5 (Karger [95])**.** *For a given graph G the number of cuts in G of size at most $j \cdot |\textsc{Cut}(G)|$ is at most n^{2j}.*

Using the above lemma along with Chernoff bound, we can show the following.

Lemma 10.6. *Let G be a graph, $\hat{t} \leq t = |\textsc{Cut}(G)|$ and $\varepsilon \in (0,1)$. If $H(V(G), E_p)$ is a subgraph of G where each edge in $E(G)$ is included in E_p with probability $p = \min\left\{\frac{200\log n}{\varepsilon^2\hat{t}}, 1\right\}$ independently, then every cut of size k in G has size $pk(1 \pm \varepsilon)$ in H with probability at least $1 - \frac{1}{n^{10}}$.*

The above lemma implies the correctness of Algorithm-Sketch, which is for minimum cut estimation when we know a lower bound $\hat{t}$ of $|\textsc{Cut}(G)|$. But in general we do not know any such $\hat{t}$. To get around the problem, we start guessing $\hat{t}$ starting from $\frac{n}{2}$ each time reducing

[6] Though p can be more than 1 here, we will make it explicit in the formal description

$\hat{t}$ by a factor of 2. The guessing scheme gives the desired solution due to Lemma 10.6 coupled with the following intuition when $\hat{t} = \Omega(t \log n/\varepsilon^2)$ — if we generate a random subgraph H of G by sampling each edge with probability $p = \Theta\left(\log n/\varepsilon^2 \hat{t}\right)$, then H is disconnected with at least a constant probability. So, it boils down to a connectivity check in H. The intuition is formalized in the following lemma that can be proved using Markov's inequality.

Lemma 10.7. *Let G be a graph with $|V(G)| = n$, $\hat{t} \geq \frac{2000 \log n}{\varepsilon^2} |\textsc{Cut}(G)|$ and $\varepsilon \in (0,1)$. If $H(V(G), E_p)$ be a subgraph of G where each edge in $E(G)$ is included in E_p independently with probability $p = \min\left\{\frac{200 \log n}{\varepsilon^2 \hat{t}}, 1\right\}$, then H is connected with probability at most $\frac{1}{10}$.*

Before moving to the next section, we prove Lemmas 10.6 and 10.7 here.

Proof of Lemma 10.6. If $p = 1$, we are done as the graph H is exactly the same as that of G. So, without loss of generality assume that the graph G is connected. Otherwise, the lemma holds trivially as $|\textsc{Cut}(G)| = 0$, i.e., $\hat{t} = 0$ and $p = 1$. Hence, for the rest of the proof we will assume that $p = \frac{200 \log n}{\varepsilon^2 \hat{t}}$.

Consider a cut $\mathscr{C}_G(A_1, A_2)$ of size k in G. As we are sampling each edge with probability p, the expected size of the cut $\mathscr{C}_H(A_1, A_2)$ is pk. Using Chernoff bound (see Lemma 10.4 in Section 10.2.1), we get

$$\Pr\left(|\mathscr{C}_H(A_1, A_2) - pk| \geq \varepsilon pk\right) \leq e^{-\varepsilon^2 pk/3\hat{t}} = n^{-\frac{100k}{3\hat{t}}} \tag{10.1}$$

Note that here we want to show that every cut in G is approximately preserved in H. To do so, we will use Lemma 10.5 along with Equation (10.1) as follows. Let $Z_1, Z_2, \ldots, Z_\ell$ be the partition of the set of all cuts in G such that each cut in Z_j has the number of edges between $[j \cdot |\textsc{Cut}(G)|, (j+1)|\textsc{Cut}(G)|]$, where $\ell \leq \frac{n}{|\textsc{Cut}(G)|}$ and $j \leq \ell - 1$. From Lemma 10.5, $|Z_j| \leq n^{2j}$. Consider a particular Z_j, $j \in [\ell]$. Using the union bound along with Equation 10.1, the probability that there exists a cut in Z_j that is not approximately preserved in H is at most $\frac{1}{n^{11}}$. Taking union bound over all Z_j's, the probability that there exists a cut in G that is not approximately preserved is at most $\frac{1}{n^{10}}$. □

Proof of Lemma 10.7. Let $\mathscr{C}_G(A_1, A_2)$ be a minimum cut in G. Observe, $\mathbb{E}[|\mathscr{C}_H(A_1, A_2)|] = p|\mathscr{C}_G(A_1, A_2)| = p|\textsc{Cut}(G)|$. From Markov's inequality, we get $\Pr(G \text{ is connected}) \leq \Pr(|\mathscr{C}_G(A_1, A_2)| \geq 1) \leq \mathbb{E}[|\mathscr{C}_G(A_1, A_2)|] \leq \frac{1}{10}$. □

10.3.2 Formal Algorithm (Proof of Theorem 10.1)

In this Section, the main algorithm for minimum cut estimation is described in Algorithm 5 (ESTIMATOR) that makes multiple calls to Algorithm 4 (VERIFY-GUESS). The VERIFY-GUESS subroutine in turn calls Algorithm 3 (SAMPLE) multiple times.

Given degree sequence of the graph G, that can be obtained using degree queries, we will first show how to independently sample each edge of G with probability p using only NEIGHBOR queries.

Algorithm 3: SAMPLE(D, p)

Input: $D = \{d(i) : i \in [n]\}$, where $d(i)$ denotes the degree of the i-th vertex in the graph G, and $p \in (0,1]$.

Output: Return a subgraph $H(V, E_p)$ of $G(V, E)$ where each edge in $E(G)$ is included in E_p with probability p.

Set $q = 1 - \sqrt{1-p}$ and $m = \frac{\sum_{i=1}^{n} d_i}{2}$;

for *(each* $i \in [n]$*)* **do**

  for *(each* $j \in [d(i)]$ *with* $d(i) > 0$*)* **do**

    // Let r_j be the j-th neighbor of the i-th vertex;

    Add the edge (i, r_j) to the set E_p with probability q;

  end

end

Return the graph $H(V, E_p)$.

The following lemma proves the correctness of the above algorithm SAMPLE(D, p).

Lemma 10.8. SAMPLE(D, p) *returns a random subgraph* $H(V(G), E_p)$ *of* G *such that each edge* $e \in E$ *is included in* E_p *independently with probability* p*. Moreover, in expectation, the number of* NEIGHBOR *queries made by* SAMPLE(D, p) *is at most* $2pm$*.*

Proof. From the description of SAMPLE(D, p), it is clear that the probability that a particular edge $e \in E(G)$ is added to E_p with probability $1 - (1-q)^2 = p$.

Observe, $\mathbb{E}\left[\left|E_p\right|\right] = pm$. The bound on the number of NEIGHBOR queries now follows from the fact that SAMPLE(D, p) makes at most $2\left|E_p\right|$ many NEIGHBOR queries. □

One of the core ideas behind the proof of Theorem 10.1 is that, given an estimate $\hat{t}$ of t, we want to efficiently (in terms of number of local queries used by the algorithm) decide if $\hat{t} \leq t$ or if $\hat{t} \gtrsim \frac{\log n}{\varepsilon^2} \times t$. Using Algorithm 4, we will show that this can be done using $\widetilde{O}(m/\hat{t})$

many NEIGHBOR queries in expectation. Another interesting feature of Algorithm 4 is that, if estimate $\hat{t} \leq t$, then Algorithm 4 outputs an estimate which is a $(1 \pm \varepsilon)$-approximation of t.

Algorithm 4: VERIFY-GUESS$(D, \hat{t}, \varepsilon)$

Input: $D = \{d(i) : i \in [n]\}$, where $d(i)$ denotes the degree of the i-th vertex in the graph G and $m = \frac{1}{2}\sum_{i=1}^{n} d(i) \geq n-1$. Also, a guess $\hat{t}$, with $1 \leq \hat{t} \leq \frac{n}{2}$, for the size of the global minimum cut in G, and $\varepsilon \in (0,1)$.

Output: The algorithm should "ACCEPT" or "REJECT" $\hat{t}$, with high probability, depending on the following

- If $\hat{t} \leq |\text{CUT}(G)|$, then ACCEPT $\hat{t}$ and also output a $(1 \pm \varepsilon)$-approximation of $|\text{CUT}(G)|$
- If $\hat{t} \geq \frac{200 \log n}{\varepsilon^2} |\text{CUT}(G)|$, then REJECT $\hat{t}$

Set $p = \min\left\{\frac{200 \log^2 n}{\varepsilon^2 \hat{t}}, 1\right\}$;
Set $\Gamma = 100 \log n$ and Call SAMPLE(D, p) Γ times;
// Let $H_i(V, E_p^i)$ be the output of i-th call to SAMPLE(D, p), where $i \in [\Gamma]$
if *(at least* $\Gamma/2$ *many* H_i'*s are disconnected)* **then**
  REJECT $\hat{t}$
end
else if *(all* H_i*'s are connected)* **then**
  ACCEPT $\hat{t}$, find CUT(H_i) for any $i \in [\Gamma]$, and return $\tilde{t} = \frac{|\text{CUT}(H_i)|}{p}$.
end
else
  Return FAIL.
  // When we cannot decide between "REJECT" or "ACCEPT" it will return FAIL
end

The following lemma proves the correctness of Algorithm 4. The lemmas used in proof are Lemmas 10.6, 10.7 and 10.8.

Lemma 10.9. VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ *in expectation makes* $\widetilde{O}\left(\frac{m}{\hat{t}}\right)$ *many* NEIGHBOR *queries to the graph* G *and behaves as follows:*

(i) If $\hat{t} \geq \frac{2000 \log n}{\varepsilon^2} |\text{CUT}(G)|$, *then* VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ *rejects* $\hat{t}$ *with probability at least* $1 - \frac{1}{n^9}$.

(ii) If $\hat{t} \leq |\text{CUT}(G)|$, *then* VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ *accepts* $\hat{t}$ *with probability at least* $1 - \frac{1}{n^9}$. *Moreover, in this case,* VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ *reports an* $(1 \pm \varepsilon)$*-approximation to* CUT(G).

Proof. VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ calls SAMPLE(D, ε) for $\Gamma = 100 \log n$ times with p being set to $\min\left\{\frac{200 \log n}{\varepsilon^2 \hat{t}}, 1\right\}$. Recall from Lemma 10.8 that each call to SAMPLE(D, p) makes in

expectation at most $2pm$ many NEIGHBOR queries, and returns a random subgraph $H(V,E_p)$, where each edge in $E(G)$ is included in E_p with probability p. So, VERIFY-GUESS$(D,\hat{t},\varepsilon)$ makes in expectation $O(pm\log n) = \widetilde{O}(m/\hat{t})$ many NEIGHBOR queries and generates Γ many random subgraphs of G. The subgraphs are denoted by $H_1(V,E_p^1),\ldots,H_\Gamma(V,E_p^\Gamma)$.

(i) Let $\hat{t} \geq \frac{2000\log n}{\varepsilon^2}|\text{CUT}(G)|$. From Lemma 10.7, we have that H_i will be connected with probability at most $\frac{1}{10}$. Observe that in expectation, we get that at least $\frac{9\Gamma}{10}$ many H_i's will be disconnected. By Chernoff bound (see Lemma 10.4 in Section 10.2.1), the probability that at most $\frac{\Gamma}{2}$ many H_i's are disconnected is at most $\frac{1}{n^{10}}$. Therefore, VERIFY-GUESS$(D,\hat{t},\varepsilon)$ rejects any $\hat{t}$ satisfying $\hat{t} \geq \frac{2000\log n}{\varepsilon^2}|\text{CUT}(G)|$ with probability at least $1-\frac{1}{n^9}$.

(ii) Let $\hat{t} \leq |\text{CUT}(G)|$. Using Lemma 10.6, we have that every cut of size k in G has size $pk(1\pm\varepsilon)$ in H_i with probability at least $1-\frac{1}{n^{10}}$. Therefore, with probability at least $1-\frac{\Gamma}{n^{10}}$, for all $i\in[\Gamma]$, every cut of size k in G has size $pk(1\pm\varepsilon)$ in H_i. This implies that if $\hat{t} \leq |\text{CUT}(G)|$ then VERIFY-GUESS$(D,\hat{t},\varepsilon)$ accepts any $\hat{t}$ with probability at least $1-\frac{1}{n^9}$. Moreover, for any H_i, observe that $\frac{|\text{CUT}(H_i)|}{p}$ is a $(1\pm\varepsilon)$-approximation to $|\text{CUT}(G)|$. Hence, when $\hat{t} \leq |\text{CUT}(G)|$, VERIFY-GUESS$(D,\hat{t},\varepsilon)$ also returns a $(1\pm\varepsilon)$ approximation to $|\text{CUT}(G)|$ with probability $1-\frac{1}{n^9}$.

□

ESTIMATOR(ε) (Algorithm 5) will estimate the size of the minimum cut in G using DEGREE and NEIGHBOR queries. The main subroutine used by the algorithm will be VERIFY-GUESS$(D,\hat{t},\varepsilon)$.

The following lemma shows that with high probability ESTIMATOR(ε) correctly estimates the size of the minimum cut in the graph G, and it also bounds the expected number of queries used by the algorithm.

Lemma 10.10. ESTIMATOR(ε) *returns a* $(1\pm\varepsilon)$ *approximation to* $|\text{CUT}(G)|$ *with probability at least* $1-\frac{1}{n^8}$ *by making in expectation* $\min\left\{m+n,\frac{m}{t}\right\}$ *poly* $\left(\log n,\frac{1}{\varepsilon}\right)$ *queries and each query is either a* DEGREE *or a* NEIGHBOR *query to the unknown graph* G.

Proof. Without loss of generality, assume that n is a power of 2. If $m < n-1$ or if there exists a $i\in[n]$ such that $d_i = 0$ then the graph G is disconnected. In this case the algorithm ESTIMATOR(ε) makes n DEGREE queries and returns the correct answer. Thus we assume that $m \geq n-1$.

First, we prove the correctness and query complexity when the graph is connected, that is, $t \geq 1$. Note that ESTIMATOR(ε) calls VERIFY-GUESS($D, \hat{t}, \varepsilon$) for different values of $\hat{t}$ starting from $\frac{n}{2}$. Recall that $\kappa = \frac{2000 \log n}{\varepsilon^2}$. For a particular $\hat{t}$ with $\hat{t} \geq \kappa t$, VERIFY-GUESS($D, \hat{t}, \varepsilon$) does not REJECT $\hat{t}$ with probability at most $\frac{1}{n^9}$ by Lemma 10.9 (i). So, by the union bound, the probability that VERIFY-GUESS($D, \hat{t}, \varepsilon$) will either ACCEPT or FAIL for some $\hat{t}$ with $\hat{t} \geq \kappa t$, is at most $\frac{\log n}{n^9}$. Hence, with probability at least $1 - \frac{\log n}{n^9}$, we can say that VERIFY-GUESS($D, \hat{t}, \varepsilon$) rejects all $\hat{t}$ with $\hat{t} \geq \kappa t$.

Observe that, from Lemma 10.9 (ii), the first time $\hat{t}$ satisfies the following inequality $\frac{t}{2} < \hat{t} \leq t$, VERIFY-GUESS($D, \hat{t}, \varepsilon$) will accept $\hat{t}$ with probability at least $1 - \frac{1}{n^9}$. Therefore, for the first time VERIFY-GUESS($D, \hat{t}, \varepsilon$) will either ACCEPT or FAIL, then $\hat{t}$ satisfies the following inequality $\frac{t}{2} < \hat{t} < \kappa t$ with probability at least $1 - \frac{\log n + 1}{n^9}$. Let $\hat{t}_0$ denote the first time VERIFY-GUESS returns ACCEPT or FAIL. From the description of ESTIMATOR(ε), note that, we get $\hat{t}_u$ by dividing $\hat{t}_0$ by κ. Note that, with probability at least $1 - \frac{1 + \log n}{n^9}$, we have $\hat{t}_u < t$. We then call the procedure VERIFY-GUESS($D, \hat{t}, \varepsilon$) with $\hat{t} = \hat{t}_u$. By Lemma 10.9 (ii), VERIFY-GUESS($D, \hat{t}_u, \varepsilon$) will ACCEPT and report a $(1 \pm \varepsilon)$ approximation to t with probability at least $1 - \frac{1}{n^9}$.

We will now analyze the number of DEGREE and NEIGHBOR queries made by the algorithm. We make an initial n many queries to construct the set D. Then at the worst case, we call VERIFY-GUESS($D, \hat{t}, \varepsilon$) for $\hat{t} = \frac{n}{2}, \ldots, t'$ and $\hat{t} = \frac{t'}{\kappa} \geq \frac{t}{2\kappa}$, where $\frac{t}{2} < t' < \kappa t$. It is because VERIFY-GUESS($D, \hat{t}, \varepsilon$) accepts $\hat{t}$ with probability $1 - \frac{1}{n^9}$ when the first time $\hat{t}$ satisfy the inequality $\hat{t} \leq t$. Hence, by Lemma 10.9 and the facts that $n \leq \frac{m}{t}$ and $\hat{t}_u \geq \frac{t}{2\kappa}$ with probability at least $1 - \frac{\log n + 1}{n^9}$, in expectation the total number of queries made by the algorithm is at most $n + \log n \cdot \left(1 - \frac{\log n + 1}{n^9}\right) \cdot \widetilde{O}\left(\frac{2\kappa m}{t}\right) + \log n \cdot \left(\frac{\log n + 1}{n^9}\right) \cdot \widetilde{O}(m) = \widetilde{O}\left(\frac{m}{t}\right)$.

Note that each query made by ESTIMATOR(ε) is either a DEGREE or a NEIGHBOR query.

Now we analyze the case when $t = 0$. Observe that VERIFY-GUESS($D, \hat{t}, \varepsilon$) rejects all $\hat{t} \geq 1$ with probability $1 - \frac{\log n}{n^9}$, and therefore, ESTIMATOR(ε) will report $t = 0$. As we have called VERIFY-GUESS($D, \hat{t}, \varepsilon$) for all $\hat{t} = \frac{n}{2}, \ldots, 1$, the number of queries made by ESTIMATOR(ε), in the case when $t = 0$, is $\widetilde{O}(m) + n$. Note that the additional term of n in the bound comes from the fact that to compute D, the algorithm needs to make n many DEGREE queries. □

10.4 Lower bounds

In this Section, we prove Theorems 10.2 and 10.3 using reductions from suitable problems in communication complexity. In Section 10.4.1, we discuss about two party communication complexity along with the problems that will be used in our reductions. We will discuss the proofs of Theorems 10.2 and 10.3 in Section 10.4.2.

10.4.1 Communication Complexity

In two-party communication complexity there are two parties, Alice and Bob, that wish to compute a function $\Pi : \{0,1\}^N \times \{0,1\}^N \to \{0,1\} \cup \{0,1\}^n$ [7]. Alice is given $\mathbf{x} \in \{0,1\}^N$ and Bob is given $\mathbf{y} \in \{0,1\}^N$. Let x_i (y_i) denotes the i-th bit of $\mathbf{x}$ ($\mathbf{y}$). While the parties know the function Π, Alice does not know $\mathbf{y}$, and similarly Bob does not know $\mathbf{x}$. Thus they communicate bits following a pre-decided protocol $\mathscr{P}$ in order to compute $\Pi(\mathbf{x},\mathbf{y})$. We say a randomized protocol $\mathscr{P}$ computes Π if for all $(\mathbf{x},\mathbf{y}) \in \{0,1\}^N \times \{0,1\}^N$ we have $\Pr[\mathscr{P}(\mathbf{x},\mathbf{y}) = \Pi(\mathbf{x},\mathbf{y})] \geq 2/3$. The model provides the parties access to common random string of arbitrary length. The cost of the protocol $\mathscr{P}$ is the maximum number of bits communicated, where maximum is over all inputs $(\mathbf{x},\mathbf{y}) \in \{0,1\}^N \times \{0,1\}^N$. The communication complexity of the function is the cost of the most efficient protocol computing Π. For more details on communication complexity see [101]. We now define two functions k-INTERSECTION and FIND-k-INTERSECTION and discuss their communication complexity. Both these functions will be used in our reductions.

Definition 10.11 (FIND-k-INTERSECTION)**.** *Let $k,N \in \mathbb{N}$ such that $k \leq N$. Let $S = \{(\mathbf{x},\mathbf{y}) \in \{0,1\}^N \times \{0,1\}^N : \sum_{i=1}^N x_i y_i = k\}$. The* FIND-$k$-INTERSECTION *function on N bits is a partial function and is defined as* FIND-INT$_k^N : S \to \{0,1\}^N$*, and is defined as* FIND-INT$_k^N(\mathbf{x},\mathbf{y}) = \mathbf{z}$*, where $z_i = x_i y_i$ for each $i \in [N]$.*

Note that the objective is that at the end of the protocol Alice and Bob know $\mathbf{z}$.

Definition 10.12 (k-INTERSECTION)**.** *Let $k,N \in \mathbb{N}$ such that $k \leq N$. Let $S = \{(\mathbf{x},\mathbf{y}) : \sum_{i=1}^N x_i y_i = k \text{ or } k-1\}$. The k-*INTERSECTION *function on N bits is a partial function denoted by* INT$_k^N : S \to \{0,1\}$*, and is defined as follows:* INT$_k^N(\mathbf{x},\mathbf{y}) = 1$ *if $\sum_{i=1}^N x_i y_i = k$ and 0, otherwise.*

[7] The co-domain of Π looks odd, as the the co-domain is $\{0,1\}$ usually. However, we need $\{0,1\} \cup \{0,1\}^n$ to take care of all the problems in communication complexity we discuss in this chapter.

In communication complexity, the k-INTERSECTION function on N bits when $k = 1$ is known as DISJOINTNESS function on N. The following lemmas follow easily from the communication complexity of DISJOINTNESS (see [101]).

Lemma 10.13. *Let $k, N \in \mathbb{N}$ such that $k \leq cN$ for some constant $c < 1$. The randomized communication complexity of* FIND-k-INTERSECTION *function on N bits is $\Omega(N)$.*

Lemma 10.14. *Let $k, N \in \mathbb{N}$ such that $k \leq cN$ for some constant $c < 1$. The randomized communication complexity of k-*INTERSECTION *function on N bits (*INT_k^N*) is $\Omega(N)$.*

10.4.2 Proofs of Theorems 10.2 and 10.3

The proofs of Theorems 10.2 and 10.3 are inspired from the lower bound proof of Eden and Rosenbaum [54] for estimating MINCUT [8].

Proof of Theorem 10.2. We prove by giving a reduction from FIND-$t/2$-INTERSECTION on N bits. Without loss of generality assume that t is even. Let $\mathbf{x}$ and $\mathbf{y}$ be the inputs of Alice and Bob. Note that $\sum_{i=1}^{N} x_i y_i = t/2$.

We first discuss a graph $G_{(\mathbf{xy})}(V,E)$ that can be generated from $(\mathbf{x},\mathbf{y})$, such that $|V| = n$ and $|E| = m \geq 2nt$, and works as the 'hard' instance for our proof. Note that $G_{(\mathbf{x},\mathbf{y})}$ should be such that no useful information about the MINCUT can be derived by knowing only one of $\mathbf{x}$ and $\mathbf{y}$. Let $s = t + \sqrt{t^2 + (m-nt)/2}$ and $N = s^2$. In particular, $2t \leq s \leq 2t + 3\sqrt{m}$. Also, $s \geq \sqrt{m/2}$ and therefore $s = \Theta(\sqrt{m})$.

The graph $G_{(\mathbf{x},\mathbf{y})}$ and its properties:

$G_{(\mathbf{x},\mathbf{y})}$ has the following structure.

- $V = S_A \cup T_A \cup S_B \cup T_B \cup C$ such that $|S_A| = |T_A| = |S_B| = |T_B| = s$ and $|C| = n - 4s$. Let $S_A = \{s_i^A : i \in [s]\}$ and similarly $T_A = \{t_i^A : i \in [s]\}$, $S_B = \{s_i^B : i \in [s]\}$ and $T_B = \{t_i^B : i \in [s]\}$.
- Each vertex in C is connected to $2t$ different vertices in S_A.
- For $i, j \in [s]$: if $x_{ij} = y_{ij} = 1$, then $(s_i^A, t_j^B) \in E$ and $(s_i^B, t_j^A) \in E$; otherwise, $(s_i^A, t_j^A) \in E$ and $(s_i^B, t_j^B) \in E$.

Observation 10.15. *$G_{(\mathbf{x},\mathbf{y})}$ satisfies the following properties.*

[8] Note that Eden and Rosenbaum [54] stated the result in terms k-Edge Connectivity.

Property-1: *The degree of every vertex in C is $2t$. For any $v \notin C$, the neighbors of v inside C are fixed irrespective of* $\mathbf{x}$ *and* $\mathbf{y}$*; and the number of neighbors outside C is $s \geq 2t$.*

Property-2: *There are t edges between the vertex sets $(C \cup S_A \cup T_A)$ and $(S_B \cup T_B)$, and removing them $G_{(\mathbf{x},\mathbf{y})}$ becomes disconnected.*

Property-3: *Every pair of vertices $(S_A \cup T_A \cup C)$ is connected by at least $3t/2$ edge disjoint paths. Also, every pair of vertices in $(S_B \cup T_B)$ is connected by at least $3t/2$ edge disjoint paths.*

Property-4: *The set of t edges between the vertex sets $(C \cup S_A \cup T_A)$ and $(S_B \cup T_B)$ forms the unique global minimum cut of $G(\mathbf{x},\mathbf{y})$,*

Property-5: $x_{ij} = y_{ij} = 1$ *if and only if (s_i^A, t_j^B) and (s_i^B, t_j^A) are the edges in the unique global minimum cut of $G_{(\mathbf{x},\mathbf{y})}$.*

Proof. Property-1 and Property-2 directly follow from the construction. Now, we will prove Property-3. We first show that every pair of vertices $(S_A \cup T_A \cup C)$ is connected by at least $3t/2$ edge disjoint paths by breaking the analysis into the following cases.

(i) Consider $s_i^A, s_j^A \in S_A$, for $i, j \in [s]$. Under the promise that $\sum_{i=1}^{N} x_i y_i = t/2$, s_i^A, s_j^A have at least $s - t \geq 3t/2$ common neighbors in T_A and thus there are at least $3t/2$ edge disjoint paths connecting them.

(ii) Consider $s_i^A \in S_A$ and $t_j^A \in T_A$, for $i, j \in [s]$. Let $s_{j_1}^A, \ldots, s_{j_{3t/2}}^A$ be $3t/2$ distinct neighbors of t_j^A in S_A. Since, s_i^A has $3t/2$ common neighbors with each $s_{j_r}^A$, $r \in [3t/2]$, there is a matching of size $3t/2$. Denote this matching by $(t_{j_r}^A, s_{j_r}^A)$, $r \in [3t/2]$. Thus $(s_i^A, t_{j_r}^A), (t_{j_r}^A, s_{j_r}^A), (s_{j_r}^A, t_j^A)$, for $r \in [3t/2]$, forms a set of edge disjoint paths of size $3t/2$ from s_i^A to t_j^A, each of length 3. In case s_i^A is one of the neighbors of t_j^A, then one of the $3t/2$ paths gets reduced to (s_i^A, t_j^A), a length 1 path that is edge disjoint from the remaining paths.

(iii) Consider $u, v \in C$. Let $u_1, \ldots, u_{2t} \in S_A$ and $v_1, \ldots, v_{2t} \in S_A$ be the neighbors of u and v respectively in S_A. If for some $i, j \in [2t]$, $u_i = v_j$ then $(u, u_i), (u_i, v_j), (v_j, v)$ is a desired path. Thus, assume $u_i \neq v_j$ for all $i, j \in [2t]$. For all $i \in [2t]$, since u_i and v_i have at least $3t/2$ common neighbors in T_A we can find $3t/2$ edge disjoint paths $(u_i, t_i^A), (t_i^A, v_i)$, where $t_i^A \in T^A$. Existence of $3t/2$ edge disjoint paths from $u \in C$ to $v \in S_A$ can be proved as in (i). and from $u \in C$ to $v \in T_A$ can be proved as in (ii).

Similarly, we can show that every pair of vertices in $(S_B \cup T_B)$ is connected by $3t/2$ many edge disjoint paths.

Observe that Property-4 follows from Property-3, and Property-5 follows from the construction of $G_{(\mathbf{x},\mathbf{y})}$ and Property-4. □

Now, by contradiction assume that there exists an algorithm $\mathscr{A}$ that makes $o(m)$ queries to $G_{(\mathbf{x},\mathbf{y})}$ and finds all the edges of a global minimum cut with probability $2/3$. Now, we give a protocol $\mathscr{P}$ for FIND-$t/2$-INTERSECTION on N bits when the $\mathbf{x}$ and $\mathbf{y}$ are the inputs of Alice and Bob, respectively. Note that $x, y \in \{0,1\}^N$ such that $\sum_{i=1}^{N} x_i y_i = t/2$.

Protocol $\mathscr{P}$ for FIND-$t/2$-INTERSECTION**:**

Alice and Bob run the query algorithm $\mathscr{A}$ when the unknown graph is $G_{(\mathbf{x},\mathbf{y})}$. Now we explain how they simulate the local queries and random edge query on $G_{(\mathbf{x},\mathbf{y})}$ by communication. We would like to note that each query can be answered deterministically.

DEGREE **query:** By Property-1, the degree of every vertex does not depend on the inputs of Alice and Bob, and therefore any degree query can be simulated without any communication.

NEIGHBOR **query:** For $v \in C$, the set of $2t$ neighbors are fixed by the construction. So, any neighbor query involving any $v \in C$ can be answered without any communication. For $i \in [s]$ and $s_i^A \in S_A$, let $N_C(s_i^A)$ be the set of fixed neighbors of s_i^A inside C. So, by Property-1, $d(s_i^A) = \left|N_C(s_i^A)\right| + s$ [9]. The labels of the neighbors of s_i^A are such that the first $\left|N_C(s_i^A)\right|$ many neighbors are inside C, and they are arranged in a fixed but arbitrary order. For $j \in [s]$, the $(|N_C(v)| + j)$-th neighbor of s_i^A is either t_j^B or s_j^A depending on whether $x_{ij} = y_{ij} = 1$ or not, respectively. So, any neighbor query involving vertex in S_A can be answered by 2 bits of communication. Similar arguments also hold for the vertices in $S_B \cup T_A \cup T_B$.

ADJACENCY **query:** Observe that each adjacency query can be answered by at most 2 bits of communication, and it can be argued like the NEIGHBOR query.

RANDOM EDGE **query:** By Property-1, the degree of any vertex $v \in V$ is independent of the inputs of Alice and Bob. Alice and Bob use shared randomness to sample a vertex in V proportional to its degree. Let $r \in V$ be the sampled vertex. They again use shared randomness to sample an integer j in $[d(v)]$ uniformly at random. Then they determine the j-th neighbor of r using NEIGHBOR query. Observe that this procedure simulates a RANDOM EDGE query by using at most 2 bits of communication.

[9] $d(u)$ denotes the degree of the vertex u in $G_{(\mathbf{x},\mathbf{y})}$

Using the fact that $G_{(\mathbf{x},\mathbf{y})}$ satisfies Property-4 and 5, the output of algorithm $\mathscr{A}$ determines the output of protocol $\mathscr{P}$ for FIND-t/2-INTERSECTION. As each query of $\mathscr{A}$ can be simulated by at most two bits of communication by the protocol $\mathscr{P}$, the number of bits communicated is $o(m)$. Recall that $N = s^2$ and $s = \Theta(\sqrt{m})$. So, the number of bits communicated by Alice and Bob in $\mathscr{P}$ is $o(N)$. This contradicts Theorem 10.13. □

Proof of Theorem 10.3. The proof of this theorem uses the same construction as the one used in the proof of Theorem 10.2. The 'hard' communication problem to reduce from is t/2-INTERSECTION (see Definition 10.12) on N bits, where $N = s^2$ and $s = \Theta(\sqrt{m})$. □

10.5 Application of our approach to other cut problems

We discuss the application of our approach to other cut problems next.

Application of our approach to other cut problems

Sublinear time algorithm for Global minimum cut. For simplicity, the algorithm (Algorithm 5) presented for estimating global minimum cut is $\widetilde{\mathscr{O}}(m)$. But, our algorithm can be adapted to get a sublinear time algorithm (with time complexity $\widetilde{\mathscr{O}}\left(\frac{m}{t}\right)$) for estimating the size of the global minimum cut in the graph. We will sample $\widetilde{\mathscr{O}}\left(\frac{m}{\widehat{t}}\right)$ random edges with replacement from the graph G using $\widetilde{\mathscr{O}}\left(\frac{m}{\widehat{t}}\right)$ using local queries, where $\widehat{t}$ is the guess for the size of the global minimum, rather than sampling each edge with probability $p = \widetilde{\mathscr{O}}\left(\frac{1}{\widehat{t}}\right)$ as we have done in the Algorithm 4. Observe that, after finding the degrees of all the vertices, a random edge can be generated using $\mathscr{O}(1)$ local queries. The rest of the algorithm and its analysis can be adapted directly. Therefore, we have the following result.

Theorem 10.16. (Estimating Global minimum cut in sublinear time.) *There exists an algorithm, with* DEGREE *and* NEIGHBOR *query access to an unknown graph $G = (V,E)$, that solves the minimum cut estimation problem with high probability. With high probability, the time complexity and the query complexity of the algorithm is* $\min\left\{m+n, \frac{m}{t}\right\} \operatorname{poly}\left(\log n, \frac{1}{\varepsilon}\right)$.

Global minimum r-way cut. Global minimum r-cut, for a graph $G = ([n], E)$, $|V| = n$ and $|E| = m$, is a partition of the vertex set $[n]$ into r-sets $S_1, \ldots, S_r$ such that the following is minimized: $|\{\{i, j\} \in E \,:\, \exists k, \ell\,(k \neq \ell) \in [r],$ with $i \in S_k$ and $j \in S_\ell\}|$.

Let $\textsc{Cut}_r(G)$ denote the set of edges corresponding to a minimum r-cut, i.e., the edges that goes across different partitions, and by the size of minimum r-cut, we mean $|\textsc{Cut}_r(G)|$.

The sampling and verification idea used in the proof of Theorem 10.1 can be extended directly, together with [95, Corollary 8.2], to get the following result.

Theorem 10.17. *There exists an algorithm, with* DEGREE *and* NEIGHBOR *query access to an unknown graph $G = ([n], E)$, that with high probability outputs a $(1 \pm \varepsilon)$-approximation of the size of the minimum r-cut of G. The expected number of queries used by the algorithm is $\min\left\{m+n, \frac{m}{t_r}\right\} poly\left(r, \log n, \frac{1}{\varepsilon}\right)$, where $t_r = |\text{CUT}_r(G)|$.*

Minimum cuts in simple multigraphs. A graph with multiple edges between a pair of vertices in the graph but without any self loops are called *simple multigraphs*. If we have DEGREE and NEIGHBOR query access [1] to simple multigraphs then we can directly get the following generalization of Theorem 10.1.

Theorem 10.18. (Minimum cut estimation in simple multigraphs using local queries) *There exists an algorithm, with* DEGREE *and* NEIGHBOR *query access to an unknown simple multigraph $G = (V, E)$, that solves the minimum cut estimation problem with high probability. The expected number of queries used by the algorithm is $\min\left\{m+n, \frac{m}{t}\right\} poly\left(\log n, \frac{1}{\varepsilon}\right)$, where n is the number of vertices in the multigraph, m is the number of edges in the multigraph and t is the number of edges in a minimum cut.*

10.6 Summary

Our work first and foremost closes a gap in the query complexity of a fundamental problem of finding a minimum cut using local queries. The strength of our algorithm lies in its simplicity – it uses existing ingredients in a fashion suitable for the query framework. The crucial idea was to ensure that cuts are preserved in a sparsified graph in a query framework.

[1] For simple multigraphs, we will assume that the neighbors of a vertex are stored with multiplicities.

Algorithm 5: ESTIMATOR(ε)

Input: DEGREE and NEIGHBOR query access to an unknown graph G, and a parameter $\varepsilon \in (0,1)$.

Output: Either returns a $(1 \pm \varepsilon)$-approximation to $t = |\text{CUT}(G)|$ or FAIL

Find the degrees of all the vertices in G by making n many DEGREE queries;

// Let $D = \{d(1), \ldots, d(n)\}$, where $d(i)$ denotes the degree of the i-th vertex in G;

If $\exists i \in [n]$ such that $d(i) = 0$, then return $t = 0$ and QUIT. Otherwise, proceed as follows.

Find $m = \frac{1}{2} \sum_{i=1}^{n} d(i)$. If $m < n-1$, return $t = 0$ and QUIT. Otherwise, proceed as follows.

Set $\kappa = \frac{2000 \log n}{\varepsilon^2}$

Initialize $\hat{t} = \frac{n}{2}$.

while *(*$\hat{t} \geq 1$*)* **do**

  Call VERIFY-GUESS$(D, \hat{t}, \varepsilon)$.

  if *(*VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ *returns* REJECT*)* **then**

    set $\hat{t} = \frac{\hat{t}}{2}$ and continue.

  end

  else

    // Note that in this case VERIFY-GUESS$(D, \hat{t}, \varepsilon)$ either returns FAIL or ACCEPT.

    Set $\hat{t}_u = \max\left\{\frac{\hat{t}}{\kappa}, 1\right\}$.

    Call VERIFY-GUESS$(D, \hat{t}_u, \varepsilon)$.

    if *(*VERIFY-GUESS$(D, \hat{t}_u, \varepsilon)$ *returns* FAIL *or* REJECT*)* **then**

      return FAIL as the output of ESTIMATOR(ε)

    end

    else

      Let $\tilde{t}$ be the output of VERIFY-GUESS$(D, \hat{t}_u, \varepsilon)$.

      Return $\tilde{t}$ as the output of ESTIMATOR(ε).

    end

  end

end

Output: Return that the graph G is disconnected.

Chapter 11

Conclusion and Future Work

In this thesis we studied query complexity from various perspectives. There are many new open problems that have come out of this thesis. We discuss them next.

In Chapter 3, we studied quantum learning of n-bit functions of bounded Fourier sparsity, functions whose Fourier spectrum has at most k, $k \in [2^n]$, non-zero Fourier coefficients. We consider exact learning of this class Boolean functions using uniform quantum examples which is a generalization of the well studied uniform examples in the classical world. Each such uniform example can be generated by making a single query to the function. We gave an improvement over the best known classical algorithm, whose complexity is $\tilde{\Theta}(nk)$, by giving an $\tilde{O}(k^{1.5})$ learning algorithm. We also prove a quantum learning lower bound for this task. There are several questions for future investigation from this chapter. We look at some of then at a high level. Recall that the learning algorithm given in this chapter was in two phases. First, we would like to improve Phase 1 of our learning algorithm to $\tilde{O}(k)$ many samples, in a later chapter (Chapter 5) we gave a concrete direction towards this end. Phase 2 our learning algorithm is completely classical. It is natural to think of quantum algorithms for this phase. Indeed, the above questions are motivated by a lack of strong lower bound. Can we get a better lower bound than $\Omega(k \log k)$ for exactly learning k-Fourier sparse Boolean functions using uniform quantum examples?

In Chapter 4 we consider the relationship between quantum and classical sample complexities of exact active learning. In the model of active learning, a learner can query the truth table of the function it wants to learn instead of being restricted to examples drawn from the truth table according to some distribution. We showed that if a concept class $\mathscr{C}$ can be exactly learned using Q quantum membership queries, then it can also be learned using

$O\left(\frac{Q^2}{\log Q}\log|\mathscr{C}|\right)$ classical membership queries, improving upon the previously best known result by a $\log Q$ factor. We believe that the following conjecture should be true.

Conjecture 11.1. *For all concept classes $\mathscr{C}$ of Boolean-valued functions on a domain of size N we have:*

$$R(\mathscr{C}) = O(Q(\mathscr{C})^2 + Q(\mathscr{C})\log N).$$

In Chapter 5, we studied Chang's lemma and its relation to quantum learning theory. We gave a refinement of Chang's lemma for k-Fourier sparse Boolean functions. We also show how Chang's lemma is connected to quantum learning theory. There are several open problems left open by this chapter. Let us mention a few questions at high level. First, we have not found applications of our improvement of Chang's lemma in additive combinatorics and it is interesting to find such applications. The question of choice of threshold is very important in Chang's lemma type bounds (see [38]). It is also natural to look for tight bounds of the flavour of Chang's lemma when the threshold is smaller than max-rank entropy.

In Part II, (Chapter 6, Chapter 7 and Chapter 8), we studied the relation between quantum query and communication complexity. It is well known, in the classical world, that the query algorithm of a function $f : \{-1,1\}^n \to \{-1,1\}$ can be simulated to give a communication protocol of a related communication problem (obtained by composing f, the outer function, with inner function $G : \{-1,1\}^2 \to \{-1,1\}$) with only a constant overhead. The best known such simulation theorem in the quantum world, due to Buhrman, Cleve and Wigderson, 1998, has an overhead that is logarithmic in input size of the function. We constructed the first function F that witness this logarithmic gap i.e.

$$Q^{cc}(F) = \Omega(Q(F)\log n).$$

We also gave a general recipe of constructing functions that witness separation between quantum query-to-communication simulation. Finally, we explore the role of symmetry on this simulation problem. We showed that if the outer function f is symmetric then no overhead required in quantum query-to-communication simulation. On the other hand, we constructed a transitive function for which the logarithmic overhead was necessary. While our works resolve many problems in this line of research, it will be interesting to see whether our communication protocol can be improved in the following sense: We gave an efficient communication protocol for $f \circ G$ where f is a symmetric function and $G \in \{\mathsf{AND}_2, \mathsf{XOR}_2\}$. However our communication protocol assumes a large amount of shared entanglement, can we reduce this?

In Chapter 9 we considered the communication complexity of the Disjointness problem when the promised set systems has a bounded VC dimension. The setting of the problems was as follows: there is a collection of sets of $[n]$ of VC Dimension at most d which is known to both Alice and Bob, and their inputs for solving the Disjointness problem are promised to come from this set system. We showed that there exists such a set system such that the lower bound on communication complexity of the Disjointness problem on this promised set system is

$$\Omega\left(d\frac{\log(n/d)}{\log\log(n/d)}\right),$$

almost matched the trivial deterministic upper bound of $((d\log n)$. However, we believe that the $\log\log(n/d)$ factor is the above expression can be removed. Recalling the notation from this chapter (Chapter 9), we make the following conjecture.

Conjecture 11.2. *There exists $\mathscr{S} \subseteq 2^{[n]}$ with VC-dim$(\mathscr{S}) \leq d$ and $R(\mathsf{DISJ}_n\,|_{\mathscr{S}\times\mathscr{S}}) = \Omega\left(d\log\frac{n}{d}\right)$.*

Recall $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$ with VC-Dim$(\mathscr{T}) = 2d$ construction from Section 9.3.1, that served as the hard instance for the proof of Theorem 9.12 and Theorem 9.13. The same G and $\mathscr{T}$ cannot be the hard instance for the proof of Conjecture 11.2 because of the following result.

Theorem 11.3. *Let us consider $G \subset \mathbb{Z}^2$ with $|G| = n$ and $\mathscr{T} \subseteq 2^G$ with VC-Dim$(\mathscr{T}) = 2d$ as defined in Section 9.3.1. Also, recall the definition of $\mathscr{T}_1$ and $\mathscr{T}_2$. There exists a randomized communication protocol that can, $\forall A \in \mathscr{T}_1$ and $\forall B \in \mathscr{T}_2$, can compute $\mathsf{DISJ}_G\,|_{\mathscr{T}\times\mathscr{T}}\,(A,B)$, with probability at least $2/3$, and uses $O\left(\frac{d\log d\log\frac{n}{d}}{\log\log\frac{n}{d}}\cdot\log\log\log\frac{n}{d}\right)$ bits of communication.*

We use the following observation to prove the above theorem.

Observation 11.4. *Let us consider the communication problem $\textsc{Gcd}_k(a,b)$, where Alice and Bob get a and b respectively from $\{1,\ldots,k\}$, and the objective is for both the players to compute $\gcd(a,d)$. Then there exists a randomized protocol, with success probability at least $1-\delta$, for $\textsc{Gcd}_k$ that uses $O\left(\frac{\log k}{\log\log k}\cdot\log\log\log k\cdot\log\frac{1}{\delta}\right)$ bits of communication.*

Proof. We will give a protocol P for the case when $\delta = 1/3$ that uses $O\left(\frac{\log k}{\log\log k}\cdot\log\log\log k\right)$ bits of communication. By repeating $O\left(\log\frac{1}{\delta}\right)$ times protocol $\mathscr{P}$ and reporting the majority of the outcomes as the output, we will get the correct answer with probability at least $1-\delta$. Both Alice and Bob generate all the prime numbers $\pi_1,\ldots,\pi_t$ between 1 and k. From the Prime Number Theorem, we known that $t = \Theta\left(\frac{k}{\log k}\right)$. Alice and Bob separately, construct the sets S_a and S_b that contain the prime numbers that divides a and b respectively. Note that

$|S_a|$ and $|S_b|$ is bounded by $O\left(\frac{\log k}{\log\log k}\right)$.[1] Alice and Bob compute $S_a \cap S_b$ by solving *Sparse Set Intersection* problem on input S_a and S_b using $O\left(\frac{\log k}{\log\log k}\right)$ bits of communication [30]. For $p \in S_a \cap S_b$, let $\alpha_{p,a}$ and $\alpha_{p,b}$ denote the exponent of p in a and b, respectively. Observe that

$$\gcd(a,b) = \prod_{p \in S_a \cap S_b} p^{\min\{\alpha_{p,a}, \alpha_{p,b}\}}.$$

For each $p \in S_a$, Alice sends $\alpha_{p,a}$ to Bob. Number of bits of communication required to send the exponents of all the primes in $S_a \cap S_b$, is

$$\begin{aligned}
|S_a \cap S_b| + \sum_{p \in S_a \cap S_b} \log(\alpha_{p,a}) &\leq O\left(\frac{\log k}{\log\log k}\right) + |S_a \cap S_b| \log\left(\frac{\sum_{p \in S_a \cap S_b} \alpha_{p,a}}{|S_a \cap S_b|}\right) \\
&\leq O\left(\frac{\log k}{\log\log k}\right) + |S_a \cap S_b| \log\left(\frac{\log k}{|S_a \cap S_b|}\right) \\
&\leq O\left(\frac{\log k}{\log\log k} \cdot \log\log\log k\right)
\end{aligned}$$

In the above inequalities, we used the facts that $|S_a \cap S_b| = O\left(\frac{\log k}{\log\log k}\right)$, $\sum_{p \in S_a \cap S_b} \alpha_{p,a} \leq \log k$ and $\log x$ is a concave function. After getting the exponents $\alpha_{p,a}$ of the primes $p \in S_a \cap S_b$ from Alice, Bob also sends the exponents $\alpha_{p,b}$ of the primes $p \in S_a \cap S_b$ to Alice using $O\left(\frac{\log k}{\log\log k} \log\log\log k\right)$ bits of communication to Alice. Since both Alice and Bob now know the set $S_a \cap S_b$, and the exponents $\alpha_{p,a}$ and $\alpha_{p,b}$ for all $p \in S_a \cap S_b$, both of them can compute $\gcd(a,b)$. Total number of bits communicated in this protocol is $O\left(\frac{\log k}{\log\log k} \log\log\log k\right)$. □

We will now give the proof of Theorem 11.3.

Proof of the Theorem 11.3. Consider the case when $d = 1$. From the description of G and $\mathscr{T}$ in Section 9.3.1, we can say that $G = G_{(0,0)}$, where $G_{(0,0)} = \{(x,y) \in \mathbb{Z}^2 : 0 \leq x,y \leq \sqrt{n}\}$ [2]. Moreover, each set in $\mathscr{T}_1$ is a set of points present in a straight line of non-negative slope that passes through two points of $G_{(0,0)}$ with one point being $(0,0)$ and each set in $\mathscr{T}_2$ is a set of points present in a vertical straight line that passes through exactly $\sqrt{n}$ many grid points. Keeping Claims 9.15 and 9.16 in mind, we will be done if we can show the existence of a randomized communication protocol for computing the function $\mathsf{DISJ}_G|_{\mathscr{T} \times \mathscr{T}}$, with probability of success at least $1 - \delta$ and number of bits communicated by the protocol being bounded by $O\left(\frac{\log n}{\log\log n} \cdot \log\log\log n \cdot \log\frac{1}{\delta}\right)$, for the special case when $d = 1$. This is because

[1] The product of first t prime numbers is $e^{(1+o(1))t\log t}$.

[2] With out loss of generality assume that $\sqrt{n}$ is an integer

for general d, we will be solving d instances of the above problem, with the number of points in each grid being $\frac{n}{d}$ [3] and setting $\delta = \frac{1}{3d}$ for each of the d instances.

Protocol for $d = 1$. Alice and Bob get A and B from $\mathscr{T}_1$ and $\mathscr{T}_2$, respectively. Let A is generated by the straight line L_A and B is generated by L_B, where L_A is a straight line with non-negative slope and L_B is a vertical line. If L_A is a horizontal one : Alice just sends this information to Bob and then both report that $A \cap B \neq \emptyset$. If L_A is a vertical line : Alice sends this information to Bob and he reports $A \cap B \neq \emptyset$ if and only if L_B passes through origin. Now assume that L_A is neither a horizontal nor a vertical line. Let the equation of L_A be $y = \frac{p}{q}x$, where $1 \leq p, q \leq \sqrt{n}$, and p and q are relatively prime to each other. Also, let equation of Bob's line L_B be $x = r$, where $0 \leq r \leq \sqrt{n}$. Observe that $A \cap B \neq \emptyset$ if and only if L_A and L_B intersects at a point of $G_{(0,0)}$. Moreover, L_A and L_B intersects at a grid point if and only if q divides r and $1 \leq \frac{pr}{q} \leq \sqrt{n}$. So, Alice and Bob run the communication protocol for $\textsc{Gcd}_{\sqrt{n}}(q, r)$ to decide whether $q = \gcd(q, r)$. If $q = \gcd(q, r)$ and $1 \leq \frac{pr}{q} \leq \sqrt{n}$ (again Alice and Bob can decide this using $O(1)$ bits of communications) then $A \cap B \neq \emptyset$, otherwise $A \cap B = \emptyset$. Alice and Bob can decide if $q = \gcd(q, r)$ and $1 \leq \frac{pr}{q} \leq \sqrt{n}$ using $O(1)$ bits of communication.

The communication cost of our protocol is dominated by the communication complexity of $\textsc{Gcd}_{\sqrt{n}}(q, r)$, which is equal to $O\left(\frac{\log n}{\log\log n} \log\log\log n \log\frac{1}{\delta}\right)$ by Observation 11.4. □

In Chapter 10 we studied the problem of approximating the size of global minimum cut in the local query model. Our algorithm, along with a result of Eden and Rosenbaum, 2018, gave an almost tight algorithm for the problem of estimating global minimum cut, resolving this problem in the local query model. Our algorithms also extent to r-way cuts as discussed in Section 10.5. It will be interesting to see if we can prove matching lower bound for the problem to approximating r-way cuts in local query model.

[3] Recall that we have assumed, without loss of generality, that d divides n.

List of publications on which this thesis is based

Published

- **Two new results about quantum exact learning.**

  with Srinivasan Arunachalam, Sourav Chakraborty, Troy Lee, and Ronald de Wolf.

  QUANTUM, 2021, Volume 5, Page 587, DOI: 10.22331/q-2021-11-24-587.

  International Colloquium on Automata, Languages, and Programming (ICALP), Volume 132, Pages 16:1—16:15, 2019, DOI: 10.4230/LIPIcs.ICALP.2019.16.

- **Quantum Query-To-Communication Simulation Needs a Logarithmic Overhead.**

  with Sourav Chakraborty, Arkadev Chattopadhyay, and Nikhil Mande.

  Computational Complexity Conference (CCC), Volume 169, Pages 32:1—32:15, 2020, DOI: 10.4230/LIPIcs.CCC.2020.32.

- **Disjointness Through the Lens of Vapnik-Chervonenkis Dimension: Sparsity and Beyond.**

  with Anup Bhattacharya, Sourav Chakraborty, Arijit Ghosh, and Gopinath Mishra.

  International Conference on Randomization and Computation (RANDOM), Volume 176, Pages 23:1—23:15, 2020, DOI: 10.4230/LIPIcs.APPROX/RANDOM.2020.23.

- **Query Complexity of Global Minimum Cut.**

  with Arijit Bishnu, Arijit Ghosh, and Gopinath Mishra.

  International Conference on Approximation Algorithms for Combinatorial Optimization Problems (APPROX), Volume 207, Pages 6:1—6:15, 2021, DOI: 10.4230/LIPIcs.APPROX/RANDOM.2021.6.

- **Tight Chang's-lemma-type bounds for Boolean functions.**

  with Sourav Chakraborty, Nikhil S. Mande, Rajat Mittal, Tulasimohan Molli, Manaswi Paraashar and Swagato Sanyal.

  Foundations of Software Technology and Theoretical Computer Science (FSTTCS), 2021, Volume 213, Pages 10:1–10:22, DOI: 10.4230/LIPIcs.FSTTCS.2021.10.

Continued to the next page.

Accepted

- **Symmetry and Quantum Query-to-Communication Simulation.**

  with Sourav Chakraborty, Arkadev Chattopadhyay, Peter Høyer, Nikhil S. Mande and Ronald de Wolf.

  To appear in To appear in *International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2022. *arxiv:2012.05233*.

Submitted

- **Disjointness Through the Lens of Vapnik-Chervonenkis Dimension: Sparsity and Beyond.**

  with Anup Bhattacharya, Sourav Chakraborty, Arijit Ghosh, and Gopinath Mishra.

  Submitted to *Computational Complexity*. Date of Submission: 09/04/2021. Submission ID: CC M914.

References

[1] Scott Aaronson and Andris Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1(1):47–79, 2005. Earlier version in FOCS'03. quant-ph/0303041.

[2] Anil Ada, Omar Fawzi, and Hamed Hatami. Spectral norm of symmetric functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, pages 338–349, 2012.

[3] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price, and S. Stanisic. Advances in quantum machine learning, 9 Dec 2015. arXiv:1512.02900.

[4] D. Aharonov, A. W. Harrow, Z. Landau, D. Nagaj, M. Szegedy, and U. V. Vazirani. Local tests of global entanglement and a counterexample to the generalized area law. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 246–255, 2014. doi: 10.1109/FOCS.2014.34. URL https://doi.org/10.1109/FOCS.2014.34.

[5] K. J. Ahn, S. Guha, and A. McGregor. Graph Sketches: Sparsification, Spanners, and Subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS*, pages 5–14, 2012.

[6] M. Aliakbarpour, A. S. Biswas, T. Gouleakis, J. Peebles, R. Rubinfeld, and A. Yodpinyanee. Sublinear-Time Algorithms for Counting Star Subgraphs via Edge Sampling. *Algorithmica*, 80(2):668–697, 2018.

[7] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Earlier version in STOC'00. quant-ph/0002066.

[8] A. Ambainis and R. de Wolf. How low can approximate degree and quantum query complexity be for total Boolean functions? *Computational Complexity*, 23(2):305–322, 2014. Earlier version in CCC'13.

[9] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1987.

[10] A. Anshu, N. G. Boddu, and D. Touchette. Quantum log-approximate-rank conjecture is also false. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 982–994, 2019.

[11] S. Arunachalam and R. de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, 2017. arXiv:1701.06806.

[12] S. Arunachalam and R. de Wolf. Optimal quantum sample complexity of learning algorithms. *Journal of Machine Learning Research*, 19, 2018. Earlier version in CCC'17. arXiv:1607.00932.

[13] S. Arunachalam, S. Chakraborty, M. Koucký, N. Saurabh, and R. de Wolf. Improved bounds on Fourier entropy and min-entropy. In *Prooceedings of the 37th International Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 45:1–45:19, 2020. doi: 10.4230/LIPIcs.STACS.2020.45. URL https://doi.org/10.4230/LIPIcs.STACS.2020.45.

[14] S. Assadi, M. Kapralov, and S. Khanna. A Simple Sublinear-Time Algorithm for Counting Arbitrary Subgraphs via Edge Sampling. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference, ITCS*, pages 6:1–6:20, 2019.

[15] A. Atıcı and R. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2009. arXiv:0707.3479.

[16] L. Babai, P. Frankl, and J. Simon. Complexity classes in communication complexity theory (preliminary version). In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science, FOCS*, pages 337–347, 1986.

[17] Z. Bar-Yossef, T. S. Jayram, R. Kumar, and D. Sivakumar. An Information Statistics Approach to Data Stream and Communication Complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.

[18] H. Barnum, M. Saks, and M. Szegedy. Quantum query complexity and semi-definite programming. In *Proceedings of 18th IEEE Conference on Computational Complexity*, pages 179–193, 2003.

[19] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.

[20] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. quant-ph/9802049.

[21] E. Ben-Sasson, N. Ron-Zewi, M. Tulsiani, and J. Wolf. Sampling-based proofs of almost-periodicity results and algorithmic applications. In *41st International Colloquium on Automata, Languages, and Programming ICALP 2014*, volume 8572 of *Lecture Notes in Computer Science*, pages 955–966. Springer, 2014.

[22] P. A. Benioff. Quantum mechanical Hamiltonian models of Turing machines. *Journal of Statistical Physics*, 29(3):515–546, 1982.

[23] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[24] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

[25] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd. Quantum machine learning. *Nature*, 549(7671), 2017. arXiv:1611.09347.

[26] E. Blais, J. Brody, and K. Matulef. Property Testing Lower Bounds via Communication Complexity. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC*, pages 210–220, 2011.

[27] J. Bourgain. An improved estimate in the restricted isometry problem. In *Geometric Aspects of Functional Analysis*, volume 2116 of *Lecture Notes in Mathematics*, pages 65–70. Springer, 2014.

[28] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.

[29] M. Braverman, A. Garg, D. Pankratov, and O. Weinstein. From information to exact communication. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 151–160. ACM, 2013. doi: 10.1145/2488608.2488628. URL https://doi.org/10.1145/2488608.2488628.

[30] J. Brody, A. Chakrabarti, R. Kondapally, D. P. Woodruff, and G. Yaroslavtsev. Beyond Set Disjointness: The Communication Complexity of Finding the Intersection. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 106–113, 2014.

[31] J. Bruck. Harmonic analysis of polynomial threshold functions. *SIAM J. Discrete Math.*, 3(2):168–177, 1990.

[32] J. Bruck and R. Smolensky. Polynomial threshold functions, AC^0 functions, and spectral norms. *SIAM Journal on Computing*, 21(1):33–42, 1992. doi: 10.1137/0221003. URL https://doi.org/10.1137/0221003.

[33] N. H. Bshouty and J. C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136—-1153, 1999. Earlier version in COLT'95.

[34] H. Buhrman, R. C., and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 63–68, 1998.

[35] H. Buhrman, I. Newman, H. Röhrig, and R. de Wolf. Robust polynomials and quantum algorithms. *Theory Comput. Syst.*, 40(4):379–395, 2007.

[36] Mark Bun and Justin Thaler. Dual lower bounds for approximate degree and markov-bernstein inequalities. *Information and Computation*, 243:2–25, 2015. doi: 10.1016/j.ic.2014.12.003. URL https://doi.org/10.1016/j.ic.2014.12.003.

[37] E. J. Candés and T. Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12): 5406–5425, 2006.

[38] S. Chakraborty, N. S. Mande, R. Mittal, T. Molli, M. Paraashar, and S. Sanyal. Tight chang's-lemma-type bounds for boolean functions, 2021.

[39] Siu On Chan, James R. Lee, Prasad Raghavendra, and David Steurer. Approximate constraint satisfaction requires large LP relaxations. *J. ACM*, 63(4):34:1–34:22, 2016.

[40] M. C. Chang. A polynomial bound in Freimans theorem. *Duke Mathematics Journal*, 113(3):399–419, 2002.

[41] A. Chattopadhyay. *Circuits, Communication and Polynomials*. PhD thesis, McGill University, 2009.

[42] A. Chattopadhyay and N. Mande. A lifting theorem with applications to symmetric functions. In *Proceedings of the 37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, pages 23:1–23:14, 2017.

[43] A. Chattopadhyay, N. Mande, and S. Sherif. The log-approximate-rank conjecture is false. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 42–53, 2019.

[44] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, 2001.

[45] M. Cheraghchi, V. Guruswami, and A. Velingker. Restricted isometry of Fourier matrices and list decodability of random linear codes. *SIAM Journal on Computing*, 42(5):1888–1914, 2013.

[46] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158, New York, NY, USA, 1971. Association for Computing Machinery. ISBN 9781450374644. doi: 10.1145/800157.805047. URL https://doi.org/10.1145/800157.805047.

[47] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[48] A. Dasgupta, R. Kumar, and D. Sivakumar. Sparse and Lopsided Set Disjointness via Information Theory. In *Proceedings of the 16th International Workshop on Randomization and Computation, RANDOM*, pages 517–528, 2012.

[49] R. de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1):337–353, 2002. doi: https://doi.org/10.1016/S0304-3975(02)00377-8.

[50] R. de Wolf. Quantum communication and complexity. *Theor. Comput. Sci.*, 287(1): 337–353, 2002. doi: 10.1016/S0304-3975(02)00377-8. URL https://doi.org/10.1016/S0304-3975(02)00377-8.

[51] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

[52] D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 1st edition, 2009.

[53] V. Dunjko and H. Briegel. Machine learning & artificial intelligence in the quantum domain, 8 Sep 2017. arXiv:1709.02779.

[54] T. Eden and W. Rosenbaum. Lower Bounds for Approximating Graph Parameters via Communication Complexity. In *Proceedings of the 21st International Conference on Approximation Algorithms for Combinatorial Optimization Problems, APPROX*, pages 11:1–11:18, 2018.

[55] T. Eden and W. Rosenbaum. On Sampling Edges Almost Uniformly. In *Proceedings of the 1st Symposium on Simplicity in Algorithms, SOSA*, pages 7:1–7:9, 2018.

[56] T. Eden, A. Levi, D. Ron, and C. Seshadhri. Approximately Counting Triangles in Sublinear Time. *SIAM J. Comput.*, 46(5):1603–1646, 2017.

[57] T. Eden, D. Ron, and C. Seshadhri. On Approximating the Number of k-Cliques in Sublinear Time. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 722–734, 2018.

[58] U. Feige. On Sums of Independent Random Variables with Unbounded Variance and Estimating the Average Degree in a Graph. *SIAM J. Comput.*, 35(4):964–984, 2006.

[59] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[60] R. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.

[61] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and Systems Sciences*, 65(4):612–625, 2002. doi: 10.1016/S0022-0000(02)00019-3. URL https://doi.org/10.1016/S0022-0000(02)00019-3.

[62] E. Friedgut and G. Kalai. Every monotone graph property has a sharp threshold. *Proceedings of the American Mathematical Society*, 124(10):2993–3002, 1996.

[63] E. Friedgut, J. Kahn, G. Kalai, and N. Keller. Chvátal's conjecture and correlation inequalities. *J. Comb. Theory, Ser. A*, 156:22–43, 2018.

[64] M. Ghaffari and B. Haeupler. Distributed algorithms for planar networks II: low-congestion shortcuts, mst, and min-cut. In *Proceedings of the 2016 Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 202–219. SIAM, 2016.

[65] M. Ghaffari and F. Kuhn. Distributed minimum cut approximation. In *Distributed Computing*, volume 8205 of *Lecture Notes in Computer Science*, pages 1–15, 2013.

[66] M. Ghaffari and K. Nowicki. Massively parallel algorithms for minimum cut. pages 119–128. ACM, 2020.

[67] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[68] O. Goldreich and D. Ron. Approximating Average Parameters of Graphs. *Random Structures & Algorithms*, 32(4):473–493, 2008.

[69] O. Goldreich, S. Goldwasser, and D. Ron. Property Testing and its Connection to Learning and Approximation. *J. ACM*, 45(4):653–750, 1998.

[70] M. Gonen, D. Ron, and Y. Shavitt. Counting Stars and Other Small Subgraphs in Sublinear-Time. *SIAM Journal on Discrete Mathematics*, 25(3):1365–1411, 2011.

[71] P. Gopalan, R. O'Donnell, R. A. Servedio, A. Shpilka, and K. Wimmer. Testing Fourier dimensionality and sparsity. *SIAM Journal on Computing*, 40(4):1075–1100, 2011. Earlier version in ICALP'09.

[72] A. Graur, T. Pollner, V. Ramaswamy, and S. M. Weinberg. New Query Lower Bounds for Submodular Function Minimization. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference, ITCS*, volume 151, pages 64:1–64:16, 2020.

[73] B. Green. Arithmetic progressions in sumsets. *Geometric and Functional Analysis GAFA*, 12(3):584–597, 2002.

[74] B. Green and I. Z. Ruzsa. Freiman's theorem in an arbitrary abelian group. *Journal of the London Mathematical Society*, 75(1):163–175, 2007.

[75] B. Green and T. Sanders. Boolean functions with small spectral norm. *Geometric and Functional Analysis GAFA*, 18:144–162, 2008.

[76] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 212–219, 1996.

[77] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[78] A. Hajnal, W. Maass, and G. Turán. On the communication complexity of graph properties. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC*, pages 186–191, 1988.

[79] S. Har-Peled. *Geometric Approximation Algorithms*. Number 173. American Mathematical Soc., 2011.

[80] A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171.

[81] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. In *Proceedings of 44th ACM STOC*, pages 563–578, 2012.

[82] J. Håstad and A. Wigderson. The Randomized Communication Complexity of Set Disjointness. *Theory of Computing*, 3(1):211–219, 2007.

[83] H. Hatami, K. Hosseini, and S. Lovett. Sign rank vs discrepancy. In *Proceedings of the 35th Computational Complexity Conference (CCC)*, pages 18:1–18:14, 2020. doi: 10.4230/LIPIcs.CCC.2020.18. URL https://doi.org/10.4230/LIPIcs.CCC.2020.18.

[84] I. Haviv and O. Regev. The list-decoding size of Fourier-sparse Boolean functions. *ACM Transactions on Computation Theory*, 8(3):10:1–10:14, 2016. Earlier version in CCC'15. arXiv:1504.01649.

[85] J. D. Hidary. *A Brief History of Quantum Computing*, pages 11–16. Springer International Publishing, Cham, 2019. ISBN 978-3-030-23922-0. doi: 10.1007/978-3-030-23922-0_2.

[86] K. Hosseini, S. Lovett, and G. Yaroslavtsev. Optimality of linear sketching under modular updates. In *34th Computational Complexity Conference, CCC 2019*, volume 137 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[87] P. Høyer and R. de Wolf. Improved quantum communication complexity bounds for disjointness and equality. In *Proceedings of the 19th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 299–310, 2002.

[88] P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proceedings of the 30th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 2719 of *Lecture Notes in Computer Science*, pages 291–299. Springer, 2003. quant-ph/0304052.

[89] R. Impagliazzo, C. Moore, and A. Russell. An entropic proof of Chang's inequality. *SIAM J. Discret. Math.*, 28(1):173–176, 2014.

[90] P. Indyk and M. Kapralov. Sample-optimal Fourier sampling in any constant dimension. In *Proceedings of 55th IEEE FOCS*, pages 514–523, 2014.

[91] K. Iwama, H. Nishimura, R. Raymond, and S. Yamashita. Unbounded-error one-way classical and quantum communication complexity. In *Proceedings of the International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 110–121. Springer, 2007.

[92] T. S. Jayram, R. Kumar, and D. Sivakumar. Two Applications of Information Complexity. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, STOC*, pages 673–682, 2003.

[93] B. Kalyanasundaram and G. Schnitger. The Probabilistic Communication Complexity of Set Intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

[94] M. Kapralov, Y. T. Lee, C. Musco, C. Musco, and A. Sidford. Single pass spectral sparsification in dynamic streams. In *Proceedings of the 55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 561–570, 2014.

[95] D. R. Karger. Global Min-cuts in RNC, and Other Ramifications of a Simple Min-Cut Algorithm. In *Proceedings of the 4th Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, SODA*, pages 21–30, 1993.

[96] D. R. Karger and C. Stein. An $\widetilde{\mathscr{O}}\left(n^2\right)$ Algorithm for Minimum Cuts. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing, STOC*, pages 757–765, 1993.

[97] T. Kaufman, M. Krivelevich, and D. Ron. Tight Bounds for Testing Bipartiteness in General Graphs. *SIAM J. Comput.*, 33(6):1441–1483, 2004.

[98] K. Kawarabayashi and M. Thorup. Deterministic edge connectivity in near-linear time. *J. ACM*, 66(1):4:1–4:50, 2019.

[99] H. Klauck. Lower bounds for quantum communication complexity. *SIAM J. Comput.*, 37(1):20–46, 2007. doi: 10.1137/S0097539702405620. URL https://doi.org/10.1137/S0097539702405620. Earlier version in FOCS'01.

[100] M. Krause and P. Pudlák. On the computational power of depth-2 circuits with threshold and modulo gates. *Theor. Comput. Sci.*, 174(1-2):137–156, 1997.

[101] E. Kushilevitz. Communication complexity. In *Advances in Computers*, volume 44, pages 331–360. Elsevier, 1997.

[102] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, USA, 1996. ISBN 0521560675.

[103] E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[104] T. Lee and A. Shraibman. Lower bounds in communication complexity. *Foundations and Trends in Theoretical Computer Science*, 3(4):263–398, 2009.

[105] T. Lee and S. Zhang. Composition theorems in communication complexity. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming, (ICALP)*, pages 475–489, 2010. doi: 10.1007/978-3-642-14165-2_41. URL https://doi.org/10.1007/978-3-642-14165-2_41.

[106] T. Lee, A. Shraibman, and R. Špalek. A direct product theorem for discrepancy. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC)*, pages 71–80, 2008. doi: 10.1109/CCC.2008.25. URL https://doi.org/10.1109/CCC.2008.25.

[107] L. A. Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9 (3):115–116, 1973.

[108] N. Linial and A. Shraibman. Learning complexity vs communication complexity. *Combinatorics, Probability and Computing*, 18(1-2):227–245, 2009. doi: 10.1017/S0963548308009656. URL https://doi.org/10.1017/S0963548308009656.

[109] Y. Manin. Vychislimoe i nevychislimoe (computable and noncomputable). *Soviet Radio*, pages 13–15, 1980. In Russian.

[110] Y. Manin. Classical computing, quantum computing, and Shor's factoring algorithm. quant-ph/9903008, 2 Mar 1999.

[111] J. Matousek. *Geometric Discrepancy: An Illustrated Guide*, volume 18. Springer Science & Business Media, 2009.

[112] J. Matousek. *Lectures on Discrete Geometry*, volume 212. Springer Science & Business Media, 2013.

[113] A. McGregor. Graph Stream Algorithms: A Survey. *SIGMOD Rec.*, 43(1):9–20, 2014.

[114] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On Data Structures and Asymmetric Communication Complexity. *Journal of Computer and System Sciences*, 57(1):37–49, 1998.

[115] A. Montanaro, H. Nishimura, and R. Raymond. Unbounded-error quantum query complexity. *Theor. Comput. Sci.*, 412(35):4619–4628, 2011.

[116] E. Mossel, R. O'Donnell, and R. Servedio. Learning functions of k relevant variables. *Journal of Computer and System Sciences*, 69(3):421–434, 2004. Earlier version in STOC'03.

[117] S. Mukhopadhyay and D. Nanongkai. Weighted Min-Cut: Sequential, Cut-Query, and Streaming Algorithms. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 496–509, 2020.

[118] I. Newman. Private vs. Common Random Bits in Communication Complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[119] I. Newman. Private vs. common random bits in communication complexity. *Information Processing Letters*, 39(2):67–71, 1991.

[120] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[121] R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[122] J. Pach and P. K. Agarwal. *Combinatorial Geometry*, volume 37. John Wiley & Sons, 2011.

[123] R. Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 468–474, 1992. doi: 10.1145/129712.129758. URL https://doi.org/10.1145/129712.129758.

[124] R. Paturi and J. Simon. Probabilistic communication complexity. *Journal of Computer and System Sciences*, 33(1):106–123, 1986.

[125] A. Rao and A. Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[126] A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya of the Russian Academy of Sciences, mathematics*, 67(1):159–176, 2003. quant-ph/0204025.

[127] A. A. Razborov. On the Distributional Complexity of Disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992.

[128] T. Roughgarden. Communication Complexity (for Algorithm Designers). *Foundations and Trends in Theoretical Computer Science*, 11(3-4):217–404, 2016.

[129] A. Rubinstein, T. Schramm, and S. M. Weinberg. Computing Exact Minimum Cuts Without Knowing the Graph. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference, ITCS*, pages 39:1–39:16, 2018.

[130] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[131] M. Saglam and G. Tardos. On the Communication Complexity of Sparse Set Disjointness and Exists-Equal Problems. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 678–687, 2013.

[132] T. Sanders. Additive structures in sumsets. *Mathematical Proceedings of the Cambridge Philosophical Society*, 144(2):289–316, 2008.

[133] S. Sanyal. Near-optimal upper bound on Fourier dimension of Boolean functions in terms of Fourier sparsity. In *Proceedings of 42nd ICALP*, pages 1035–1045, 2015.

[134] N. Sauer. On the Density of Families of Sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[135] M. Schuld, I. Sinayskiy, and F. Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015. arXiv:1409.3097.

[136] R. Servedio and S. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004. Combines earlier papers from ICALP'01 and CCC'01. quant-ph/0007036.

[137] S. Shelah. A Combinatorial Problem, Stability and Order for Models and Theories in Infinitary Languages. *Pacific Journal of Mathematics*, 41:247–261, 1972.

[138] A. A. Sherstov. Algorithmic polynomials. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 311–324, 2018.

[139] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. Earlier version in FOCS'94. quant-ph/9508027.

[140] Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997.

[141] M. Sinha and R. de Wolf. Exponential separation between quantum communication and logarithm of approximate rank. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 966–981, 2019.

[142] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. In *Proceedings of 32nd ICALP*, volume 3580 of *Lecture Notes in Computer Science*, pages 1299–1311, 2005. quant-ph/0409116.

[143] H. Y. Tsang, C. H. Wong, N. X., and S. Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 658–667, 2013. URL https://doi.org/10.1109/FOCS.2013.76.

[144] A. M. Turing. On computable numbers, with an application to the entscheidungsproblem. a correction. *Proceedings of the London Mathematical Society*, 2(1):544–546, 1938.

[145] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134—-1142, 1984.

[146] V. N. Vapnik and A. Y. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[147] K. A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of 3rd Annual Workshop on Computational Learning Theory (COLT'90)*, pages 314–326, 1990.

[148] Wikipedia contributors. Timeline of computing hardware before 1950 — Wikipedia, the free encyclopedia, 2021. URL https://en.wikipedia.org/w/index.php?title=Timeline_of_computing_hardware_before_1950&oldid=1039367094. [Online; accessed 12-September-2021].

[149] Wikipedia contributors. Abacus — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Abacus&oldid=1043706124, 2021. [Online; accessed 12-September-2021].

[150] P. Wittek. *Quantum Machine Learning: What Quantum Computing Means to Data Mining*. Elsevier, 2014.

[151] R. de Wolf. Quantum communication and complexity. *Theoretical Computer Science*, 287(1):337–353, 2002.

[152] R. de Wolf. A brief introduction to Fourier analysis on the Boolean cube. *Theory of Computing*, 2008. ToC Library, Graduate Surveys 1.

[153] A. C-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of 18th IEEE FOCS*, pages 222–227, 1977.

[154] A. C-C. Yao. Some complexity questions related to distributive computing (preliminary report). In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

[155] A. C-C. Yao. Quantum circuit complexity. In *Proceedings of 1993 IEEE 34th Annual Foundations of Computer Science (FOCS)*, pages 352–361. IEEE, 1993.

[156] S. Zhang. On the tightness of the Buhrman-Cleve-Wigderson simulation. In *Algorithms and Computation, 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, pages 434–440, 2009.