

# Quantum Learning Boolean Linear Functions w.r.t. Product Distributions

Matthias C. Caro

Technische Universität München

caro@ma.tum.de

## Abstract

The problem of learning Boolean linear functions from quantum examples w.r.t. the uniform distribution can be solved on a quantum computer using the Bernstein-Vazirani algorithm [3]. A similar strategy can be applied in the case of noisy quantum training data, as was observed in [7]. We employ the biased quantum Fourier transform introduced in [8] to develop quantum algorithms for learning Boolean linear functions from quantum examples w.r.t. a biased product distribution. Here, one procedure is applicable to any (except full) bias, the other gives a better performance but is applicable only for small bias. Moreover, we discuss the stability of the second procedure w.r.t. noisy training data and w.r.t. faulty quantum gates. The latter also enables us to solve a version of the problem where the underlying distribution is not known in advance. Finally, we prove lower bounds on the classical and quantum sample complexities of the learning problem and compare these to the upper bounds implied by our algorithms.

**Keywords**— Computational learning theory, PAC learning, quantum computation, quantum Fourier transform, quantum learning

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Structure of the Paper . . . . .	4
<b>2</b>	<b>Preliminaries</b>	<b>4</b>
2.1	Basics of Quantum Information and Computation . . . . .	4
2.2	Basics of Learning Theory . . . . .	5
2.3	$\mu$ -biased Fourier Analysis of Boolean Functions . . . . .	5
2.4	$\mu$ -biased Quantum Fourier Sampling . . . . .	6
<b>3</b>	<b>The Learning Problem</b>	<b>7</b>
<b>4</b>	<b>A Generalized Bernstein-Vazirani Algorithm</b>	<b>8</b>
<b>5</b>	<b>Quantum Sample Complexity Upper Bounds</b>	<b>10</b>
5.1	Arbitrary Bias . . . . .	10
5.2	Small Bias . . . . .	12
<b>6</b>	<b>Stability w.r.t. Noise</b>	<b>15</b>
6.1	Noisy Training Data . . . . .	15
6.2	Faulty Quantum Gates . . . . .	20
6.3	The Case of Unknown Underlying Distributions . . . . .	21
<b>7</b>	<b>Sample Complexity Lower Bounds</b>	<b>25</b>
7.1	Classical Sample Complexity Lower Bounds . . . . .	25
7.2	Quantum Sample Complexity Lower Bounds . . . . .	27
<b>8</b>	<b>Conclusion and Outlook</b>	<b>31</b>
<b>9</b>	<b>Acknowledgements</b>	<b>32</b>
	<b>References</b>	<b>33</b>

# 1 Introduction

The origins of the fields of machine learning and quantum information and computation both lie in the 1980s. The arguably most influential learning model, namely the PAC model, was introduced by L. Valiant in 1984 [15] with which the problem of learning was given a rigorous mathematical framework. Around the same time, R. P. Feynman presented his idea of quantum computers [6] to the public and thus gave the starting signal for important innovations at the intersection of computer science, information theory and quantum theory. Both learning theory and quantum computation promise new realms of computation in which tasks that seem insurmountable from the perspective of classical computation become feasible. The first has already proved its practical worth and is indispensable for modern-world big data applications, the latter is not yet as practically relevant but much work is invested to make the promises of quantum computation a reality. The interested reader is referred to [14] and [12] for an introduction to statistical learning and quantum computation and information, respectively.

Considering the increasing importance of machine learning and quantum computation, attempting a merge of the two seems a natural step to take and the first step in this direction was taken already in [4]. The field of quantum learning has received growing attention over the last few years and by now some settings are known in which quantum training data and the ability to perform quantum computation can be advantageous for learning problems from an information-theoretic as well as from a computational perspective, in particular for learning problems with fixed underlying distribution (see e.g. [1] for an overview). It was, however, shown in [2] that no such information-theoretic advantage can be obtained in the (distribution-independent) quantum PAC model (based on [4]) compared to the classical PAC model (introduced in [15]).

One of the early examples of the aptness of quantum computation for learning problems is the task of learning Boolean linear functions w.r.t. the uniform distribution via the Bernstein-Vazirani algorithm presented in [3]. In particular, whereas this task of identifying an unknown  $n$ -bit string classically requires a number of examples growing (at least) linearly with  $n$ , a bound on the sufficient number of copies of the quantum example state independent of  $n$  can be established. This approach was taken up in [7] where it is shown that, essentially, the Bernstein-Vazirani-based learning method is also viable if the training data is noisy. However, also this analysis is restricted to quantum training data arising from the uniform distribution. The same limiting assumption was also made in [4] for learning Disjunctive Normal Forms and in this context an extension to product distributions was achieved in [8].

Hence, a next direction to go is building up on the reasoning of [8] to extend the applicability of quantum learning procedures for linear functions to more general distributions. The analysis hereby differs from the one for DNFs because no concentration results for the biased Fourier spectrum of a linear function are

available. Moreover, whereas many studies of specific quantum learning tasks focus on providing explicit learning procedures yielding a better performance than known classical algorithms, we complement our learning algorithms with lower bounds on the size of the training data for a comparison to the best classical procedure and for a discussion of optimality among possible quantum strategies.

## 1.1 Structure of the Paper

The paper is structured in the following way. In section 2 we introduce the well-known notions from classical learning, quantum computation and Boolean Fourier analysis required for our purposes as well as the prototypic learning algorithm which motivates our procedures. Section 3 consists of a description of the learning task to be considered. This is followed by a generalization of the Bernstein-Vazirani algorithm to product distributions in section 4. In the next section, this is used to develop two quantum algorithms for solving our problem. Section 6 then contains a stability analysis of the second of the two procedures w.r.t. noise in training data and computation. Finally, we conclude with some open questions and the references.

## 2 Preliminaries

### 2.1 Basics of Quantum Information and Computation

We first define some of the fundamental objects of quantum information theory, albeit restricted to those required in our discussion. For the purpose of our presentation, we will consider a pure  $n$ -qubit quantum state to be represented by a state vector  $|\psi\rangle \in \mathbb{C}^{2^n}$  (in Dirac notation). Such a state encodes measurement probabilities in the following way: If  $\{|b_i\rangle\}_{i=1}^{2^n}$  is an orthonormal basis of  $\mathbb{C}^{2^n}$ , then there corresponds a measurement to this basis and the probability of observing outcome  $i$  for a system in state  $|\psi\rangle$  is given by  $|\langle b_i|\psi\rangle|^2$ . Finally, when considering multiple subsystems we will denote the composite state by the tensor product, i.e. if the first system is in state  $|\psi\rangle$  and the second in state  $|\phi\rangle$ , the composite system is in state  $|\psi\rangle \otimes |\phi\rangle =: |\psi, \phi\rangle$ .

Quantum computation now consists in evolution of quantum states. Performing a computational step on an  $n$ -qubit state corresponds to applying an  $2^n \times 2^n$  unitary transformation to the current quantum state. (The most relevant example of such unitary gates in our context will be the (biased) quantum Fourier transform discussed in more detail in subsection 2.4.) As the outcome of a quantum computation is supposed to be classical, as final step of our computation we perform a measurement such that the final output will be a sample from the corresponding measurement statistics.

We will also use some standard notions from (quantum) information theory. For example, we denote the Shannon entropy of a random variable  $X$  by  $H(X)$ , the conditional entropy of a random variable  $X$  given

$Y$  as  $H(X|Y)$  and the mutual information between random variables  $X$  and  $Y$  as  $I(X : Y)$ . Similarly, the von Neumann entropy of a quantum state  $\rho$  will be denoted as  $S(\rho)$  and the mutual information for a bipartite quantum state  $\rho_{AB}$  as  $I(\rho_{AB}) = I(A : B)$ . Standard results on these quantities which will enter our discussion can e.g. be found in [12].

## 2.2 Basics of Learning Theory

Next we describe the well-known PAC model of learning, introduced by [15], and its extension to quantum examples, first discussed in [4]. In classical (supervised and proper) PAC learning for spaces  $\mathcal{X}$ ,  $\mathcal{Y}$  and a concept class  $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ , a learning algorithm receives as input labelled training data  $\{(x_i, f(x_i))\}_{i=1}^m$  for some  $f \in \mathcal{F}$ , where the  $x_i$  are drawn independently according to some probability distribution  $D$  on  $\mathcal{X}$  which is unknown to the learner. The goal of the learner is to approximate the unknown function  $f$  from such training examples with high success probability.

We can formalize this as follows: We call a concept class  $\mathcal{F}$  PAC-learnable if there exists a learning algorithm  $\mathcal{A}$  and a map  $m_{\mathcal{F}} : (0, 1) \times (0, 1) \rightarrow \mathbb{N}$  s.t. for every  $D \in \text{Prob}(\mathcal{X})$ ,  $f \in \mathcal{F}$  and  $\delta, \varepsilon \in (0, 1)$ , running  $\mathcal{A}$  on training data of size  $m \geq m_{\mathcal{F}}(\delta, \varepsilon)$  drawn according to  $D$  and  $f$  with probability  $\geq 1 - \delta$  (w.r.t. the choice of training data) yields a hypothesis  $h$  s.t.  $\mathbb{P}_{x \sim D}[h(x) \neq f(x)] \leq \varepsilon$ . The smallest possible choice for  $m_{\mathcal{F}}$  is called sample complexity of  $\mathcal{F}$ .

Note that this definition of PAC learning captures the information-theoretic challenge of the learning problem in the sample complexity, but it does not refer to the computational complexity of learning. The focus on sample complexity is typical in statistical learning theory. Hence, also our results will be formulated in terms of sample complexity bounds. As we give explicit algorithms, these results directly imply bounds on the computational complexity, however, we will not discuss them in any detail.

The quantum PAC model differs from the classical PAC model in the form of the training data and the allowed form of computation. Namely, in the quantum PAC model the training data consists of  $m$  copies of the quantum example state  $|\psi_f\rangle = \sum_{x \in \mathcal{X}} \sqrt{D(x)} |x, f(x)\rangle$ , and this training data is processed by quantum computational steps. With this small change the above definition of PAC learnability and sample complexity now carries over analogously.

We will deviate from the standard PAC setting in an important way, namely we simplify it by restricting the class of admissible underlying probability distributions. More precisely, we will either assume the underlying distribution to be known to the learner in advance or we will assume that the underlying distribution is chosen among a set of product distributions with bounded bias over the Boolean cube.

## 2.3 $\mu$ -biased Fourier Analysis of Boolean Functions

We now give the basic ingredients of  $\mu$ -biased Fourier analysis over the Boolean cube  $\{-1, 1\}^n$ . For more details, the reader is referred to [13].

For a bias vector  $\mu \in [-1, 1]^n$ , define the  $\mu$ -biased product distribution  $D_\mu$  on  $\{-1, 1\}^n$  via

$$D_\mu(x) := \left( \prod_{i: x_i=1} \frac{1+\mu_i}{2} \right) \left( \prod_{i: x_i=-1} \frac{1-\mu_i}{2} \right) = \prod_{1 \leq i \leq n} \frac{1+x_i \mu_i}{2}.$$

Thus, a positive entry  $\mu_i$  tells us that at the  $i^{\text{th}}$  position the distribution is biased towards  $+1$ , a negative entry  $\mu_i$  tells us that at the  $i^{\text{th}}$  position the distribution is biased towards  $-1$ . The absolute value of  $\mu_i$  quantifies the strength of the bias in the  $i^{\text{th}}$  component. We call  $D_\mu$   $c$ -bounded, for  $c \in (0, 1]$ , if  $\mu \in [-1+c, 1-c]^n$ . Assuming the underlying product distribution to be  $c$ -bounded thus corresponds to assuming that the bias is not arbitrarily strong.

For Fourier analysis we now need an orthonormal basis for the function space  $\{-1, 1\}^{\{-1, 1\}^n}$  w.r.t. the inner product  $\langle \cdot, \cdot \rangle_\mu$  defined by  $\langle f, g \rangle_\mu = \mathbb{E}_{D_\mu}[fg]$ . One can show (e.g. by induction on  $n$ ) that such an orthonormal basis is given by  $\{\phi_{\mu, j}\}_{j \in \{0, 1\}^n}$  with  $\phi_{\mu, j}(x) = \prod_{i: j_i=1} \frac{x_i - \mu_i}{\sqrt{1 - \mu_i^2}}$ . For a function  $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$  this now gives a representation  $f(x) = \sum_{j \in \{0, 1\}^n} \hat{f}_\mu(j) \phi_{\mu, j}(x)$  with  $\hat{f}_\mu(j) := \langle f, \phi_{\mu, j} \rangle_\mu$ .

## 2.4 $\mu$ -biased Quantum Fourier Sampling

We now turn to the description of the quantum algorithm for  $\mu$ -biased quantum Fourier sampling which constitutes the basic ingredient of our learning algorithms and which, to our knowledge, was first presented in [8]. There the authors demonstrate that the  $\mu$ -biased Fourier transform for a  $c$ -bounded  $D_\mu$  with  $c \in (0, 1]$  can be implemented on a quantum computer as the  $n$ -qubit  $\mu$ -biased quantum Fourier transform

$$H_\mu^n |x\rangle = H_\mu \otimes \dots \otimes H_\mu |x_1, \dots, x_n\rangle = \sum_{j \in \{0, 1\}^n} \sqrt{D_\mu(x)} \phi_{\mu, j}(x) |j\rangle, \quad x \in \{-1, 1\}^n.$$

In the same way as the unbiased quantum Fourier transform can be used for quantum Fourier sampling, this  $\mu$ -biased version now yield a procedure to sample from the  $\mu$ -biased Fourier spectrum of a function using a quantum computer. We describe the corresponding procedure in Algorithm 1.

---

### Algorithm 1 $\mu$ -biased Quantum Fourier Sampling

---

**Input:**  $|\psi_f\rangle = \sum_{x \in \{-1, 1\}^n} \sqrt{D_\mu(x)} |x, f(x)\rangle$  for a function  $f : \{-1, 1\}^n \rightarrow \{0, 1\}$

**Output:**  $j \in \{0, 1\}^n$  with probability  $(\hat{g}_\mu(j))^2$ , where the function  $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$  is defined as  $g(x) = (-1)^{f(x)}$ .

**Success Probability:**  $\frac{1}{2}$

- 1: Perform the  $\mu$ -biased QFT  $H_\mu$  on the first  $n$  qubits, obtain the state  $(H_\mu \otimes \mathbb{1})|\psi_f\rangle$ .
  - 2: Perform a Hadamard gate on the last qubit, obtain the state  $(H_\mu \otimes H)|\psi_f\rangle$ .
  - 3: Measure each qubit in the computational basis and observe outcome  $j = j_1 \dots j_{n+1}$ .
  - 4: **if**  $j_{n+1} = 0$  **then** ▷ This corresponds to a failure of the sampling algorithm.
  - 5:     Output  $o \leftarrow \perp$  and end computation.
  - 6: **else if**  $j_{n+1} = 1$  **then** ▷ This corresponds to a success of the sampling algorithm.
  - 7:     Output  $o \leftarrow j_1 \dots j_n$  and end computation.
  - 8: **end if**
-

That this algorithm works as claimed follows by analysing the transformation of the quantum state throughout the algorithm and making use of the orthonormality of the basis. This is the content of the following

**Lemma 2.1.** (Lemma 3 in [8])

*With probability  $\frac{(\hat{g}_\mu(j))^2}{2}$ , where  $g : \{-1, 1\}^n \rightarrow \{-1, 1\}$ ,  $g(x) = (-1)^{f(x)}$ , Algorithm 1 outputs the string  $j \in \{0, 1\}^n$ .*

This result allows us to generalize results based on quantum Fourier sampling w.r.t. the uniform distribution. In particular, we will apply it to obtain a generalization of the Bernstein-Vazirani algorithm.

### 3 The Learning Problem

We now describe the learning task which this contribution aims to understand. For  $a \in \{0, 1\}^n$ , define

$$f^{(a)} : \{-1, 1\}^n \rightarrow \{0, 1\}, \quad f^{(a)}(x) := \sum_{i=1}^n a_i \frac{1 - x_i}{2} \pmod{2}.$$

When we observe that  $\frac{1-x_i}{2}$  is simply the bit-description of  $x_i$ , it becomes clear that  $f^{(a)}$  computes the parity of the entries of the bit-description of  $x_i$  at the positions at which  $a$  has a 1-entry.

The classical task which inspires our problem is the following: Given a set of  $m$  labelled examples  $S = \{(x_i, f^{(a)}(x_i))\}_{i=1}^m$ , where the  $x_i$  are drawn i.i.d. according to  $D_\mu$ , determine the string  $a$  with high success probability. Here, we assume prior knowledge of the underlying distribution and that the underlying distribution is  $c$ -bounded.

This learning model is different from the PAC model introduced by L. Valiant [15] in two ways. On the one hand, we are not looking for an approximate hypothesis but for the output of the exact target concept. On the other hand, in the standard PAC model the underlying distribution is unknown to the learner. The first of these two differences is a stronger restriction on the success criteria of our learner, the second is a relaxation that will be addressed shortly in section 5.

Classically, this problem is, in some sense, hard. As we show in subsection 7.1, successfully solving the task requires a number of examples that grows at least linearly in  $n$ . If we consider a version of this problem with noisy training data, then known classical algorithms perform worse both w.r.t. sample complexity as well as running time. (E.g. [11] exhibits an algorithm with polynomial (superlinear) sample complexity but barely subexponential runtime.)

The step to the quantum version of this problem now is the same as from classical to quantum PAC learning. This means that training data is given as  $m$  copies of the quantum example state

$$|\psi_a\rangle = \sum_{x \in \{-1, 1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle \text{ and the learner is allowed to use quantum computation to process}$$

the training data. The goal of the quantum learner remains that of outputting the unknown string  $a$  with high success probability.

## 4 A Generalized Bernstein-Vazirani Algorithm

To understand how  $\mu$ -biased quantum Fourier sampling can help us with our learning problem we first compute the  $\mu$ -biased Fourier coefficients of  $g^{(a)} := (-1)^{f^{(a)}}$ , with  $f^{(a)}, a \in \{0, 1\}^n$  the linear functions defined in section 3.

**Lemma 4.1.** *Let  $a \in \{0, 1\}^n$ ,  $g^{(a)} := (-1)^{f^{(a)}}$  and  $\mu \in (-1, 1)^n$ . Then the  $\mu$ -biased Fourier coefficients of  $g^{(a)}$  satisfy*

(i) *If  $\exists 1 \leq i \leq n$  s.t.  $a_i = 0 \neq j_i$ , then  $\hat{g}_\mu^{(a)}(j) = 0$ .*

(ii) *If  $\forall 1 \leq i \leq n$  s.t.  $a_i = 0$  also  $j_i = 0$ , then*

$$\hat{g}_\mu^{(a)}(j) = \left( \prod_{l: a_l=1 \neq j_l} \mu_l \right) \left( \prod_{l: a_l=1=j_l} \sqrt{1 - \mu_l^2} \right).$$

We can reformulate this as

$$\hat{g}_\mu^{(a)}(j) = \left( \prod_{l: a_l=0} (1 - j_l) \right) \left( \prod_{l: a_l=1} \left( (1 - j_l)\mu_l + j_l \sqrt{1 - \mu_l^2} \right) \right).$$

**Proof:** We do a proof by induction on  $n \in \mathbb{N}$ .

$n = 1$ : In this case we have  $f^{(a)}(x) = a\tilde{x}$ ,  $g^{(a)}(x) = (-1)^{a\tilde{x}}$  for  $\tilde{x} = \frac{1-x}{2}$ ,  $\phi_{\mu,0}(x) = 1$ , and  $\phi_{\mu,1}(x) = \frac{x-\mu}{\sqrt{1-\mu^2}}$ . (We leave out unnecessary indices to improve readability.) We compute

$$\hat{g}_\mu^{(a)}(j) = \mathbb{E}_{D_\mu} [(-1)^{a\tilde{x}} \phi_{\mu,j}(x)] = \frac{1+\mu}{2} \cdot 1 \cdot \phi_{\mu,j}(1) + \frac{1-\mu}{2} \cdot (-1)^a \cdot \phi_{\mu,j}(-1).$$

By plugging in we now obtain

$$\hat{g}_\mu^{(0)}(0) = 1, \quad \hat{g}_\mu^{(0)}(1) = 0, \quad \hat{g}_\mu^{(1)}(0) = \mu, \quad \hat{g}_\mu^{(1)}(1) = \sqrt{1 - \mu^2},$$

which is exactly the claim for  $n = 1$ .

$n \rightarrow n + 1$ : Suppose the claim is true for some  $n \in \mathbb{N}$ . Observe that our "objects of interest" all factorise:

$D_\mu$  and  $\phi_{\mu,j}$  factorise by definition and

$$g^{(a)}(x) = (-1)^{\sum_{i=1}^{n+1} a_i \tilde{x}_i} = (-1)^{\sum_{i=1}^n a_i \tilde{x}_i} \cdot (-1)^{a_{n+1} \tilde{x}_{n+1}}.$$



Hence, we can write for  $j = j_1 \dots j_{n+1} \in \{0, 1\}^{n+1}$

$$\begin{aligned}\hat{g}_\mu^{(a)}(j) &= \mathbb{E}_{D_\mu}[\phi_{\mu,j}(x)g^{(a)}(x)] \\ &= \mathbb{E}_{D_{\mu_{n+1}}}[\phi_{\mu_{n+1},j_{n+1}}(x_{n+1}) \cdot (-1)^{a_{n+1} \cdot \tilde{x}_{n+1}}] \cdot \mathbb{E}_{D_{\mu_1, \dots, \mu_n}}[\phi_{\mu, j_1 \dots j_n}(x_1 \dots x_n)g^{(a)}(x_1 \dots x_n)].\end{aligned}$$

For the first factor we reuse our computation for  $n = 1$ , for the second factor we use the induction assumption. Combining these now directly yields the claim.  $\square$

**Remark 4.2.** The above proof in particular shows that

$$\hat{g}_\mu^{(a_1 \dots a_n)}(j_1 \dots j_n) = \prod_{i=1}^n \mathbb{E}_{D_{\mu_i}}[\phi_{\mu_i, j_i}(x_i) \cdot (-1)^{a_i \cdot \tilde{x}_i}],$$

i.e. the  $\mu$ -biased Fourier coefficients factorise.

For clarity, we write down explicitly the algorithm which we obtain as a generalisation of the Bernstein-Vazirani algorithm to a  $\mu$ -biased product distribution as Algorithm 2.

---

**Algorithm 2** Generalised Bernstein-Vazirani algorithm

---

**Input:**  $|\psi_a\rangle = \sum_{x \in \{-1, 1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$  for  $a \in \{0, 1\}^n$ , and  $\mu \in [-1, 1]^n$

**Output:**  $c \in \{0, 1\}^n$  with probability  $\left( \prod_{l: a_l=0} (1 - c_l) \right) \left( \prod_{l: a_l=1} ((1 - c_l)\mu_l^2 + c_l(1 - \mu_l^2)) \right)$

**Success Probability:**  $\frac{1}{2}$

- 1: Perform the  $\mu$ -biased QFT  $H_\mu$  on the first  $n$  qubits, obtain the state  $(H_\mu \otimes \mathbb{1})|\psi_a\rangle$ .
  - 2: Perform a Hadamard gate on the last qubit, obtain the state  $(H_\mu \otimes H)|\psi_a\rangle$ .
  - 3: Measure each qubit in the computational basis and observe outcome  $j = j_1 \dots j_{n+1}$ .
  - 4: **if**  $j_{n+1} = 0$  **then**  $\triangleright$  This corresponds to a failure of the algorithm.
  - 5:     Output  $o = \perp$ .
  - 6: **else if**  $j_{n+1} = 1$  **then**  $\triangleright$  This corresponds to a success of the algorithm.
  - 7:     Output  $o = j_1 \dots j_n$ .
  - 8: **end if**
- 

We now show that the output probabilities of Algorithm 2 are as claimed in its description. This follows directly by combining Lemma 2.1 on the workings of  $\mu$ -biased quantum Fourier sampling with Lemma 4.1 on the  $\mu$ -biased Fourier coefficients of our target functions and is the content of the following

**Theorem 4.3.** *Let  $|\psi_a\rangle = \sum_{x \in \{-1, 1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$  be a quantum example state,  $a \in \{0, 1\}^n$  and  $\mu \in (-1, 1)^n$ . Then Algorithm 2 provides an outcome  $|j_1 \dots j_{n+1}\rangle$  with the following properties:*

- (i)  $\mathbb{P}[j_{n+1} = 0] = \frac{1}{2} = \mathbb{P}[j_{n+1} = 1]$ ,
- (ii)  $\mathbb{P}[j_1 \dots j_n = a | j_{n+1} = 1] = \prod_{l: a_l=1} (1 - \mu_l^2)$ ,

- (iii) for  $c \neq a$ :  $\mathbb{P}[j_1 \dots j_n = c | j_{n+1} = 1] = \left( \prod_{l:a_l=0} (1 - c_l) \right) \left( \prod_{l:a_l=1} ((1 - c_l)\mu_l^2 + c_l(1 - \mu_l^2)) \right),$
- (iv)  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq j_i | j_{n+1} = 1] = 0$ , and
- (v)  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq \sum_{i=1}^n \mu_i^2$ . In particular, if  $D_\mu$  is  $c$ -bounded, then  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq n(1 - c)^2$ .

Note that (v) can be trivial if the bias is too strong. This observation already hints at why we use different procedures for arbitrary and for small bias.

## 5 Quantum Sample Complexity Upper Bounds

This section contains the description of two procedures for solving the task of learning an unknown Boolean linear function from quantum examples w.r.t. a product distribution. (Here, we assume perfect quantum examples, noisy examples will be taken into consideration in the next section.) It is subdivided into an approach which is applicable for arbitrary (not full) bias in the product distribution and a strategy which produces better results but is only valid for small bias.

### 5.1 Arbitrary Bias

As in the case of learning w.r.t. the uniform distribution we intend to run the generalised Bernstein-Vazirani algorithm multiple times as subroutine and then use our knowledge of the outcome of the subroutine together with probability-theoretic arguments. The main difficulty compared to the case of an example state arising from the uniform distribution lies in the fact that, whereas an observation of  $j_{n+1} = 1$  when performing the standard Bernstein-Vazirani algorithm guarantees that  $j_1 \dots j_{n+1}$  equals the desired string, this is not true in the  $\mu$ -biased case. Hence, we have to develop a different procedure of learning from the outcomes of the subroutines. For this purpose we propose Algorithm 3.

We now show that the number of copies postulated in Algorithm 3 is actually sufficient to achieve the desired success probability.

**Theorem 5.1.** *Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for some  $c \in (0,1]$ . Then  $\mathcal{O}\left(\left[\left(2 \ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1} (\ln(n) + \ln(\frac{2}{\delta}))\right]\right)$  copies of the quantum example state  $|\psi_a\rangle$  are sufficient to guarantee that with probability  $\geq 1 - \delta$  Algorithm 3 outputs the string  $a$ .*

**Proof:** We want to show that  $\mathbb{P}[\text{Algorithm 3 does not output } a] \leq \delta$ . We do so by distinguishing cases in which Algorithm 3 does not output  $a$ . The first such case occurs if  $o = \perp$ . The second such case would be that  $\exists 1 \leq i \leq n : a_i = 0 \neq o_i$ , but due to Theorem 4.3, this is an event of probability 0. The third

---

**Algorithm 3** Amplified Generalised Bernstein-Vazirani algorithm - Version 1

---

**Input:**  $m$  copies of  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$  for  $a \in \{0,1\}^n$ , where the number of copies is  $m \geq \Omega\left(\left\lceil \left(2 \ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1} (\ln(n) + \ln(\frac{2}{\delta})) \right\rceil\right)$ , and  $\mu \in (-1,1)^n$  and  $c \in (0,1]$  s.t.  $D_\mu$  is  $c$ -bounded

**Output:**  $a \in \{0,1\}^n$

**Success Probability:**  $\geq 1 - \delta$

- 1: **for**  $1 \leq l \leq m$  **do**
- 2:     Run Algorithm 2 on the  $l^{th}$  copy of  $|\psi_a\rangle$ , store the output as  $o^{(l)}$ .
- 3: **end for**
- 4: **if**  $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$  **then**
- 5:     **for**  $1 \leq i \leq n$  **do**
- 6:         Let  $o_i := \max_{l: o^{(l)} \neq \perp} o_i^{(l)}$ .
- 7:     **end for**
- 8:     Output  $o = o_1 \dots o_n$ .
- 9: **else if**  $\forall 1 \leq l \leq m : o^{(l)} = \perp$  **then**
- 10:     Output  $o = \perp$ .
- 11: **end if**

---

and last such case is  $\exists 1 \leq i \leq n : a_i = 1 \neq o_i$ . Hence, we get

$$\mathbb{P}[\text{Algorithm 3 does not output } a] = \mathbb{P}[\text{Algorithm 3 outputs } \perp] + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i]. \quad (5.1)$$

First, we bound the probability of the algorithm outputting  $\perp$  (i.e. each subroutine failing) as follows:

$$\mathbb{P}[\text{Algorithm 3 outputs } \perp] = \mathbb{P}[\forall 1 \leq l \leq m : \text{Algorithm 2 applied to } |\psi_a\rangle \text{ outputs } \perp] \stackrel{\text{Theorem 4.3}}{=} \left(\frac{1}{2}\right)^m.$$

The choice of  $m$  now guarantees that this last term is  $\leq \frac{\delta}{2}$  (if we choose the constant hidden in the  $\Omega(\dots)$ -notation sufficiently large).

Now we bound the second term in equation (5.1). We make the following observation: Suppose  $1 \leq i \leq n$  is s.t.  $a_i = 1$ . As the Fourier coefficients, and with them the output probabilities, factorise, the probability of Algorithm 2 outputting a string  $j_1 \dots j_n$  with  $j_i = 1 = a_i$  is simply the probability of Algorithm 2 applied to only the subsystem state of  $|\psi_a\rangle$  corresponding to the  $i^{th}$  and the  $(n+1)^{st}$  subsystem outputting a 1. By Theorem 4.3, this probability is

$$\mathbb{P}[j_i = 1] = \mathbb{P}[j_{n+1} = 1] \cdot \mathbb{P}[j_i = 1 | j_{n+1} = 1] = \frac{1}{2} \cdot (1 - \mu_i^2).$$

Hence, assuming  $a_i = 1$ , the probability of not observing a 1 at the  $i^{th}$  position in any of the  $m$  runs of

Algorithm 2 is  $(1 - \frac{1}{2} \cdot (1 - \mu_i^2))^m = (\frac{1}{2}(1 + \mu_i^2))^m$ . By  $c$ -boundedness we get

$$\left(\frac{1}{2}(1 + \mu_i^2)\right)^m \leq \left(\frac{1}{2} + \frac{1}{2}(1 - c)^2\right)^m = \left(1 - c + \frac{c^2}{2}\right)^m.$$

So using the union bound we arrive at

$$\begin{aligned} \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i] &= \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \text{ and in } m \text{ runs no 1 is observed at the } i^{th} \text{ entry}] \\ &\leq \sum_{i=1}^n \mathbb{P}[a_i = 1 \text{ and in } m \text{ runs no 1 is observed at the } i^{th} \text{ entry}] \\ &\leq n \cdot \left(1 - c + \frac{c^2}{2}\right)^m. \end{aligned}$$

The choice of  $m$  guarantees that this last term is  $\leq \frac{\delta}{2}$  (if we choose the constant hidden in the  $\Omega(\dots)$ -notation sufficiently large).

We now combine this with equation (5.1) and obtain

$$\mathbb{P}[\text{Algorithm 3 does not output } a] \leq \frac{\delta}{2} + \frac{\delta}{2} = \delta,$$

which finishes the proof.  $\square$

**Remark 5.2.** We want to comment shortly on the dependence of the sample complexity bound on the  $c$ -boundedness constant by considering extreme cases. As  $c \rightarrow 0$ , i.e. we allow more and more strongly biased distributions, we get  $m \rightarrow \infty$ , which reflects the fact that in the case of a fully biased underlying product distribution, only a single bit of information about  $a$  can be extracted, so exactly learning the string  $a$  is (in general) not possible.

For  $c = 1$ , i.e. the case of no bias, we simply obtain that  $\mathcal{O}(\lceil (\ln(n) + \ln(\frac{2}{\delta})) \rceil)$  copies of the quantum example state are sufficient. Note that this does not coincide with the bound obtained for the standard Bernstein-Vazirani procedure which is independent of  $n$  (see e.g. [7]). This discrepancy is due to the difference in "amplification procedures". Namely, in Algorithm 3 we do not explicitly make use of the knowledge that, given  $j_{n+1} = 1$ , we know the probability of  $j_1 \dots j_n = a_1 \dots a_n$  because, whereas for  $\mu = 0$  this probability equals 1, for  $\mu \neq 0$  it can become small. Hence, for  $\mu \neq 0$  our algorithm introduces an additional procedure to deal with the uncertainty of  $j_1 \dots j_n$  even knowing  $j_{n+1}$  and we see in the proof that this yields the additional  $\ln(n)$  term. In the next subsection we describe a way to get rid of the  $\ln(n)$  term for "small" bias.

## 5.2 Small Bias

In this subsection we want to study the case in which (v) of Theorem 4.3 gives a good bound. Namely, throughout this subsection we will assume that the  $c$ -boundedness constant is s.t.  $n(1 - c)^2 < \frac{1}{2}$  or,

equivalently,  $c > 1 - \frac{1}{\sqrt{2n}}$ . This assumption will allow us to apply a different procedure to learn from the output of Algorithm 2 and thus obtain a different bound on the sample complexity of the problem. Note, however, that this requirement becomes more restrictive with growing  $n$  and can in the limit  $n \rightarrow \infty$  only be satisfied by  $c = 1$ , i.e. for the underlying distributions being uniform.

Our procedure for the case of small bias is given in Algorithm 4.

---

**Algorithm 4** Amplified Generalised Bernstein-Vazirani algorithm - Version 2

---

**Input:**  $m$  copies of  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$  for  $a \in \{0,1\}^n$ , where the number of copies is  $m \geq \Omega\left(\frac{4}{(1-2n(1-c)^2)^2} \ln\left(\frac{2}{\delta}\right)\right)$ , as well as  $\mu \in [-1,1]^n$  and  $c \in (0,1]$  s.t.  $D_\mu$  is  $c$ -bounded

**Output:**  $a \in \{0,1\}^n$

**Success Probability:**  $\geq 1 - \delta$

- 1: **for**  $1 \leq l \leq m$  **do**
- 2:     Run Algorithm 2 on the  $l^{\text{th}}$  copy of  $|\psi_a\rangle$ , store the output as  $o^{(l)}$ .
- 3: **end for**
- 4: **if**  $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$  **then**
- 5:     **for**  $1 \leq i \leq n$  **do**
- 6:         Let  $o_i = \arg \max_{r \in \{0,1\}} |\{1 \leq l \leq m : o_i^{(l)} = r\}|$ .
- 7:     **end for**
- 8:     Output  $o = o_1 \dots o_n$ .
- 9: **else if**  $\forall 1 \leq l \leq m : o^{(l)} = \perp$  **then**
- 10:     Output  $o = \perp$ .
- 11: **end if**

---

**Theorem 5.3.** Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for some  $c \in (0,1]$  satisfying  $c > 1 - \frac{1}{\sqrt{2n}}$ . Then  $\mathcal{O}\left(\frac{4}{(1-2n(1-c)^2)^2} \ln\left(\frac{1}{\delta}\right)\right)$  copies of the quantum example state  $|\psi_a\rangle$  are sufficient to guarantee that with probability  $\geq 1 - \delta$  Algorithm 4 outputs the string  $a$ .

**Proof:** By Theorem 4.3, we have  $\mathbb{P}[j_{n+1} = 1] = \frac{1}{2}$ . Hence, the probability of observing  $j_{n+1} = 1$  in at most  $k-1$  of the  $m$  runs of Algorithm 2 is given by  $\sum_{l=0}^{k-1} \binom{m}{l} \left(\frac{1}{2}\right)^l \left(\frac{1}{2}\right)^{m-l} = \mathbb{P}[\text{Binom}(m, \frac{1}{2}) \geq m-k]$ . Next we assume  $k \leq \frac{m}{2}$  (this will be justified later in the proof) and use Hoeffding's inequality (compare e.g. Theorem 2.2.6 in [16]) to obtain

$$\mathbb{P}\left[\text{Bin}(m, \frac{1}{2}) \geq m-k\right] = \mathbb{P}\left[\text{Bin}(m, \frac{1}{2}) - \frac{m}{2} \geq m-k - \frac{m}{2}\right] \leq \exp\left(-\frac{2\left(\frac{m}{2} - k\right)^2}{m}\right). \quad (5.2)$$

We will now search for the number of observations of  $j_{n+1} = 1$  which is required to guarantee that the majority string is correct with high probability. Assume that we observe  $j_{n+1} = 1$  in  $k$  runs of Algorithm 2,  $k \in 2\mathbb{N}$ . (The latter assumption clearly does not significantly change the number of copies.). Using (v)

from Theorem 4.3 we see that

$$\begin{aligned}
\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] &\leq \mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq o_i] + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i] \\
&\leq 0 + \sum_{l=\lceil \frac{k}{2} \rceil}^k \binom{k}{l} \cdot (1 - n(1-c)^2)^{k-l} \cdot (n(1-c)^2)^l \\
&= \mathbb{P} \left[ \text{Bin}(k, n(1-c)^2) \geq \frac{k}{2} \right],
\end{aligned}$$

where the second inequality uses that the majority string can only be wrong if in at least half of the runs where we observed  $j_{n+1} = 1$  there was some error in the remaining string.

Next we use Hoeffding's inequality and obtain, using our assumption  $n(1-c)^2 < \frac{1}{2}$ , that

$$\begin{aligned}
\mathbb{P} \left[ \text{Bin}(k, n(1-c)^2) \geq \frac{k}{2} \right] &= \mathbb{P} \left[ \text{Bin}(k, n(1-c)^2) - kn(1-c)^2 \geq \frac{k}{2} - kn(1-c)^2 \right] \\
&\leq \exp \left( -k \frac{(1 - 2n(1-c)^2)^2}{2} \right).
\end{aligned}$$

We now set this last expression  $\leq \frac{\delta}{2}$  for  $\delta \in (0, 1)$  and rearrange the inequality to

$$k \geq \frac{2}{(1 - 2n(1-c)^2)^2} \ln \left( \frac{2}{\delta} \right). \quad (5.3)$$

Combining equations (5.3) and (5.2) we now require

$$\exp \left( -\frac{2 \left( \frac{m}{2} - \frac{2}{(1-2n(1-c)^2)^2} \ln \left( \frac{2}{\delta} \right) \right)^2}{m} \right) \stackrel{!}{\leq} \frac{\delta}{2}.$$

Rearranging gives

$$m^2 - 2m \left( \left( \frac{1 - 2n(1-c)^2}{2} \right)^{-2} - 1 \right) \ln \left( \frac{2}{\delta} \right) + \left( \frac{1 - 2n(1-c)^2}{2} \right)^{-4} \ln^2 \left( \frac{2}{\delta} \right) \geq 0.$$

By finding the zeros of this quadratic function we get to the sufficient sample size

$$\begin{aligned}
m &\geq \left( \left( \frac{1 - 2n(1-c)^2}{2} \right)^{-2} - 1 \right) \ln \left( \frac{2}{\delta} \right) \\
&\quad + \sqrt{\left( \left( \left( \frac{1 - 2n(1-c)^2}{2} \right)^{-2} - 1 \right) \ln \left( \frac{2}{\delta} \right) \right)^2 - \left( \frac{1 - 2n(1-c)^2}{2} \right)^{-4} \ln^2 \left( \frac{2}{\delta} \right)}.
\end{aligned}$$

This is in particular guaranteed if

$$m \geq \frac{4}{(1 - 2n(1-c)^2)^2} \ln \left( \frac{2}{\delta} \right).$$

Note that this lower bound in particular implies  $m \geq 2k$ , as required earlier in the proof. This proves the claim of the theorem thanks to the union bound.  $\square$

Morally speaking, Theorem 5.3 shows that for product distributions which are close enough to the uniform distribution the sample complexity upper bound is the same as for the unbiased case. We conjecture that there is an explicit noise threshold above which this sample complexity cannot be reached (see section 7), but have not yet succeeded in identifying that critical value.

## 6 Stability w.r.t. Noise

Both algorithms presented in the previous sections implicitly assume that the quantum example state perfectly represents the underlying function and that all quantum gates performed during the computation are perfectly accurate. In this section we relax these assumptions. We do so separately, but our analysis shows that moderate noise in the training data and moderately faulty quantum gates can be tolerated at the same time.

### 6.1 Noisy Training Data

We first analyse the performance of our algorithm in the case of noisy training data similarly to [7]. This means that we now assume our quantum example state to be of the form

$$|\psi_a^{\text{noisy}}\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x\rangle, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i,$$

where the  $\xi_{x_i}^i$  for  $1 \leq i \leq n$  and  $x_i \in \{-1,1\}$  are independent Bernoulli distributions according to parameters  $\eta^i$  (i.e.  $\mathbb{P}[\xi_{x_i}^i = 1] = \eta^i = 1 - \mathbb{P}[\xi_{x_i}^i = 0] \forall 1 \leq i \leq n$ ) and addition is understood modulo 2.

Here, we choose a noise model that is rather general but we make an important restriction. Namely, we do not allow a noise  $\xi_x$  that depends in an arbitrary way on  $x$  but rather we require the noise to have a specific sum structure  $\xi_x = \sum_{i=1}^n \xi_{x_i}^i$ . This requirement will later imply that also the noisy Fourier coefficients factorise. As this factorization is crucial for our analysis, with our strategy we cannot generalize the results of [7] on this more general noise model.

We first examine the result of applying the same procedure as in Algorithm 2 to a copy of a noisy quantum example state  $|\psi_a^{\text{noisy}}\rangle$ . To simplify referencing we write this down one more time as Algorithm 5 even though the procedure is exactly the same, only the form of the input changes.

Similarly to our previous analysis, we will first study the Fourier coefficients that are relevant for the sampling process in Algorithm 5.

---

**Algorithm 5** Generalised Bernstein-Vazirani algorithm with noisy training data

---

**Input:**  $|\psi_a^{\text{noisy}}\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x\rangle, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i$ , as well as  $\mu \in [-1, 1]$

**Output:** See Theorem 6.2

**Success Probability:**  $\frac{1}{2}$ .

- 1: Perform the  $\mu$ -biased QFT  $H_\mu$  on the first  $n$  qubits, obtain the state  $(H_\mu \otimes \mathbb{1})|\psi_a^{\text{noisy}}\rangle$ .
  - 2: Perform a Hadamard gate on the last qubit, obtain the state  $(H_\mu \otimes H)|\psi_a^{\text{noisy}}\rangle$ .
  - 3: Measure each qubit in the computational basis and observe outcome  $j = j_1 \dots j_{n+1}$ .
  - 4: **if**  $j_{n+1} = 0$  **then**  $\triangleright$  This corresponds to a failure of the algorithm.
  - 5:     Output  $o = \perp$ .
  - 6: **else if**  $j_{n+1} = 1$  **then**  $\triangleright$  This corresponds to a success of the algorithm.
  - 7:     Output  $o = j_1 \dots j_n$ .
  - 8: **end if**
- 

**Lemma 6.1.** Let  $a \in \{0, 1\}^n$ , let  $\xi_{x_i}^i$  for  $1 \leq i \leq n$  and  $x_i \in \{-1, 1\}$  be independent Bernoulli distributions, let  $g^{(a)}(x) := (-1)^{\sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i}$  and let  $\mu \in (-1, 1)$ . Then the  $\mu$ -biased Fourier coefficients of  $g^{(a)}$  satisfy: For  $y \in \{0, 1\}^n$ , with probability

$$\prod_{l=1}^n (y_l \cdot 2\eta^l(1 - \eta^l) + (1 - y_l) \cdot (1 - 2\eta^l(1 - \eta^l))),$$

it holds that

$$\begin{aligned} \hat{g}_\mu^{(a)}(j) &= \prod_{l: a_l=0} \left( y_l \cdot (-1)^{b_l} \left( (1 - j_l)\mu_l + j_l \sqrt{1 - \mu_l^2} \right) + (1 - y_l) \cdot (-1)^{b_l} (1 - j_l) \right) \\ &\quad \cdot \prod_{l: a_l=1} \left( y_l \cdot (-1)^{b_l} (1 - j_l) + (1 - y_l) \cdot (-1)^{b_l} \left( (1 - j_l)\mu_l + j_l \sqrt{1 - \mu_l^2} \right) \right). \end{aligned}$$

**Proof:** We do a proof by induction on  $n \in \mathbb{N}$ .

$n = 1$ : In this case we have  $f^{(a)}(x) = a\tilde{x}$ ,  $g^{(a)}(x) = (-1)^{a\tilde{x} + \xi_x}$  for  $\tilde{x} = \frac{1-x}{2}$ ,  $\phi_{\mu,0}(x) = 1$ , and  $\phi_{\mu,1}(x) = \frac{x-\mu}{\sqrt{1-\mu^2}}$ . (We leave out unnecessary indices to improve readability.) We compute

$$\hat{g}_\mu^{(a)}(j) = \mathbb{E}_{D_\mu} [(-1)^{a\tilde{x} + \xi_x} \phi_{\mu,j}(x)] = \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \phi_{\mu,j}(1) + \frac{1-\mu}{2} \cdot (-1)^{a+\xi_{-1}} \cdot \phi_{\mu,j}(-1).$$

By plugging in we now obtain

$$\begin{aligned} \hat{g}_\mu^{(0)}(0) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot 1 + \frac{1-\mu}{2} \cdot (-1)^{\xi_{-1}} \cdot 1, \\ \hat{g}_\mu^{(0)}(1) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \frac{1-\mu}{\sqrt{1-\mu^2}} + \frac{1-\mu}{2} \cdot (-1)^{\xi_{-1}} \cdot \frac{-1-\mu}{\sqrt{1-\mu^2}}, \\ \hat{g}_\mu^{(1)}(0) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot 1 + \frac{1-\mu}{2} \cdot (-1)^{1+\xi_{-1}} \cdot 1, \\ \hat{g}_\mu^{(1)}(1) &= \frac{1+\mu}{2} \cdot (-1)^{\xi_1} \cdot \frac{1-\mu}{\sqrt{1-\mu^2}} + \frac{1-\mu}{2} \cdot (-1)^{1+\xi_{-1}} \cdot \frac{-1-\mu}{\sqrt{1-\mu^2}}. \end{aligned}$$



So with probability  $(\eta^1)^2 + (1 - \eta^1)^2 = 1 - 2\eta^1(1 - \eta^1)$ , namely if  $\xi_1 = \xi_{-1} = b \in \{0, 1\}$ , we obtain

$$\hat{g}_\mu^{(0)}(0) = (-1)^b, \quad \hat{g}_\mu^{(0)}(1) = 0, \quad \hat{g}_\mu^{(1)}(0) = (-1)^b \mu, \quad \hat{g}_\mu^{(1)}(1) = (-1)^b \sqrt{1 - \mu^2},$$

and with probability  $2\eta^1(1 - \eta^1)$ , namely if  $\xi_1 = b \neq \xi_{-1}$ , we obtain

$$\hat{g}_\mu^{(0)}(0) = (-1)^b \mu, \quad \hat{g}_\mu^{(0)}(1) = (-1)^b \sqrt{1 - \mu^2}, \quad \hat{g}_\mu^{(1)}(0) = (-1)^b, \quad \hat{g}_\mu^{(1)}(1) = 0.$$

Therefore we obtain: With probability  $1 - 2\eta^1(1 - \eta^1)$  the  $\mu$ -biased Fourier coefficients satisfy

$$\hat{g}_\mu^{(a)}(j) = \begin{cases} (-1)^b(1 - j), & \text{for } a = 0 \\ (-1)^b((1 - j)\mu + j\sqrt{1 - \mu^2}) & \text{for } a = 1 \end{cases},$$

and with probability  $2\eta^1(1 - \eta^1)$  the  $\mu$ -biased Fourier coefficients satisfy

$$\hat{g}_\mu^{(a)}(j) = \begin{cases} (-1)^b((1 - j)\mu + j\sqrt{1 - \mu^2}) & \text{for } a = 0 \\ (-1)^b(1 - j), & \text{for } a = 1 \end{cases},$$

which is exactly the claim for  $n = 1$ .

$n \rightarrow n + 1$ : Suppose the claim is true for some  $n \in \mathbb{N}$ . Observe that our "objects of interest" all factorise:  $D_\mu$  and  $\phi_{\mu,j}$  factorise by definition and

$$g^{(a)}(x) = (-1)^{\sum_{i=1}^{n+1} a_i \tilde{x}_i + \xi_{x_i}^i} = (-1)^{\sum_{i=1}^n a_i \tilde{x}_i + \xi_{x_i}^i} \cdot (-1)^{a_{n+1} \tilde{x}_{n+1} + \xi_{x_{n+1}}^{n+1}}.$$

Hence, we can write for  $j = j_1 \dots j_{n+1} \in \{0, 1\}^{n+1}$

$$\begin{aligned} \hat{g}_\mu^{(a)}(j) &= \mathbb{E}_{D_\mu}[\phi_{\mu,j}(x)g^{(a)}(x)] \\ &= \mathbb{E}_{D_{\mu_{n+1}}}[\phi_{\mu_{n+1},j_{n+1}}(x) \cdot (-1)^{a_{n+1} \cdot \tilde{x}_{n+1} + \xi_{x_{n+1}}^{n+1}}] \cdot \mathbb{E}_{D_{\mu_1, \dots, \mu_n}}[\phi_{\mu, j_1 \dots j_n}(x_1 \dots x_n)g^{(a)}(x_1 \dots x_n)]. \end{aligned}$$

For the first factor we reuse our computation for  $n = 1$ , for the second factor we use the induction assumption. Then we obtain: For  $y \in \{0, 1\}$  with probability

$$\prod_{l=1}^n (y_l \cdot 2\eta^l(1 - \eta^l) + (1 - y_l) \cdot (1 - 2\eta^l(1 - \eta^l))),$$

namely when  $\begin{cases} \xi_1^l = b_l = \xi_{-1}^l & \text{if } y_l = 1 \\ \xi_1^l = b_l \neq \xi_{-1}^l & \text{if } y_l = 0 \end{cases}$ , the  $\mu$ -biased Fourier coefficients satisfy

$$\begin{aligned} \hat{g}_\mu^{(a)}(j) &= \prod_{l:a_l=0} \left( y_l \cdot (-1)^{b_l} \left( (1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2} \right) + (1-y_l) \cdot (-1)^{b_l} (1-j_l) \right) \\ &\cdot \prod_{l:a_l=1} \left( y_l \cdot (-1)^{b_l} (1-j_l) + (1-y_l) \cdot (-1)^{b_l} \left( (1-j_l)\mu_l + j_l\sqrt{1-\mu_l^2} \right) \right), \end{aligned}$$

as claimed.  $\square$

We now make a step analogous to the one from Lemma 4.1 to Theorem 4.3 in order to understand the output of Algorithm 5.

**Theorem 6.2.** *Let  $|\psi_a^{\text{noisy}}\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$  be a noisy quantum example state,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$ . Then Algorithm 5 provides an outcome  $|j_1 \dots j_{n+1}\rangle$  with the following properties:*

(i)  $\mathbb{P}[j_{n+1} = 0] = \frac{1}{2} = \mathbb{P}[j_{n+1} = 1]$ .

(ii) For any  $1 \leq i \leq n$ , with probability  $1 - 2\eta^i(1 - \eta^i)$  it holds that

$$\mathbb{P}[a_i = 0 \neq j_i | j_{n+1} = 1] = 0, \quad \mathbb{P}[a_i = 1 \neq j_i | j_{n+1} = 1] = \mu^2.$$

(iii) For any  $1 \leq i \leq n$ , with probability  $2\eta^i(1 - \eta^i)$  it holds that

$$\mathbb{P}[a_i = 0 \neq j_i | j_{n+1} = 1] = 1 - \mu^2, \quad \mathbb{P}[a_i = 1 \neq j_i | j_{n+1} = 1] = 1.$$

Building on this subroutine we will now describe an amplified procedure for moderate noise in Algorithm 6 analogous to the one described in subsection 5.2. Again, only the input changes, but we write the procedure down explicitly to simplify referencing.

**Theorem 6.3.** *Let  $|\psi_a^{\text{noisy}}\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for some  $c \in (0,1]$  satisfying  $c > 1 - \frac{1}{2\sqrt{n}}$ . Further assume that  $2\eta^i(1 - \eta^i) < \frac{1}{5n}$  for all  $1 \leq i \leq n$ , write  $\rho := \max_{1 \leq i \leq n} 2\eta^i(1 - \eta^i)$ . Then  $\mathcal{O}\left(\max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{1}{\delta}\right)\right)$  copies of the quantum example state  $|\psi_a\rangle$  suffice to guarantee that with probability  $\geq 1 - \delta$  Algorithm 6 outputs the string  $a$ .*

**Proof:** We want to prove that  $\mathbb{P}[\text{Algorithm 6 does not output } a] \leq \delta$ , where the probability is w.r.t. both the internal randomness of the algorithm and the random variables.

First observe that, due to (i) in Theorem 6.2, exactly the same reasoning as in the proof of Theorem 5.3

---

**Algorithm 6** Amplified Generalised Bernstein-Vazirani algorithm with noisy training data

---

**Input:**  $m$  copies of  $|\psi_a^{\text{noisy}}\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, \sum_{i=1}^n a_i \frac{1-x_i}{2} + \xi_{x_i}^i\rangle$  for  $a \in \{0,1\}^n$ , where the number of copies is  $m \geq \Omega\left(\max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{1}{\delta}\right)\right)$ , as well as  $\mu \in [-1,1]^n$  and  $c \in (0,1]$  s.t.  $D_\mu$  is  $c$ -bounded

**Output:**  $a \in \{0,1\}^n$

**Success Probability:**  $\geq 1 - \delta$

```

1: for  $1 \leq l \leq m$  do
2:   Run Algorithm 5 on the  $l^{\text{th}}$  copy of  $|\psi_a^{\text{noisy}}\rangle$ , store the output as  $o^{(l)}$ .
3: end for
4: if  $\exists 1 \leq l \leq m : o^{(l)} \neq \perp$  then
5:   for  $1 \leq i \leq n$  do
6:     Let  $o_i = \arg \max_{r \in \{0,1\}} |\{1 \leq l \leq m : o_i^{(l)} = r\}|$ .
7:   end for
8:   Output  $o = o_1 \dots o_n$ .
9: else if  $\forall 1 \leq l \leq m : o^{(l)} = \perp$  then
10:  Output  $o = \perp$ .
11: end if

```

---

shows that the probability of observing  $j_{n+1} = 1$  in at most  $k-1$  of the  $m$  runs of Algorithm 5 (assuming  $k \leq \frac{m}{2}$ ) is bounded by

$$\mathbb{P}\left[\text{Bin}(m, \frac{1}{2}) \geq m - k\right] \leq \exp\left(-\frac{2\left(\frac{m}{2} - k\right)^2}{m}\right). \quad (6.1)$$

We will now search for the number of observations of  $j_{n+1} = 1$  which is required to guarantee that the majority string is correct with high probability. Suppose we observe  $j_{n+1} = 1$  in  $k$  runs of Algorithm 5,  $k \in 2\mathbb{N}$ . Again we see that

$$\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] \leq \mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq o_i] + \mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq o_i].$$

As "false 1's" can only appear in the case where our noise variables have an influence (compare Theorem 6.2), we will first find a lower bound on  $k$  which guarantees that the probability of the noise variable influence becoming relevant for at least  $\frac{k}{5}$  runs is  $\leq \frac{\delta}{4}$ . Namely, we bound (again via Hoeffding)

$$\mathbb{P}[\text{Binom}(k, n\rho) \geq \frac{k}{5}] = \mathbb{P}[\text{Binom}(k, n\rho) - kn\rho \geq k(\frac{1}{5} - n\rho)] \leq \exp\left(-2k\left(\frac{1-5n\rho}{5}\right)^2\right).$$

We now set this last expression  $\leq \frac{\delta}{4}$  and rearrange the inequality to

$$k \geq \frac{25}{2(1-5n\rho)^2} \ln\left(\frac{4}{\delta}\right).$$

Now we will find a lower bound on  $k$  which guarantees that, if the noise variable influence is relevant in

at most  $\frac{k}{5}$  of the runs, among the remaining  $\frac{4k}{5}$  runs with probability  $\geq 1 - \frac{\delta}{4}$  we make at most  $\frac{k}{5}$  "false 0" observations. To this end we bound (again via Hoeffding)

$$\begin{aligned}\mathbb{P}[\text{Binom}(\frac{4k}{5}, n(1-c)^2) \geq \frac{k}{5}] &= \mathbb{P}[\text{Binom}(\frac{4k}{5}, n(1-c)^2) - k\frac{4n(1-c)^2}{5} \geq k(\frac{1}{5} - \frac{4n(1-c)^2}{5})] \\ &\leq \exp\left(-2k(\frac{1}{5} - \frac{4n(1-c)^2}{5})^2\right).\end{aligned}$$

We now set this last expression  $\leq \frac{\delta}{4}$  and rearrange the inequality to

$$k \geq \frac{25}{2(1-4n(1-c)^2)^2} \ln\left(\frac{4}{\delta}\right).$$

Hence, by the union bound a sufficient condition for  $\mathbb{P}[\exists 1 \leq i \leq n : a_i \neq o_i] \leq \frac{\delta}{2}$  is given by

$$k \geq \frac{25}{2} \max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{4}{\delta}\right). \quad (6.2)$$

Combining equations (6.2) and (6.1) we now require

$$\exp\left(-\frac{2\left(\frac{25}{2} \max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{4}{\delta}\right) - \frac{m}{2}\right)^2}{m}\right) \stackrel{!}{\leq} \frac{\delta}{4}.$$

Rearranging gives the sufficient condition

$$m \geq 25 \max\left\{\frac{1}{(1-5n\rho)^2}, \frac{1}{(1-4n(1-c)^2)^2}\right\} \ln\left(\frac{4}{\delta}\right).$$

This proves the claim of the theorem thanks to the union bound.  $\square$

The previous Theorem shows that if the bias is not too strong and if the noise is not too random (i.e. the probability of adding a random 1 is either very low or very high), then learning is possible with essentially the same sample complexity as in the case without noise (compare Theorem 5.3).

Note that the proof of Theorem 6.3 shows that the exact choices of the bounds (in our formulation  $c > 1 - \frac{1}{2\sqrt{n}}$  and  $2\eta^i(1-\eta^i) < \frac{1}{5n}$ ) are flexible to some degree with a trade-off. If we have a better bound on  $c$ , we can loosen our requirement on  $\eta^i$  and vice versa.

Also observe that the requirement of "not too random noise" is natural. If  $2\eta^i(1-\eta^i) \rightarrow 1 \Leftrightarrow \eta^i \rightarrow \frac{1}{2}$ , then the label in the noisy quantum example state becomes random and thus no information on the string  $a$  can be extracted from it. Our bound gives a quantitative version of this intuition.

## 6.2 Faulty Quantum Gates

We now turn to the (more realistic) setting where the quantum gates in our computation (i.e. the  $\mu$ -biased quantum Fourier transforms) are not implemented exactly but only approximately. Then we obtain

**Lemma 6.4.** Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$  be a quantum example state,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$ . Then a version of Algorithm 2 with  $H_\mu$  replaced by  $H_{\tilde{\mu}}$  for  $\|H_\mu - H_{\tilde{\mu}}\|_2 \leq \varepsilon$  provides an outcome  $|j_1 \dots j_{n+1}\rangle$  with the following properties:

- (i)  $|\mathbb{P}[j_{n+1} = 0] - \frac{1}{2}| \leq \varepsilon$  and  $|\mathbb{P}[j_{n+1} = 1] - \frac{1}{2}| \leq \varepsilon$ ,
- (ii)  $|\mathbb{P}[j_1 \dots j_n = a | j_{n+1} = 1] - \prod_{l: a_l=1} (1 - \mu_l^2)| \leq \varepsilon$ ,
- (iii) for  $c \neq a$ :  $\left| \mathbb{P}[j_1 \dots j_n = c | j_{n+1} = 1] - \left( \prod_{l: a_l=0} (1 - c_l) \right) \left( \prod_{l: a_l=1} ((1 - c_l)\mu_l^2 + c_l(1 - \mu_l^2)) \right) \right| \leq \varepsilon$ ,
- (iv)  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 0 \neq j_i | j_{n+1} = 1] \leq \varepsilon$ , and
- (v)  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq \sum_{i=1}^n \mu_i^2 + \varepsilon$ . In particular, if  $D_\mu$  is  $c$ -bounded, then  $\mathbb{P}[\exists 1 \leq i \leq n : a_i = 1 \neq j_i | j_{n+1} = 1] \leq n(1 - c)^2 + \varepsilon$ .

**Proof:** This follows from Theorem 4.3 because the outcome probabilities are the squares of the amplitudes and thus the difference in outcome probabilities can be bounded by the 2-norm of the difference of the quantum states after applying the biased QFT and its approximate version.  $\square$

Now we can proceed analogously to the proof strategy employed in Theorem 6.3 to derive

**Theorem 6.5.** Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for some  $c \in (0,1]$  satisfying  $c > 1 - \sqrt{\frac{1-2\varepsilon}{2n}}$ . Then  $\mathcal{O}\left(\max\left\{\frac{1}{(1-2\varepsilon)^2}, \frac{1}{1-2(n(1-c)^2+\varepsilon)^2}\right\} \ln\left(\frac{1}{\delta}\right) + \varepsilon\right)$  copies of the quantum example state  $|\psi_a\rangle$  suffice to guarantee that with probability  $\geq 1 - \delta$  a version of Algorithm 4 with  $H_\mu$  replaced by  $H_{\tilde{\mu}}$  for  $\|H_\mu - H_{\tilde{\mu}}\|_2 \leq \varepsilon \in (0, \frac{1}{2})$  outputs the string  $a$ .

In particular, the sample complexity upper bound from Theorem 5.3 remains basically untouched if moderately faulty quantum gates are used.

### 6.3 The Case of Unknown Underlying Distributions

An interesting consequence of the result of the previous subsection is the possibility to drop the assumption of prior knowledge of the underlying product distribution, as was already observed in [8] in a similar scenario. The important observations towards this end are given in the following

**Lemma 6.6.** (Lemma 5 in [8])

Let  $A = A_n \dots A_1$  be a product of unitary operators  $A_j$ . Assume that for every  $A_j$  there exists an approximation  $\tilde{A}_j$  s.t.  $\|A_j - \tilde{A}_j\| \leq \varepsilon_j$ . Then it holds that

$$\|A_n \dots A_1 - \tilde{A}_n \dots \tilde{A}_1\| \leq \sum_{j=1}^n \varepsilon_j,$$

i.e. the operator  $\tilde{A} := \tilde{A}_n \dots \tilde{A}_1$  is an  $\varepsilon$ -approximation to  $A$  w.r.t. the operator norm.

**Proof:** This can be proven by induction using the triangle inequality and the fact the a unitary operator has operator norm equal to 1. For details, the reader is referred to [8].  $\square$

This can be used to derive (compare [8])

**Corollary 6.7.** *Let  $\mu \in (-1, 1)^n$  be s.t. the distribution  $D_\mu$  is  $c$ -bounded for  $c \in (0, 1]$ . Let  $\tilde{\mu} \in (-1, 1)^n$  satisfy  $\|\mu - \tilde{\mu}\|_1 \leq \varepsilon$ . Then the corresponding biased quantum Fourier transforms satisfy*

$$\|H_\mu - H_{\tilde{\mu}}\| \leq 2\sqrt{2}n\gamma\varepsilon,$$

where  $\gamma = \frac{1}{c^2} \left( (2-c)\frac{3}{2\sqrt{2}c} + 1 \right)$ .

**Proof:** According to the previous Lemma it holds that

$$\begin{aligned} \|H_\mu - H_{\tilde{\mu}}\| &\leq \sum_{i=1}^n \|\mathbb{1} \otimes \dots \otimes \mathbb{1} \otimes H_{\mu_i} \otimes \mathbb{1} \otimes \dots \otimes \mathbb{1} - \mathbb{1} \otimes \dots \otimes \mathbb{1} \otimes H_{\tilde{\mu}_i} \otimes \mathbb{1} \otimes \dots \otimes \mathbb{1}\| \\ &= \sum_{i=1}^n \|H_{\mu_i} - H_{\tilde{\mu}_i}\|. \end{aligned}$$

Thus it suffices to bound the operator norm of the difference of the 1-qubit biased quantum Fourier transforms. So let  $|\varphi\rangle = \sum_{x \in \{-1, 1\}} \alpha_x |x\rangle$  be a qubit state. Then

$$(H_{\mu_j} - H_{\tilde{\mu}_j})|\varphi\rangle = \sum_{x \in \{-1, 1\}} \sum_{j \in \{0, 1\}} \left( \sqrt{D_{\mu_i}(x)} \phi_{\mu_i, j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)} \phi_{\tilde{\mu}_i, j}(x) \right) \alpha_x |j\rangle.$$

We have to bound the (Euclidean) norm of this vector. To achieve this we will bound (for arbitrary  $x \in \{-1, 1\}$  and  $j \in \{0, 1\}$ ) the expression

$$|\sqrt{D_{\mu_i}(x)} \phi_{\mu_i, j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)} \phi_{\tilde{\mu}_i, j}(x)|^2.$$

This is done by direct computation using  $1 - \mu_i^2 \geq 1 - (1 - c)^2 \geq c^2$ ,  $1 - \tilde{\mu}_i^2 \geq c^2$  and  $|\mu_i - \tilde{\mu}_i| \leq \varepsilon$  as

follows:

$$\begin{aligned}
& \left| \sqrt{D_{\mu_i}(x)} \phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)} \phi_{\tilde{\mu}_i,j}(x) \right| \\
&= \left| \frac{(x_i - \mu_i) \sqrt{1 - \tilde{\mu}_i^2} \sqrt{D_{\mu_i}(x)} - (x_i - \tilde{\mu}_i) \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)}}{\sqrt{1 - \tilde{\mu}_i^2} \sqrt{1 - \mu_i^2}} \right| \\
&\leq \frac{1}{c^2} \left| (x_i - \mu_i) \sqrt{1 - \tilde{\mu}_i^2} \sqrt{D_{\mu_i}(x)} - (x_i - \tilde{\mu}_i) \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right| \\
&= \frac{1}{c^2} \left| (x_i - \mu_i) \left( \sqrt{1 - \tilde{\mu}_i^2} \sqrt{D_{\mu_i}(x)} - \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right) + (\tilde{\mu}_i - \mu_i) \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right| \\
&\leq \frac{1}{c^2} \left( \left| (x_i - \mu_i) \left( \sqrt{1 - \tilde{\mu}_i^2} \sqrt{D_{\mu_i}(x)} - \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right) \right| + \left| (\tilde{\mu}_i - \mu_i) \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right| \right) \\
&\leq \frac{1}{c^2} \left( (2 - c) \left| \sqrt{1 - \tilde{\mu}_i^2} \sqrt{D_{\mu_i}(x)} - \sqrt{1 - \mu_i^2} \sqrt{D_{\tilde{\mu}_i}(x)} \right| + \varepsilon \right) \\
&\leq \frac{1}{c^2} \left( (2 - c) \left( \left| \sqrt{D_{\mu_i}(x)} - \sqrt{D_{\tilde{\mu}_i}(x)} \right| + \left| \sqrt{1 - \mu_i^2} - \sqrt{1 - \tilde{\mu}_i^2} \right| \right) + \varepsilon \right).
\end{aligned}$$

Now note that

$$\begin{aligned}
& \left| \left( \sqrt{D_{\mu_i}(x)} - \sqrt{D_{\tilde{\mu}_i}(x)} \right) \left( \sqrt{D_{\mu_i}(x)} + \sqrt{D_{\tilde{\mu}_i}(x)} \right) \right| = |D_{\mu_i}(x) - D_{\tilde{\mu}_i}(x)| \\
&= \left| \frac{1 + \tilde{x}_i \mu_i}{2} - \frac{1 + \tilde{x}_i \tilde{\mu}_i}{2} \right| \\
&= \frac{1}{2} |\mu_i - \tilde{\mu}_i|,
\end{aligned}$$

which implies

$$\begin{aligned}
& \left| \sqrt{D_{\mu_i}(x)} - \sqrt{D_{\tilde{\mu}_i}(x)} \right| = \left| \frac{\mu_i - \tilde{\mu}_i}{2 \left( \sqrt{D_{\mu_i}(x)} + \sqrt{D_{\tilde{\mu}_i}(x)} \right)} \right| \\
&\leq \frac{\varepsilon}{2} \frac{1}{2\sqrt{\frac{c}{2}}} \\
&= \frac{\varepsilon}{2\sqrt{2c}},
\end{aligned}$$

and that moreover

$$\begin{aligned}
& \left| \left( \sqrt{1 - \mu_i^2} - \sqrt{1 - \tilde{\mu}_i^2} \right) \left( \sqrt{1 - \mu_i^2} + \sqrt{1 - \tilde{\mu}_i^2} \right) \right| = |1 - \mu_i^2 - (1 - \tilde{\mu}_i^2)| \\
&= |\mu_i^2 - \tilde{\mu}_i^2|,
\end{aligned}$$

which in turn implies

$$\begin{aligned}
\left| \sqrt{1 - \mu_i^2} - \sqrt{1 - \tilde{\mu}_i^2} \right| &= \left| \frac{\mu_i^2 - \tilde{\mu}_i^2}{\sqrt{1 - \mu_i^2} + \sqrt{1 - \tilde{\mu}_i^2}} \right| \\
&\leq \frac{|\mu_i + \tilde{\mu}_i| \cdot |\mu_i - \tilde{\mu}_i|}{2\sqrt{1 - (1 - c)^2}} \\
&\leq \frac{2\varepsilon}{2\sqrt{2c - c^2}} \\
&\leq \frac{\varepsilon}{\sqrt{2c}}.
\end{aligned}$$

Hence, we obtain

$$\left| \sqrt{D_{\mu_i}(x)} \phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)} \phi_{\tilde{\mu}_i,j}(x) \right| \leq \frac{1}{c^2} \left( (2 - c) \left( \frac{\varepsilon}{2\sqrt{2c}} + \frac{\varepsilon}{\sqrt{2c}} \right) + \varepsilon \right) \leq \gamma\varepsilon,$$

where we defined  $\gamma := \frac{1}{c^2} \left( (2 - c) \frac{3}{2\sqrt{2c}} + 1 \right)$ . This now implies

$$\begin{aligned}
\|(H_{\mu_j} - H_{\tilde{\mu}_j})|\varphi\rangle\|_2 &\leq \sum_{x \in \{-1,1\}} \sum_{j \in \{0,1\}} \left\| \left( \sqrt{D_{\mu_i}(x)} \phi_{\mu_i,j}(x) - \sqrt{D_{\tilde{\mu}_i}(x)} \phi_{\tilde{\mu}_i,j}(x) \right) \alpha_x |j\rangle \right\|_2 \\
&\leq \gamma\varepsilon \sum_{x \in \{-1,1\}} \sum_{j \in \{0,1\}} |\alpha_x| \\
&= 2\gamma\varepsilon \sum_{x \in \{-1,1\}} |\alpha_x| \\
&\leq 2\sqrt{2}\gamma\varepsilon.
\end{aligned}$$

Finally, we get

$$\|H_\mu - H_{\tilde{\mu}}\| \leq \sum_{i=1}^n \|H_{\mu_i} - H_{\tilde{\mu}_i}\| \leq 2\sqrt{2}n\gamma\varepsilon,$$

as claimed.  $\square$

The next Lemma is on approximating the bias parameter of an unknown product distribution from examples. (Compare the closing remark in Appendix A of [8].)

**Lemma 6.8.** *Using  $m \leq \mathcal{O}(n^4 \ln(\frac{1}{\delta}))$  copies of the quantum example state  $|\psi_a\rangle$  (or of  $|\psi_a^{noisy}\rangle$ ) for a product distribution  $D_\mu$  with bias vector  $\mu \in (-1, 1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for  $c \in (0, 1]$  one can with probability  $\geq 1 - \delta$  output  $\tilde{\mu} \in (-1, 1)^n$  s.t.  $\|H_\mu - H_{\tilde{\mu}}\| \leq \mathcal{O}(1)$ .*

**Proof:** Recall that  $\mu_i = \mathbb{E}_{D_\mu}[x_i]$ . Via a standard application of Hoeffding's inequality we now conclude that  $\mathcal{O}(n^4 \ln(\frac{1}{\delta}))$  examples drawn i.i.d. from  $D_\mu$  (which can be obtained from copies of the quantum example state by measuring the corresponding subsystem) are sufficient to guarantee that with probability  $\geq 1 - \delta$  the empirical estimate  $\hat{\mu}_i$  satisfies  $|\mu_i - \hat{\mu}_i| \leq \mathcal{O}(\frac{1}{n^2})$ . As each component of a copy of the quantum



example state can be measured separately, we see that  $\mathcal{O}(n^4 \ln(\frac{1}{\delta}))$  copies of the (possibly noisy) quantum example state suffice to guarantee that with probability  $\geq 1 - \delta$  it holds that  $\|\mu - \hat{\mu}\|_1 = \sum_{i=1}^n |\mu_i - \hat{\mu}_i| \leq \mathcal{O}(n \frac{1}{n^2}) = \mathcal{O}(\frac{1}{n})$ . Now we can apply the previous Corollary to finish the proof.  $\square$

If we now combine this result with Theorem 6.5, we obtain a sample complexity upper bound for our learning problem without assuming the underlying distribution to be known in advance.

**Corollary 6.9.** *Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded for some  $c \in (0,1]$  satisfying  $c > 1 - \sqrt{\frac{1-2\varepsilon}{2n}}$ . Then  $\mathcal{O}\left(n^4 \ln(\frac{1}{\delta}) + \max\left\{\frac{1}{(1-2\varepsilon)^2}, \frac{1}{1-2(n(1-c)^2+\varepsilon)^2}\right\} \ln\left(\frac{1}{\delta}\right)\right)$  copies of the quantum example state  $|\psi_a\rangle$  suffice for an algorithm that with probability  $\geq 1 - \delta$  outputs the string  $a$ , without prior knowledge of the underlying distribution.*

Note that we do not explicitly formulate the result for noisy quantum training data but such a generalization is clearly possible.

## 7 Sample Complexity Lower Bounds

After proving upper bounds on the number of required quantum examples by exhibiting explicit learning procedures in the previous sections, we now study the converse question of sample complexity lower bounds. We will prove both classical and quantum sample complexity lower bounds and then relate them to the above results. Our proof strategy follows the information-theoretic procedure first presented in [2].

### 7.1 Classical Sample Complexity Lower Bounds

We first prove a sample complexity lower bound for the classical version of our learning problem that upon comparison with our obtained quantum sample complexity upper bounds shows the advantage of quantum examples over classical training data in this setting. Neither the result nor the proof strategy are new but we include it for completeness.

**Theorem 7.1.** *Let  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $\mu$  is  $c$ -bounded for some  $c \in (0,1]$ . Let  $\mathcal{A}$  be a classical learning algorithm and let  $m \in \mathbb{N}$  be such that upon input of  $m$  examples of the form  $(x_i, f^{(a)}(x_i))$ , with  $x_i$  drawn i.i.d. according to  $D_\mu$ , with probability  $\geq 1 - \delta$  w.r.t. the choice of training data  $\mathcal{A}$  outputs the string  $a$ . Then  $m \geq \Omega(n)$ .*

**Proof:** Let  $A$  be a random variable uniformly distributed on  $\{0,1\}^n$ . ( $A$  describes the underlying string from the initial perspective of the learner. Let  $B = (B_1, \dots, B_m)$  be a random variable describing the training data corresponding to the underlying string. Our proof will have three main steps: First, we prove a lower bound on  $I(A : B)$  from the learning requirement. Second, we observe that  $I(A : B) \leq mI(A : B_1)$ . And third, we prove an upper bound on  $I(A : B_1)$ . Then combining the three steps will lead to a lower bound on  $m$ .

We start with the mutual information lower bound. Let  $h(B) \in \{0,1\}^n$  denote the random variable

describing the output hypothesis of the algorithm  $\mathcal{A}$  upon input of training data  $B$ . Let  $Z = \mathbb{1}_{h(B)=A}$ . By the learning requirement we have  $\mathbb{P}[Z = 1] \geq 1 - \delta$  and thus  $H(Z) \leq H(\delta)$ . Thus we obtain

$$\begin{aligned}
I(A : B) &= H(A) - H(A|B) \\
&\geq H(A) - H(A|B, Z) - H(Z) \\
&= H(A) - \mathbb{P}[Z = 1]H(A|B, Z = 1) - \mathbb{P}[Z = 0]H(A|B, Z = 0) - H(Z) \\
&\geq n - (1 - \delta) \cdot 0 - \delta n - H(\delta) \\
&= (1 - \delta)n - H(\delta) \\
&= \Omega(n).
\end{aligned}$$

We now show that from  $m$  examples we can gather at most  $m$  times as much information as from a single example. Here we directly cite from [2]. Namely,

$$\begin{aligned}
I(A : B) &= H(B) - H(B|A) = H(B) - \sum_{i=1}^m H(B_i|A) \\
&\leq \sum_{i=1}^m H(B_i) - H(B_i|A) = \sum_{i=1}^m I(A : B_i) = m \cdot I(A : B_1).
\end{aligned}$$

Here, the second step uses independence of the  $B_i$  conditioned on  $A$ , the third step uses subadditivity of the Shannon entropy and the final step uses that the distributions of  $(A, B_i)$  are the same for all  $1 \leq i \leq m$ .

We come to the upper bound on the mutual information. Write  $B_1 = (X, L)$  for  $X \in \{-1, 1\}^n$  and  $L \in \{0, 1\}$ , i.e. with probability  $D_\mu(x)$  we have  $(X, L) = (x, f^{(a)}(x))$ . Note that  $I(A : X) = 0$  because  $X$  and  $A$  are independent random variables. Also,  $I(A : L|X = 1 \dots 1) = 0$  because  $f^{(a)}(1 \dots 1) = 0 \forall a \in \{0, 1\}^n$ , and for  $x \in \{-1, 1\}^n \setminus \{1 \dots 1\}$

$$\begin{aligned}
I(A : L|X = x) &= I(A_{\{i|X_i=-1\}} : L|X = x) \\
&= H(A_{\{i|X_i=-1\}}|X = x) - H(A_{\{i|X_i=-1\}}|L, X = x) \\
&= |\{i|x_i = -1\}| - (|\{i|x_i = -1\}| - 1) \\
&= 1.
\end{aligned}$$

Here, the first step is due to the fact that  $f^{(a)}(x)$  does not depend on the entries  $a_j$  with  $x_j = 1$ , the third step follows because  $A_{\{i|x_i=-1\}}$  is uniformly distributed on a set of size  $2^{|\{i|x_i=-1\}|}$  and  $f^{(a)}$  assigns the labels 0 and 1 to half of the elements of that set, respectively.

This now implies

$$\begin{aligned}
I(A : B_1) &= I(A : X) + I(A : L|X) \\
&= 0 + \sum_{x \in \{-1,1\}^n} D_\mu(x) I(A : L|X = x) \\
&= 1.
\end{aligned}$$

Here, the first step is due to the chain rule for mutual information and the last step simply uses the fact that  $D_\mu$  defines a probability distribution.

Now we combine our upper and lower bounds on the mutual information and obtain

$$m \geq (1 - \delta)n - H(\delta) = \Omega(n),$$

as claimed. □

**Remark 7.2.** The result of Theorem 7.1 is intuitively clear: In order to identify the underlying string the learning algorithm has to learn  $n$  bits of information. However, a condition of the form  $f^{(a)}(x) = l$  for  $x \in \{0,1\}^n, l \in \{0,1\}$  takes away at most one degree of freedom from the initial space  $\{0,1\}^n$  for  $a$  and thus from such an equality the algorithm can extract at most 1 bit of information. So at least  $n$  examples will be required. This observation is thus neither new nor surprising. But we want to emphasize that this analysis works independently of the product structure of the underlying distribution  $D_\mu$ .

If we compare the classical lower bound from Theorem 7.1 with our quantum upper bounds from Theorems 5.1 and 5.3, we conclude that quantum examples allow us to strictly outperform the best possible classical algorithm w.r.t. the number of required examples.

## 7.2 Quantum Sample Complexity Lower Bounds

We now want to use a similar argument to prove quantum sample complexity lower bounds. Note that steps 1 and 2 carry over with (almost) no changes. Only the analysis of step 3 will change significantly. Moreover, the argument will give us the  $n$ -dependent part of a quantum sample complexity lower bound, for the  $n$ -independent part (i.e. the part governed by the confidence parameter  $\delta$ ) we will provide a different (simpler) argument based on state discrimination. This was already used in [2] to show a confidence-dependent quantum sample complexity lower bound. (Note that any quantum sample complexity lower bound will also lower bound the classical sample complexity, which is why we did not discuss the  $\delta$ -dependence previously.)

An  $n$ -independent quantum sample complexity lower bound is given in the following

**Lemma 7.3.** *Let  $|\psi_a\rangle = \sum_{x \in \{-1,1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0,1\}^n$ ,  $\mu \in (-1,1)^n$  s.t.  $D_\mu$  is  $c$ -bounded*

for some  $c \in (0, 1]$ . Let  $\mathcal{A}$  be a quantum learning algorithm and let  $m \in \mathbb{N}$  be such that upon input of  $m$  copies of  $|\psi_a\rangle$ , with probability  $\geq 1 - \delta$   $\mathcal{A}$  outputs the string  $a$ . Then  $m \geq \Omega(\frac{1}{c} \ln(\frac{1}{\delta}))$ .

**Proof:** Let  $a, b \in \{0, 1\}^n$  s.t. there is exactly one  $1 \leq i \leq n$  s.t.  $a_i \neq b_i$ . As  $\mathcal{A}$  is able to distinguish the quantum states  $|\psi_a\rangle^{\otimes m}$  and  $|\psi_b\rangle^{\otimes m}$  with success probability  $\geq 1 - \delta$ , we have  $|\langle \psi_a | \psi_b \rangle^m| \leq \sqrt{\delta(1 - \delta)}$  (see e.g. chapter 9 in [12]). We compute

$$\langle \psi_a | \psi_b \rangle = \sum_{x, y \in \{-1, 1\}^n} \sqrt{D_\mu(x) D_\mu(y)} \langle x, f^{(a)}(x) | y, f^{(b)}(y) \rangle = \sum_{x \in \{-1, 1\}^n} D_\mu(x) \delta_{f^{(a)}(x), f^{(b)}(x)}.$$

By our assumption on  $a$  and  $b$ ,  $\delta_{f^{(a)}(x), f^{(b)}(x)} \geq \delta_{x_i, 1}$ . Therefore

$$\langle \psi_a | \psi_b \rangle \geq \mathbb{P}_{D_\mu}[x_i = 1] = \frac{1 + \mu_i}{2}.$$

We now combine this with our upper bound and rearrange to obtain

$$m \geq \left( \ln \left( \frac{1 + \mu_i}{2} \right) \right)^{-1} \left( \frac{1}{2} \ln(\delta(1 - \delta)) \right) \geq \Omega \left( \frac{1}{\mu_i - 1} \ln(\delta) \right) \geq \Omega \left( \frac{1}{c} \ln \left( \frac{1}{\delta} \right) \right),$$

where we used the elementary inequality  $\frac{1}{x-1} - \left( \ln \left( \frac{1+x}{2} \right) \right)^{-1} \geq 0$  for  $x \in [0, 1)$  combined with  $\ln(\delta) \leq 0$ .  $\square$

We will compare this lower bound with our upper bound(s) from section 5 later on. Now we turn to the  $n$ -dependent part of the sample complexity lower bound.

**Theorem 7.4.** Let  $|\psi_a\rangle = \sum_{x \in \{-1, 1\}^n} \sqrt{D_\mu(x)} |x, f^{(a)}(x)\rangle$ ,  $a \in \{0, 1\}^n$ , and  $\mu \in (-1, 1)^n$  be such that  $\mu_i \geq 2 \sqrt[n]{1 - \frac{1}{\ln(n)}} - 1 \forall 1 \leq i \leq n$ . Let  $\mathcal{A}$  be a quantum learning algorithm and let  $m \in \mathbb{N}$  be such that upon input of  $m$  copies  $|\psi_a\rangle$ , with probability  $\geq 1 - \delta$   $\mathcal{A}$  outputs the string  $a$ . Then  $m \geq \Omega(\ln(n))$ .

**Proof:** Let  $A$  be as in the proof of Theorem 7.1. As discussed above, only step 3 of the proof of Theorem 7.1 has to be significantly modified. For step 1, we simply replace in our reasoning the quantum example system  $B$  by the classical output random variable  $h(B)$ , then we can use the same classical tools as before. Step 2 carries over as is because of additivity of the entropy of product states and because of subadditivity of the quantum entropy.

Now for step 3. The composite  $AB$ -system of underlying string and corresponding quantum examples is described by the classical-quantum state

$$\rho_{AB} = \frac{1}{2^n} \sum_{a \in \{0, 1\}^n} |a\rangle \langle a| \otimes |\psi_a\rangle \langle \psi_a|^{\otimes m}.$$

As  $\rho_{AB_1}$  is a classical-quantum state, we have

$$I(A : B_1) = S(A) + S(B_1) - S(AB_1).$$

As  $A \sim \text{Uniform}(\{0, 1\}^n)$ ,  $S(A) = H(A) = n$ . As  $\rho_{AB_1} = \frac{1}{2^n} \sum_{a \in \{0, 1\}^n} |a\rangle\langle a| \otimes |\psi_a\rangle\langle\psi_a|$ , is block-diagonal with  $2^n$  rank-1 blocks on the diagonal, also  $S(AB_1) = n$ . So  $I(A : B_1) = S(B_1)$ .

We now have that  $\rho_{B_1} = \frac{1}{2^n} \sum_{a \in \{0, 1\}^n} |\psi_a\rangle\langle\psi_a|$  and we want to bound the entropy of this state. Observe that  $\forall a \in \{0, 1\}^n$  the (pure) state  $|\psi_a\rangle\langle\psi_a|$  represented w.r.t. the standard ONB  $\{|x, l\rangle\}_{x \in \{-1, 1\}^n, l \in \{0, 1\}}$  has diagonal entries

$$\langle x, l | \psi_a \rangle \langle \psi_a | x, l \rangle = \delta_{l, f^{(a)}(x)} D_\mu(x) \quad \text{for } x \in \{-1, 1\}^n, l \in \{0, 1\}.$$

So the diagonal entries of  $\rho_{B_1}$  are

$$\langle x, l | \rho_{B_1} | x, l \rangle = \frac{D_\mu(x)}{2^n} \sum_{a \in \{0, 1\}^n} \delta_{l, f^{(a)}(x)}.$$

We have  $f^{(a)}((1, \dots, 1)) = 0 \forall a \in \{0, 1\}^n$  as well as

$$\left| \{a \in \{0, 1\}^n | f^{(a)}(x) = 0\} \right| = \left| \{a \in \{0, 1\}^n | f^{(a)}(x) = 1\} \right| \quad \text{for all } x \in \{-1, 1\}^n \setminus \{(1, \dots, 1)\}.$$

Hence, we obtain

$$\langle x, l | \rho_{B_1} | x, l \rangle = \begin{cases} \delta_{l, 0} D_\mu((1, \dots, 1)) & \text{if } x = (1, \dots, 1) \\ \frac{1}{2} D_\mu(x) & \text{else} \end{cases}.$$

Now according to the Courant-Fischer-Weyl min-max principle (here we use that  $\rho_{B_1}$  is positive semidefinite and thus in particular hermitian) we get

$$\lambda_1 \geq \langle x, l | \rho_{B_1} | x, l \rangle = \max\{D_\mu((1, \dots, 1)), \frac{1}{2} \max_{x \in \{-1, 1\}^n \setminus \{(1, \dots, 1)\}} D_\mu(x)\},$$

where  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{2^{n+1}} \geq 0$  denote the eigenvalues of  $\rho_{B_1}$ . Now let  $N$  be sampled from  $\{1, \dots, 2^{n+1}\}$  according to the eigenvalues of  $\rho_{B_1}$ ,  $Z = \mathbb{1}_{N \neq 1}$ , then we obtain

$$\begin{aligned} S(\rho_{B_1}) &= H(N) = H(N, Z) = H(Z) + H(N|Z) \\ &= H(Z) + \mathbb{P}[Z = 0] \underbrace{H(N|Z = 0)}_{=0} + \mathbb{P}[Z = 1] H(N|Z = 1) \\ &= H(Z) + (1 - \lambda_1) \underbrace{H(N|Z = 1)}_{\leq \log(2^{n+1}-1) \leq n+1} \\ &\leq H(1 - \lambda_1) + n + 1. \end{aligned}$$

Clearly,  $D_\mu(1, \dots, 1) = \prod_{i=1}^n \frac{1+\mu_i}{2}$  and  $\max_{x \in \{-1, 1\}^n \setminus \{(1, \dots, 1)\}} D_\mu(x) = \prod_{i=1}^n \frac{1+|\mu_i|}{2}$ . Now by our assumption on  $\mu$

we have

$$\prod_{i=1}^n \frac{1 + \mu_i}{2} \geq 1 - \frac{1}{\ln(n)},$$

which now implies  $\lambda_1 \geq 1 - \frac{1}{\ln(n)}$  and we obtain

$$S(\rho_{B_1}) \leq H\left(\frac{1}{\ln(n)}\right) + \frac{n+1}{\ln(n)}.$$

We now combine this upper bound with steps 1 and 2 and obtain

$$(1 - \delta)n - H(\delta) \leq m \left( H\left(\frac{1}{\ln(n)}\right) + \frac{n+1}{\ln(n)} \right).$$

Rearranging this we obtain  $m \geq \Omega(\ln(n))$ , as desired.  $\square$

We now compare this lower bound to our previously obtained upper bounds. First, we consider the  $n$ -independent part of the bounds. When comparing Theorem 5.1 with Lemma 7.3, we obtain

$$\Omega\left(\frac{1}{c} \ln\left(\frac{1}{\delta}\right)\right) \leq m \leq \mathcal{O}\left(\left(\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1} \ln\left(\frac{1}{\delta}\right)\right).$$

We study this for  $\delta \ll 1$  (high confidence) and  $c \ll 1$  (high bias). Then Taylor expansion shows

$$\left(\ln\left(\frac{1}{1-c+\frac{c^2}{2}}\right)\right)^{-1} = \frac{1}{c} + \frac{c}{6} + \mathcal{O}(c^2) \quad \text{for } c \ll 1.$$

Hence, lower and upper bound coincide in the relevant region for  $\delta$  and  $c$ , so the  $n$ -independent part of the sample complexity upper bound provided by algorithm 3 is optimal.

However, in comparing 5.3 with Lemma 7.3 we see a discrepancy between lower and upper bound for the relevant region  $\delta \ll 1$  and  $c - (1 - \frac{1}{\sqrt{2n}}) \ll 1$ . Therefore we conjecture that the  $c$ -dependence of the upper bound arising from Theorem 5.3 is not optimal.

Now we compare the bounds w.r.t. the  $n$ -dependence, i.e. we compare Theorem 5.1 with Lemma 7.4, and obtain

$$\Omega(\ln(n)) \leq m \leq \mathcal{O}\left(\frac{1}{c} \ln(n)\right).$$

So here we lack the information of the dependence of the lower bound on the  $c$ -boundedness constant (which we have to assume to be small in order for Theorem 7.4 to be applicable), which makes comparing the two difficult. However, the  $n$ -dependence (when thinking of  $c$  as a fixed constant) is the same.

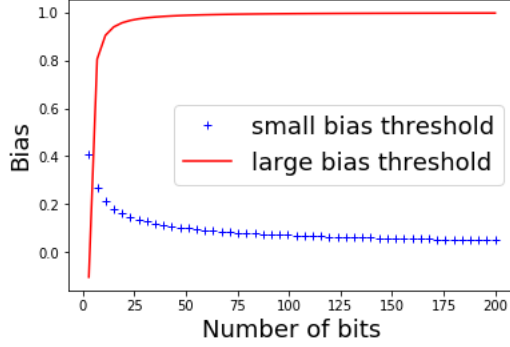


Figure 1: A plot comparing the maximal bias allowed in Theorem 5.3 (depicted by the blue crosses) with the minimal bias required in Theorem 7.4 (depicted by the red line).

Finally, we want to point towards an unsatisfactory aspect of our results. We provide an  $n$ -dependent quantum sample complexity lower bound for "large" noise and an  $n$ -independent quantum sample complexity upper bound for "small" noise. However, there is a large discrepancy between the obtained characterizations of "small" and "large" noise, which can be seen in Figure 1.

Hence, we did not succeed in identifying a bias threshold beyond which the sample complexity qualitatively differs from the unbiased case, but merely provided a region in which it lies. To improve upon our results it would be necessary to modify either the proof of Theorem 5.3 to allow for stronger bias or the proof of Theorem 7.4 to allow for weaker bias. However, in both cases we currently do not see whether our proof strategy admits such an improvement.

## 8 Conclusion and Outlook

In this paper we extended a well-known quantum learning strategy for linear functions from the uniform distribution to biased product distributions. This approach naturally led to a distinction between a procedure for arbitrary (not full) bias and a procedure for small bias, the latter with a significantly better performance. Moreover, we showed that the second procedure is (to a certain degree) stable w.r.t. noise in the training data and in the performed quantum gates. Finally, we also provided lower bounds on the size of the training data required for the learning problem, both in the classical and in the quantum setting.

We want to conclude by outlining some open questions for future work:

- Can we identify a bias threshold s.t. the optimal sample complexity below the threshold differs qualitatively from the one above it?
- Is our learning procedure for small bias also stable w.r.t. different types of noise in the training data, e.g. malicious noise?

- Our explicit learning algorithms also give upper bounds on the computational complexity of our learning problem. Can we find corresponding lower bounds to facilitate a discussion of optimality w.r.t. runtime?
- Can we find more examples of learning tasks (i.e. function classes) where quantum training data yields an advantage w.r.t. sample and/or time complexity?

## 9 Acknowledgements

First, I want to thank my supervisor Michael Wolf for several stimulating discussions concerning questions of quantum learning. Also, I want to thank Benedikt Graswald for proofreading a first draft of this paper and for his constructive comments.

Support from the TopMath Graduate Center of the TUM Graduate School at the Technische Universität München, Germany and from the TopMath Program at the Elite Network of Bavaria is gratefully acknowledged.



## References

- [1] Srinivasan Arunachalam and Ronald de Wolf. *A Survey of Quantum Learning Theory*. 2017. URL: <https://arxiv.org/pdf/1701.06806> (cited on p. 3).
- [2] Srinivasan Arunachalam and Ronald de Wolf, eds. *Optimal Quantum Sample Complexity of Learning Algorithms: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik GmbH, Wadern/Saarbruecken, Germany*. 2017. DOI: 10.4230/LIPICS.CCC.2017.25 (cited on pp. 3, 25–27).
- [3] Ethan Bernstein and Umesh Vazirani. ‘Quantum complexity theory’. In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. Ed. by Rao Kosaraju. New York, NY: ACM, 1993, pp. 11–20. ISBN: 0897915917. DOI: 10.1145/167088.167097 (cited on pp. 1, 3).
- [4] Nader H. Bshouty and Jeffrey C. Jackson. ‘Learning DNF over the Uniform Distribution Using a Quantum Example Oracle’. In: *SIAM Journal on Computing* 28.3 (1998), pp. 1136–1153. ISSN: 0097-5397. DOI: 10.1137/S0097539795293123 (cited on pp. 3, 5).
- [5] Chandra Chekuri, ed. *Approximation, randomization and combinatorial optimization: Algorithms and techniques ; 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th International Workshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22 - 24, 2005 ; proceedings*. Vol. 3624. Lecture Notes in Computer Science. Berlin: Springer, 2005. ISBN: 978-3-540-28239-6. DOI: 10.1007/11538462.
- [6] Richard P. Feynman. ‘Quantum Mechanical Computers’. In: *Optics News* 11.2 (1985), p. 11. ISSN: 0098-907X. DOI: 10.1364/ON.11.2.000011 (cited on p. 3).
- [7] Alex B. Grilo, Iordanis Kerenidis, and Timo Zijlstra. *Learning with Errors is easy with quantum samples*. 2017. URL: <http://arxiv.org/pdf/1702.08255> (cited on pp. 1, 3, 12, 15).
- [8] Varun Kanade, Andrea Rocchetto, and Simone Severini. *Learning DNFs under product distributions via  $\mu$ -biased quantum Fourier sampling*. 2018. URL: <http://arxiv.org/pdf/1802.05690> (cited on pp. 1, 3, 6, 7, 21, 22, 24).
- [9] A. Yu Kitaev. ‘Quantum computations: Algorithms and error correction’. In: *Russian Mathematical Surveys* 52.6 (1997), pp. 1191–1249. ISSN: 0036-0279. DOI: 10.1070/RM1997v052n06ABEH002155.

- [10] Rao Kosaraju, ed. *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. New York, NY: ACM, 1993. ISBN: 0897915917. DOI: 10.1145/167088.
- [11] Vadim Lyubashevsky. ‘The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem’. In: *Approximation, randomization and combinatorial optimization*. Ed. by Chandra Chekuri. Vol. 3624. Lecture Notes in Computer Science. Berlin: Springer, 2005, pp. 378–389. ISBN: 978-3-540-28239-6. DOI: 10.1007/11538462\_32 (cited on p. 7).
- [12] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. 10th anniversary ed. Cambridge: Cambridge Univ. Press, 2010. ISBN: 9781282967298. URL: <https://ebookcentral.proquest.com/lib/subhh/detail.action?docID=647366> (cited on pp. 3, 5, 28).
- [13] Ryan O’Donnell. *Analysis of Boolean functions*. 1. publ. 2014. ISBN: 9781107038325 (cited on p. 5).
- [14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. 1. publ. 2014. ISBN: 9781107057135 (cited on p. 3).
- [15] L. G. Valiant. ‘A theory of the learnable’. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142. ISSN: 00010782. DOI: 10.1145/1968.1972 (cited on pp. 3, 5, 7).
- [16] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Vol. 47. Cambridge series in statistical and probabilistic mathematics. Cambridge: Cambridge University Press, 2018. ISBN: 1108415199 (cited on p. 13).