# Algorithmic Models in Quantum Mechanics


Andrea Rocchetto

St Peter's College

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

Trinity 2019

# Acknowledgements

I would like to thank Scott Aaronson, Simon Benjamin, Giuseppe Carleo, Varun Kanade, Fabio Sciarrino, Simone Severini and Ronald de Wolf for helpful discussions, and Jonathan Barrett, Edoardo Borgomeo, Carlo Ciliberto, Alessandro Ialongo, Iordanis Kerenidis, Francesca Lussana, Frank Schindler, and Mithuna Yoganathan for a thorough reading of parts of this manuscript. I gratefully acknowledge the support from the Engineering and Physical Sciences Research Council and from QinetiQ. I thank Muddy Bhatt, Alessandro Ialongo, Meng Lu, and Sara Venturini for their hospitality in Cambridge and London. Portions of the work included in this thesis were completed while I was visiting the Institut Henri Poincaré, the Kavli Institute for Theoretical Physics, the University of California, Berkeley, and University College London, and I would like to thank these institutions for their hospitality.

# Abstract

We study classical and quantum learning algorithms with access to data produced by a quantum process. First, we consider the problem of learning quantum states and, in the framework of the probably approximately correct (PAC) model, prove that stabiliser states are efficiently learnable. Second, we introduce a generative model based on artificial neural networks capable of finding efficient representations of quantum states and assess its performance on states with varying levels of complexity. Third, we discuss the time complexity of classical and quantum learning algorithms and prove that Boolean functions in disjunctive normal form are efficiently quantum PAC learnable under product distributions.

# Contents


**1 Introduction** **1**

1.1 Summary of our contributions . . . . . 2

1.1.1 Identifying quantum states that can be learned efficiently . . . 2

1.1.2 Learning efficient representations of quantum many-body states 3

1.1.3 Separating classical and quantum learnability . . . . . 4

1.1.4 Statement of authorship . . . . . 5

**2 Preliminaries** **6**

2.1 Notation . . . . . 6

2.2 Learning theory . . . . . 7

2.2.1 The probably approximately correct model . . . . . 9

2.3 Quantum computation . . . . . 10

2.3.1 Classical simulation . . . . . 10

2.3.2 The Pauli group . . . . . 11

**3 Efficient learning of quantum states** **12**

3.1 Overview of our results . . . . . 13

3.2 Related work . . . . . 14

3.3 Sample complexity of learning quantum states . . . . . 16

3.4 Time complexity of learning quantum states . . . . . 18

3.5 Stabiliser states . . . . . 19

3.6 Learning stabiliser states . . . . . 20

3.7 Experimental learning of stabiliser states . . . . . 24

3.7.1 GHZ states and learning distributions . . . . . 24

3.7.2 Learning theorems and experimental data . . . . . 25

3.7.3 Numerical simulations . . . . . 27

3.7.4 Experimental demonstration of GHZ learnability . . . . . 29

**4 Learning efficient representations of quantum states** **32**
4.1 Overview of our results . . . 33
4.2 Related work . . . 34
4.3 Generative modelling with VAEs . . . 35
4.4 Encoding quantum probability distributions with VAEs . . . 39
4.5 Hard and easy quantum states . . . 41
4.5.1 States that are classically hard to sample from . . . 43
4.6 The role of depth in compressibility . . . 45
4.7 Efficient representation of quantum states . . . 46

**5 Quantum PAC learnability of DNFs** **49**
5.1 Overview of our results . . . 50
5.2 Related work . . . 52
5.3 Fourier analysis over the Boolean cube . . . 53
5.3.1 $\mu$–biased Fourier analysis . . . 54
5.4 The quantum PAC model . . . 55
5.5 DNF learnability and Fourier spectrum . . . 56
5.6 Quantum $\mu$–biased Fourier transform . . . 58
5.7 Quantum computation of $\mu$–biased Fourier spectrum . . . 61
5.8 Error analysis . . . 63

**6 Conclusions** **67**
6.1 Future work . . . 67
6.2 Perspective: algorithmic models in physics . . . 69

**A Group Theory** **72**

**B Concentration of measure** **74**

**Bibliography** **76**

# Chapter 1

# Introduction

Machine learning algorithms can be used to model chaotic systems [Pat+18], classify jets of subatomic particles [KMS17], and compute ground state energies of quantum many-body states [CT17] using data as the only starting point (we recommend the review article by Carleo et al. for more applications of learning algorithms in physics [Car+19a]). As data-driven algorithmic models make their way into physics, it is becoming relevant to assess the validity of the conclusions they provide and identify problems for which they can be effective. This calls for a theoretical analysis of the strengths and limitations of learning algorithms applied to physical problems. There is a vast toolbox at our disposal. The fields of statistical and computational learning theory, themselves undergoing a rapid development, provide coherent mathematical frameworks to formulate and answer questions on the performance of learning models. How much data do we need to approximate a given function to a certain accuracy? What is the computational cost of a learning algorithm? Although the assumptions of learning theory can be too conservative and are unable to explain the success of popular learning models (see for example [Zha+16; BMM18]), they provide a good starting point for the analysis of learning algorithms applied to physical problems. Undertaking such study will be the main subject of this thesis.

According to our current best description of reality, quantum mechanics is the most fundamental description of the natural world (at least at small scales). As such, any effort devoted to understand the modelling capabilities of learning algorithms must consider data and learning agents governed by the laws of quantum mechanics. Such an endeavour poses novel challenges and opportunities for learning theory. On the one hand, the exponential scaling of the wave function threatens the possibility of developing efficient, and thus useful, algorithmic models. On the other, devices that can exploit coherent quantum information for computational purposes promise to solve efficiently tasks for which no classical computer is expected to do so. Starting from

these observations we seek to advance our understanding of the power and limitations of algorithmic models of quantum systems focusing on three main issues:

1. The study of provable performance guarantees for classical learning algorithms with access to data measured on a quantum system.

2. The design of algorithms capable of finding efficient representations of quantum states solely from data.

3. The identification of learning problems that can only be solved efficiently by a quantum computer.

In the following section we discuss the main contributions of this thesis.

## 1.1 Summary of our contributions

We begin in Chapter 2 with a brief review of relevant concepts from learning theory and quantum computation. The structure of the thesis follows the two challenges outlined above. First, we study classical learning algorithms operating on data coming from a quantum experiment. More specifically, in Chapter 3 we analyse from a computational perspective the problem of the learnability of quantum states, while in Chapter 4 we discuss how machine learned representations can be used to perform efficient calculations with quantum many-body states. We summarise the major contributions of these chapters in Section 1.1.1 and Section 1.1.2, respectively. Second, in Chapter 5 we consider quantum learning algorithms with access to a quantum superposition of the training set. The results of the chapter are summarised in Section 1.1.3. In Section 1.1.4 we present a statement of authorship. Finally, in Chapter 6 we discuss future directions and give some remarks on the difference between scientific theories and algorithmic models.

### 1.1.1 Identifying quantum states that can be learned efficiently

Finding a quantum state in agreement with a set of expected values of observables is a common task in quantum information processing. If the set of observables is complete, i.e. forms a basis in the space of density matrices, then the problem reduces to quantum state tomography and it is known to require exponentially many measurements [Haa+17].

If the set is not complete there might be multiple quantum states that satisfy the constraints and in order to have a point estimate it is necessary to adopt an inference rule. Common methods in quantum information include the maximum likelihood principle [Ban+99], Bayesian inference [Blu10], or Jaynes's maximum entropy principle [Bužˇ04].

The problem of learning quantum states, which has been formally introduced by Aaronson in [Aar07], is related to partial state reconstruction. When learning a quantum state, the goal is not to find a state that best matches the experimental data, but to produce an hypothesis that is able to predict unseen measurements. Interestingly, Aaronson proved that, under suitably defined notions of learning, a set of measurements with a number of elements that grows linearly with the number of qubits contains sufficient information to construct an hypothesis that generalises well. However, although from an information theoretic perspective the problem can be solved efficiently, the underlying computational task required to generate the hypothesis is generally hard. This raised the question of whether learning quantum states can not only be informationally but also computationally efficient.

In Chapter 3 we positively resolve this question and show that stabiliser states under Pauli measurements are efficiently learnable. We also present the first experimental demonstration, on a photonic platform with up to 6 qubits, of the learning theorem proved by Aaronson. We do not discuss in detail the experimental setup (all the specifics are presented in [Roc+19]) and focus on showing how to derive an experimentally testable case of the learning theorem.

### 1.1.2 Learning efficient representations of quantum many-body states

As mentioned in Section 1.1.1, the challenges posed by the many-body problem in quantum physics stems from the difficulty of performing mathematical operations on an object that contains an exponential number of parameters. This has implications for our ability to describe and simulate the evolution of quantum systems.

Traditionally, hard computational problems in quantum physics have been approached using methods that leverage a combination of approximation techniques (such as perturbation theory or semi-classical approaches) and the use of symmetries to reduce the complexity of the problem. Machine learning techniques offer a new way of solving the problem. Rather than exploiting a structural property of the system (such as the entanglement structure) a learning algorithm relies on minimal assumptions to find an efficient representation of the state solely from data.

In Chapter 4 we present a novel representation of quantum states based on variational autoencoders. These are types of generative models, i.e. models that can learn a probability distribution and then generate new samples from it, built on feedforward neural networks [KW13; RMW14]. After having defined a suitable notion of complexity, we analyse the behaviour of our model on hard and easy states. We show that variational autoencoders are capable of representing quantum states that are known to possess an efficient representation using exponentially fewer parameters than those contained in their wave functions. In this sense the model 'rediscovers' from data that the state can be represented efficiently.

We also analyse states that are hard to represent and show that in this case the network representation has only a small compression factor. Finally, we study how the depth of the network influences its representational capabilities and present evidence that depth plays an important role, even when the state has correlations that arise from quantum processes that can not be efficiently reproduced by a classical computer.

### 1.1.3 Separating classical and quantum learnability

A central topic in quantum learning theory is to identify learning problems for which it can be advantageous to use a quantum computer. Very few such problems are known. Among them is the learnability of the class of Boolean functions that can be expressed as polynomial size formulae in disjunctive normal form (DNF). Classically, the time complexity of the best algorithm for learning DNFs under an unknown distribution is exponential [KS01], but when considering examples drawn from the uniform distribution then the runtime of the best learner becomes quasipolynomial [Ver90]. By using a quantum learning model, Bshouty and Jackson showed that DNFs can be efficiently learned under the uniform distribution [BJ98]. The speedup is obtained by using an efficient quantum algorithm for sampling the probability distribution described by the Fourier coefficients of a Boolean function.

In Chapter 5 we extend the result on the learnability of disjunctive normal forms to product distribution, a class of probability distributions that generalises the uniform distribution. We thus establish a superpolynomial separation between the classical and quantum case for a larger class of problems than was previously known. Our proof 'quantises' in a rigorous way two classical algorithms: the Kushilevitz-Mansour theorem for learning decision trees [KM93] and the $\mu$-biased Fourier transform, a variant of the Fourier transform defined over the weighted Boolean cube.

### 1.1.4 Statement of authorship

This thesis is based on the following papers:

1. [Roc18] *Stabiliser states are efficiently PAC learnable*, Quantum Information and Computation, Vol. 18, No. 7&8 (2018).

2. [KRS18] *Learning DNFs under product distributions via $\mu$-biased quantum Fourier sampling*, arXiv preprint arXiv:1802.05690 (2018). With V. Kanade and S. Severini.

3. [Cil+18] *Quantum machine learning: a classical perspective*, Proceedings of the Royal Society A 474, No. 2209 (2018). With C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, S. Severini, and L. Wossnig.

4. [Roc+18] *Learning hard quantum distributions with variational autoencoders*, npj Quantum Information, 4 (2018). With G. Carleo, E. Grant, S. Severini, and S. Strelchuk.

5. [Roc+19] *Experimental learning of quantum states*, Science Advances 5, No. 3, eaau1946 (2019). With S. Aaronson, I. Agresti, M. Bentivegna, G. Carvacho, D. Poderini, and S. Severini.

I have additionally co-authored the following articles that are not included in this thesis:

1. [Ban+18] *Modelling non-Markovian quantum processes with recurrent neural networks*, New Journal of Physics, Vol. 20, No. 12 (2018). With L. Banchi, E. Grant, and S. Severini.

2. [Rud+18] *Approximating Hamiltonian dynamics with the Nyström method*, arXiv preprint arXiv: 1804.02484 (2018). With C. Ciliberto, M. Pontil, A. Rudi, S. Severini, and L. Wossnig.

3. [RBL16] *Stabilizers as a design tool for new forms of the Lechner-Hauke-Zoller annealer*, Science Advances 2, No. 10, e1601246 (2016). With S. Benjamin and Y. Li.

# Chapter 2

# Preliminaries

We introduce some notions from the theory of quantum computation and from the theory of learning. We only present concepts that are strictly relevant for our general discussion and leave introductions to Boolean analysis and generative models in chapter-specific preliminary sections.

We assume the reader is familiar with basic notions of computational complexity theory, quantum mechanics, and with the quantum circuit model. For a comprehensive introduction to quantum computation we refer the reader to the textbooks by Kitaev, Shen, and Vyalyi [Kit+02] and Nielsen and Chuang [NC10]. For learning theory we refer the reader to the textbooks by Kearns and Vazirani [KV94] and Shalev-Shwartz and Ben-David [SB14].

We will make repeated use of standard definitions and results from group theory and measure concentration. For convenience we list relevant concepts from both fields in Appendix A and B.

## 2.1 Notation

We denote vectors with lower-case letters. For a vector $x \in \mathbb{R}^n$, let $x_i$ denote the $i$-th element of $x$. If $x$ is sparse we can describe it using only its non-zero coefficients. We call this the *succinct representation* of $x$. For an integer $k$, let $[k]$ denote the set $\{1, \ldots, k\}$. We use the following standard norms. The $\ell_0$ 'norm' $\|x\|_0 = |\{i \in [k] | x_i \neq 0\}|$, the $\ell_2$ norm $\|x\|_2 = \sqrt{\sum_{i\in[k]} x_i^2}$, and the $\ell_\infty$ norm $\|x\|_\infty = \max_{i\in[k]}\{|x_i|\}$.

Let $f : \mathbb{R} \to \mathbb{R}^+$ and $g : \mathbb{R} \to \mathbb{R}^+$. We use $f(n) = O(g(n))$ to indicate that the asymptotic scaling of $|f|$ is upper-bounded, up to a constant factor, by $g(n)$. If the bound is not asymptotically tight we write $f(n) = o(g(n))$. Similarly, $f(n) = \Omega(g(n))$ indicates that the asymptotic scaling of $|f|$ is lower-bounded, up to a constant factor, by $g$. If the bound is not asymptotically tight we write $f(n) = \omega(g(n))$. The notation

$f(n) = \Theta(g(n))$ indicates that $f$ is bounded both above and below by $g$ asymptotically. The notations $\tilde{O}(g(n))$ and $\tilde{\Omega}(g(n))$ hide logarithmic factors.

The probability that an event $E$ occurs is denoted by $\Pr[E]$. Given a set $A$ the indicator function $\mathbf{1}_A : A \to \{0, 1\}$ takes values $\mathbf{1}_A(x) = 0$ if $x \notin A$ and $\mathbf{1}_A(x) = 1$ if $x \in A$. Let $X$ be a continuous or discrete random variable. By abuse of notation, we write $\mathcal{P}_X(X = x)$ to indicate both the *probability density function* (pdf) of the distribution at $x$ (for continuous variables) and the probability that $X = x$ (for discrete variables). For a realisation $x$ we often write $\mathcal{P}_X(x)$ rather than $\mathcal{P}_X(X = x)$. We continue the presentation assuming $X$ is a continuous variable as the notation extends straightforwardly to the discrete case. If a pdf $\mathcal{P}_X$ depends on a parameter $\mu$ we write $\mathcal{P}_X(x; \mu)$. For convenience we often drop the subscript and write the pdf as $\mathcal{P}(x)$. Note that using this notation $\mathcal{P}(x)$ and $\mathcal{P}(y)$ denote different pdfs, referring to two different random variables $X$ and $Y$, respectively. The notation $X \sim \mathcal{P}$ indicates that the random variable has pdf $\mathcal{P}$ while the notation $x \sim \mathcal{P}$ indicates that $x$ is sampled according to $\mathcal{P}$. For a finite sequence $S = (x^{(1)}, \ldots, x^{(n)})$, $S \sim \mathcal{P}^n$ indicates that the sequence $S$ is *independent and identically distributed* (i.i.d.) according to $\mathcal{P}$. The expected value of a random variable $f(X)$ is denoted as $\mathbb{E}_{X\sim\mathcal{P}}[f(X)] = \int f(x)d\mathcal{P}(x) = \int f(x)\mathcal{P}(x)dx$, where we assumed that $X$ has density $\mathcal{P}$ with respect to the Lebesgue measure. When $\mathcal{P}$ is the uniform distribution we omit the distribution in the subscript and write $\mathbb{E}[\,\cdot\,]$. We often use $\mathbb{E}_{\mathcal{P}}[\,\cdot\,]$ to indicate $\mathbb{E}_{X\sim\mathcal{P}}[\,\cdot\,]$. By abuse of notation, and only when there is only a single random variable involved in the discussion, we write $\mathbb{E}_{\mu}[\,\cdot\,]$ to indicate $\mathbb{E}_{X\sim\mathcal{P}(\cdot;\mu)}[\,\cdot\,]$. We use similar notation for Pr.

## 2.2 Learning theory

Let $\mathcal{X}$ and $\mathcal{Y}$ be sets and let $\mathcal{P}$ be a probability distribution over $\mathcal{X} \times \mathcal{Y}$. $\mathcal{X}$ is called the *domain set* and $\mathcal{Y}$ the *label set*. A *training set* $T$ is a finite sequence $T = ((x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)}))$ of i.i.d samples from $\mathcal{P}$ and we often refer to elements of $T$ as *examples*. In the task of supervised learning, a learning algorithm $\mathcal{A}$ receives a training set $T$ and returns a function $h$, known as *hypothesis*. The set of functions $\mathcal{H} = \{f : \mathcal{X} \to \mathcal{Y}\}$ that $\mathcal{A}$ can output is called the *hypothesis space*. The goal of learning is to produce an hypothesis $h$ that is good at predicting labels $y \in \mathcal{Y}$ of unseen points $x \in \mathcal{X}$ with respect to some given loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ that measures how far $h(x)$ is from the true label. Depending on whether the label set $\mathcal{Y}$ is continuous or discrete the task is called *regression* (continuous) or *classification*

(discrete). A common loss function for classification tasks where $\mathcal{Y} = \{-1, 1\}$ is the $0-1$ loss $\ell_{0-1}(h(x), y) = \mathbf{1}_{h(x)\neq y}$. A common loss function for regression tasks where $\mathcal{Y} = \mathbb{R}$ is the quadratic loss $\ell_{sq}(h(x), y) = (h(x) - y)^2$. Formally, a supervised learning algorithm seeks to minimise the following quantity known as the *risk* or *generalisation error*

$$R(h) := \mathbb{E}_{(X,Y)\sim\mathcal{P}}[\ell(h(X), Y)]. \tag{2.1}$$

Because $\mathcal{P}$ is unknown to the learner, it is not possible to compute the true risk. One common way of addressing this problem is by approximating the true risk with an empirical estimate known as the *empirical risk* $\hat{R}(h) := \frac{1}{m}\sum_{(x,y)\in T}\ell(h(x), y)$.

In the learning paradigm known as *empirical risk minimisation* (ERM) the learner outputs an hypothesis $h_T$ that minimises $\hat{R}(h)$

$$h_T = \arg\min_{h\in\mathcal{H}} \hat{R}(h) := \arg\min_{h\in\mathcal{H}} \frac{1}{m}\sum_{(x,y)\in T}\ell(h(x), y). \tag{2.2}$$

If the size of the training set is sufficiently large it is possible to prove, under suitable assumptions on the hypothesis space, that the empirical risk will lead to an hypothesis that minimises the true risk. This is the core intuition of the *statistical learning* model developed by Vapnik and Chervonenkis [Vap13]. One problem with this approach is that if the hypothesis space is very large, the minimisation of the empirical risk might produce an hypothesis that describes well the training set but performs poorly on the true risk. This phenomenon is known as *overfitting* and can be avoided by restricting the size of the hypothesis space $\mathcal{H}$. For this reason, the generalisation bounds of statistical learning theory are given in terms of mathematical measures of the complexity of the hypothesis space. In general, ERM analyses produce bounds of the difference between true and empirical risk $|R(h_T) - \hat{R}(h_T)| \leq O(\sqrt{c/m})$, where $c$ is a measure of the complexity of $\mathcal{H}$ such as the Vapnik-Chervonenkis (VC) dimension [Vap13], the fat shattering dimension [BLW96], the Rademacher complexity [BM02], or the covering numbers [AB09].

ERM is not the only paradigm for analysing the generalisation capability of a learning model. Other approaches include regularisation techniques that limit the capacity of the hypothesis space (either explicitly [BPR07] or by limiting the computational cost [RCR15]), compression bounds [LW86], or probably approximately correct-Bayesian bounds [McA99]. One common feature of all these methods is to limit the expressive power of the learned estimator in order to avoid overfitting the training dataset.

### 2.2.1 The probably approximately correct model

Starting from a statistical learning analysis of supervised learning problems, Valiant developed the *probably approximately correct* (PAC) model [Val84]. While statistical learning theory is only concerned with the information-theoretic side of learning (i.e. how many samples?), the PAC model also considers the computational aspects of the problem (i.e. what is the runtime of the best learning algorithm?). We present the PAC model for a label set $\mathcal{Y} = \{-1, 1\}$ and assume that there exists a target function $f \in \mathcal{H}$ such that $R(f) = 0$ (it follows that it suffices to consider distributions $\mathcal{P}$ over $\mathcal{X}$ instead than $\mathcal{X} \times \mathcal{Y}$). The latter assumption is known as the *realisability assumption* and will be used for all the results presented in this paper that concern the PAC model. Note that the realisability assumption implies that with probability 1 the empirical risk for a ERM hypothesis is $\hat{R}(h_T) = 0$. Furthermore, note that when $\mathcal{Y} = \{-1, 1\}$ the risk over the $\ell_{0-1}$ loss becomes $R(h) = \Pr_{x\sim\mathcal{P}}[f(x) \neq h(x)]$.

In the PAC model we use the ERM framework and give bound to the generalisation performance of an hypothesis $h_T$. In particular, we bound the minimum size of the training set such that $R(h_T) \leq \epsilon$, where $\epsilon$ is known as the *accuracy parameter*. Note that because the training set is randomly sampled from $\mathcal{P}$, we can treat $R(h_T)$ as a random variable and introduce the probability $\delta$, known as the *confidence parameter*, of sampling a training set that is unrepresentative of the underlying distribution. More formally, we are interested in lower bounding the probability to sample training sets that will lead to the success of the learner

$$\Pr_{T\sim\mathcal{P}^n}\left[R(h_T) \leq \epsilon\right] \geq 1 - \delta. \tag{2.3}$$

Using these notions we can define the concept of PAC learnability. It is convenient to introduce the *example oracle* $\mathrm{EX}(f, \mathcal{P})$. A call to $\mathrm{EX}(f, \mathcal{P})$ returns an example $(x, f(x))$ where $x$ is randomly sampled from $\mathcal{P}$.

**Definition 1.** *A hypothesis class $\mathcal{H}$ is PAC learnable with respect to a loss function $\ell$ if there exists a learning algorithm $\mathcal{A}$ that for every $\epsilon, \delta \in (0, 1)$, every target function $f \in \mathcal{H}$ and every distribution $\mathcal{P}$ over $\mathcal{X}$ with access to oracle $\mathrm{EX}(f, \mathcal{P})$ takes as input $m$ i.i.d. examples generated by $\mathrm{EX}(f, \mathcal{P})$ and outputs a hypothesis $h$ that satisfies with probability at least $1 - \delta$*

$$R(h(x), f(x)) = \mathbb{E}_{x\sim\mathcal{P}}\left[\ell(h(x), f(x))\right] \leq \epsilon.$$

*Furthermore, the running time of the algorithm $\mathcal{A}$ must be bounded by a polynomial in $m$, $1/\epsilon$, and $1/\delta$ and the output hypothesis, $h$, is required to be polynomially evaluable.*

The minimum $m$ such that the class is PAC learnable is known as *sample complexity*. The runtime of the best learner for the class is known as *time complexity*.

The bounds of the PAC model are known to be *distribution free* that is, they are independent of the underlying probability distribution. This is a conservative requirement and it is common that, in order to prove the efficient learnability of an hypothesis class, we either relax this assumption (i.e. we study the learnability under a restricted class of probability distributions) or we introduce gadgets that aid the learner. Two extensions of the PAC model are relevant for our purposes. In the *membership query* (MQ) model the learner has access, in addition to the example oracle $\mathrm{EX}(f, \mathcal{P})$, to a membership oracle $\mathrm{MQ}(f)$ that on input $x$ returns $f(x)$. In the *quantum PAC* model, the examples are given by a *quantum example oracle* $\mathrm{QEX}(f, \mathcal{P})$ that returns the superposition $\sum_x \sqrt{\mathcal{P}(x)}\,|x, f(x)\rangle$. We study these models in greater detail in Chapter 5.

## 2.3 Quantum computation

We often work in the *computational basis* $\{|i\rangle\}$ where, for an $n$-qubit system, each basis element corresponds to an $n$-bit string. A single qubit system can take two values $|0\rangle$ and $|1\rangle$. When working on the Boolean hypercube $\{-1, 1\}^n$ we take $0 \equiv -1$ and $1 \equiv 1$. A *quantum register* is a collection of qubits. Given a Boolean–valued function $f : \{-1, 1\}^n \to \{-1, 1\}$ a *quantum membership oracle* $O_f$ is a unitary map that applied on $n + 1$ qubits acts as follows $O_f : |i\rangle\,|0\rangle \to |i\rangle\,|f(i)\rangle$. By combining a membership oracle with the Hadamard transform it is possible to make a phase query $|i\rangle\,|-\rangle \to (-1)^{f(i)}\,|i\rangle\,|-\rangle$, where $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. This operation is also known as *phase kickback*. For ease of notation, in the following we will not write explicitly the ancilla register $|-\rangle$.

Let $x \in \{0, 1\}^n$, the quantum Fourier transform over $\mathbb{Z}_2^n$ is defined as $H^{\otimes n}\,|x\rangle = 2^{-n/2} \sum_{a \in \{0,1\}^n} (-1)^{x \cdot a}\,|a\rangle$, where $H$ is the Hadamard gate

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}.$$

### 2.3.1 Classical simulation

There are multiple ways of defining the notion of a classical simulation of a quantum computation. Different notions are based on different types of access to the output of

the circuit (sampling or direct) and different types of error (multiplicative or additive). The major definitions are reviewed in Section 2 of [BJS10].

For our purposes it suffices to distinguish between *weak simulation* and *strong simulation*. Note that we do not discuss the issues of accuracy in this work. Let $C$ be an $n$-qubit polynomial-size quantum circuit from a uniform family and let $P(i) = |\langle i| C |0\rangle|^2$ be the probability distribution induced when measuring $C|0\rangle$ in the computational basis.

**Definition 2.** *(Weak classical simulation) We say that $\mathcal{C}$ is weak simulable if there exists a classical algorithm that generates samples from $P(i)$ in time $O(\mathsf{poly}(n))$.*

Weak simulation is the notion that most closely resembles the idea of reproducing the functioning of a quantum computer on a classical machine. Strong simulation sets stricter requirements.

**Definition 3.** *(Strong classical simulation) We say that $\mathcal{C}$ is strong simulable if there exists a classical algorithm that computes $P(i)$ in time $O(\mathsf{poly}(n))$.*

For general quantum circuits the problem of strong classical simulation is known to be #P-hard [Nes08; Rud09; Mon17b]. For some families of circuits such as Clifford [AG04] or matchgates [Val02] there exist efficient strong simulation algorithms.

### 2.3.2 The Pauli group

The Pauli matrices are

$$X = \sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \text{ and } Z = \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Every Pauli matrix is Hermitian, traceless and unitary. We define the Pauli group $\mathcal{P}_n$ of $n$-qubit as

$$\mathcal{P}_n = \{\pm 1, \pm i\} \cdot \{I, X, Y, Z\}^{\otimes n}.$$

A general Pauli operator can be written, for example, as $P = X \otimes Z \otimes Z \otimes Y$ but in the following we omit the tensor product signs and write $P = XZZY$. The algebraic structure of the Pauli group is determined by the commutation relation

$$XZ = -ZX,$$

so for every $P, Q \in \mathcal{P}_n$ they either commute $[P, Q] = 0$ or anticommute $\{P, Q\} = 0$.

# Chapter 3

# Efficient learning of quantum states

Properties of quantum information such as the exponential scaling of the wave function, non-local effects, and the probabilistic nature of the measurement process, make the task of finding classical descriptions of quantum states challenging and theoretically rich.

A well studied problem is to quantify the resources required to give a complete description of a quantum state. The task is known in the literature as *quantum state tomography* and for general $n$-qubit quantum states it requires $\Omega\left(\exp(n)\right)$ measurements [Haa+17]. In order for a set of measurement to be used for quantum state tomography, it must form a basis on the Hilbert space of the system. Any such set is known as a *tomographically complete set*.

By using prior information on the nature of the state, such as its entanglement structure, it is possible to circumvent these bounds and obtain efficient tomographic schemes. For example, knowing that the state is low-rank [Gro+10; OW16], or that it is well described by a matrix product state with low bond dimension [Cra+10], can reduce the computational cost of the procedure to polynomial in the number of qubits.

When the observational data do not come from a tomographically complete set or the measurements are noisy, the problem becomes to determine what is the best description that can be obtained from partial information. This is an inference problem whose answer depends on the modelling assumptions. A popular approach in quantum information, albeit with several shortcomings, is maximum likelihood estimation [Hra97; Ban+99]. Other methods include Bayesian mean estimation [Blu10] and Jaynes' maximum entropy principle [Jay57; Bužˇ04].

In this chapter, we discuss the problem of inference from an incomplete set of measurements from the perspective of learning theory. Differently from the results mentioned in the previous paragraph, the focus of the learning problem is not to find

the best state consistent with the data, but to find the description that is best at predicting unseen measurements.

## 3.1 Overview of our results

The first to analyse the problem of learning quantum states under the lens of computational learning theory, and specifically the PAC model, was Aaronson [Aar07]. This work proved an interesting fact: if the goodness of learning is measured against the same distribution that generates the data, as it happens in the PAC model, then a set of measurements with a number of elements that grows linearly with the number of qubits can provide sufficient information to accomplish the task. This might appear in contradiction with the exponential lower bounds of quantum state tomography. The inconsistency is readily resolved by noting that we are asked to make predictions only on measurements that are sampled from the same distribution of the training set. This implies that a good hypothesis might be far from the true state in terms of a standard quantum distance such as fidelity but it might generalise well in the probabilistic sense defined in learning theory. In order to see this let us consider the case of an unknown quantum state that we wish to characterise with a PAC learning scheme. If no prior information is available it is reasonable to use measurements sampled uniformly at random from a tomographically complete set. But under a uniform distribution any quantum state is exponentially difficult to distinguish from the completely mixed state and therefore a PAC scheme will output as hypothesis the completely mixed state without making any measurements at all. The completely mixed state will be far in fidelity from the true state but will be a good predictor in a PAC theoretic sense.

Aaronson bounded the sample complexity of PAC learning quantum states but left open the problem of the computational resources required to produce the hypothesis. Mathematically, finding such a state corresponds to solving a semidefinite programming (SDP) feasibility problem (recall that the *feasible region* of an optimisation problem is the set of points that satisfies the problem's constraints). SDPs are well studied and can be solved efficiently in the dimension of the input matrices using interior point methods [NN94; Ali95], the ellipsoid method [GLS12] or algorithms specifically designed for the problem of learning quantum states [Haz08]. However, because for quantum states the dimensions of the input matrices scale exponentially with the system size, in general, quantum states are not efficiently learnable.

In this chapter, we analyse the problem of whether prior information can be used to find polynomial time algorithms for learning quantum states. We answer

this question affirmatively and show that the class of stabiliser states under Pauli measurements is efficiently PAC learnable. Stabiliser states are an important family of highly symmetrical quantum states that have important applications in quantum error correction.

We also develop an experimentally testable model of the learning theorem proved in [Aar07]. In particular, we study the learnability of noisy GHZ states under two non-trivial probability distributions. By means of numerical simulations we check the consistency of the model and estimate a relevant scaling constant. Finally, we test the model on a photonic platform including up to 6 qubits.

## 3.2 Related work

It is important to remark that the goals of quantum state tomography and learning quantum states are different. In tomography one has control of the measurements that are being performed and seeks to optimise them in order to find a complete description of a quantum state using the least possible number of measurements. Tomography has proven exponential upper [OW16; Haa+17] and lower [Haa+17] bounds on the number of copies of the state required to achieve the task.

In learning, instead, the goal is to bound the number of measurements that let the empirical risk generalise to the true risk. But because the risk is an expectation of the loss over the distribution that generates the training set, a good hypothesis must only be good with respect to measurements that are 'similar' to each other, where the notion of similarity is modelled by the learning distribution. If there is no similarity in the training set, i.e. the distribution is uniform, and the learner receives only polynomially many samples, then the PAC learning algorithm will most likely output the completely mixed state as hypothesis. Predictions from the completely mixed state will generalise well to unseen measurements but in terms of fidelity they will be far from the true state.

The PAC model is not the only way to formalise the notion of learnability of quantum states. Most approaches in the quantum information literature have focused on 'active learners', i.e. learners that can actively choose a set of measurements that maximise the probability of reconstructing the state. This is different from a PAC scenario where the goal is not to identify a particular state but to predict the outcome of a measurement randomly sampled from an unknown probability distribution based only on information contained in the training set. Examples of active learning results are the works of Aaronson and Gottesman [AG08] and of Montanaro [Mon17a]. Both

approaches show that it is possible to classify an $n$-qubit quantum state, assumed to be a stabiliser, using only $O(n)$ of its copies. Similarly, Low focused on determining an unknown element of the Clifford group [Low09] while Zhao, Pérez–Delgado and Fitzsimons tackled the problem of identifying an unknown graph state [ZPF16]. Note that every graph can be interpreted as a stabiliser state.

Other models of learning quantum states that have been introduced are the 'shadow tomography' problem [Aar18] and the 'online learning' problem [Aar+18]. In the first, the learner must predict the expected values of all the measurements contained in a given set and performed on an unknown quantum state. The learner can perform any of the measurements at a cost of using one copy of the state. The goal of the shadow tomography problem is to minimise the number of copies that are required to approximate all the expected values for all the measurements. Aaronson proved that even in this case the number of copies required to accomplish the task scales linearly in the number of qubits.

In the second, that can be seen as a generalisation of the PAC learnability results, the examples are coming sequentially and at every new measurement the learner outputs a prediction. Once a prediction has been filed the learner can output a new hypothesis. A further difference of the online learning model is that the examples are no longer assumed to be coming from i.i.d. samples of an unknown distribution and the sequence of measurements can even be chosen by an adversary. In this setting Aaronson et al. proved that the maximum number of erroneous predictions in a sequence can be upper bounded by $n$ [Aar+18].

Finally, the problem of determining the time complexity of learning quantum states can be analysed from the perspective of quantum computation. Can we broaden the class of efficiently learnable states using quantum resources? One way to address the question is by comparing classical and quantum SDP solvers.

Currently, the best classical SDP solvers run in polynomial time in the dimension of the input matrices [AK07; LSW15]. Brandão and Svore, building on the *multiplicative weights update* framework developed in [AK07], introduced a quantum SDP solver with a quadratic improvement in $n$ but with a worse dependency on the error parameters [BS17]. Better lower and upper bounds were given by van Apeldoorn et al. [Ape+17] and Brandão et al. [Bra+17]. In the same work Brandão et al. used a quantum input model (roughly speaking, a model in which the input matrices of the SDP are encoded as quantum states) to derive a quantum SDP solver with poly-logarithmic dependency in $n$. The main limitation of this approach is that it

introduces a dependency on the rank of the input matrices. In the problem of state learning this restricts the measurements to be described by low-rank POVMs.

The dependency on the rank was recently circumvented by van Apeldoorn and Gilyén [AG18]. Building on the framework developed in [Bra+17] the authors introduced, always under the assumption of the quantum input model, a quantum algorithm with poly-logarithmic dependence on $n$, no dependency on the rank, but a dependency on a normalisation factor for the input matrices. This factor is specific to the quantum input model and is equal to 1 if the input matrices have a natural correspondence with quantum states.

The algorithm by van Apeldoorn and Gilyén can be straightforwardly applied to learn efficiently quantum states and in fact the authors discuss the case of *shadow tomography*, a learning problem which presents the same SDP that arises in the learning framework of Aaronson [Aar07]. Determining the class of quantum states and measurements that satisfy the requirements of this quantum SDP solver is an open question.

Comparing classical and quantum algorithms for learning problems requires a careful analysis of how the data are presented to the algorithm. Such analysis was conducted by [Rud+18] and [Tan18] who independently showed that, assuming a classical input model where it is efficient to sample the rows of the input matrices in a way proportional to the row-norms, it is possible to derive classical algorithms with poly-logarithmic runtime for problems where the best quantum algorithms have also poly-logarithmic runtime. Using the same assumptions Chia et al. [Chi+19] derived a poly-logarithmic time SDP solver with polynomial dependence on the rank, thus matching the time complexity of the quantum algorithm by Brandão et al. [Bra+17]. We conclude by noting that, although under suitable assumptions classical and quantum learning algorithms can have the same polynomial asymptotic behaviour, at the moment the degree of the polynomials of the quantum algorithms is significantly lower.

## 3.3 Sample complexity of learning quantum states

The sample complexity of learning quantum states in the PAC model was studied by Aaronson in [Aar07]. In this section, we present his results discussing the case of learning quantum states under '*two-outcome*' POVM measurements. The extension to many-outcomes POVMs is also treated in [Aar07] and results in a worse scaling in the error parameters.

For convenience we remind the reader that a two-outcome POVM acting on an $n$ qubit quantum state is a set $E = \{E^{(1)}, E^{(2)}\}$, where each $E^{(j)}$ is a $2^n \times 2^n$ dimensional hermitian positive semidefinite matrix with eigenvalues in $[0, 1]$ and is known as a POVM element. The POVM elements satisfy $\sum_j E^{(j)} = I$ and the probability of measurement outcome $j$ is $p(j) = \mathrm{Tr}(E^{(j)}\rho)$. Because in the following we always consider only the first element $E^{(1)}$ of a POVM $E$ for ease of notation we refer to the first POVM element using the same symbol of the POVM that is, $E^{(1)} = E$.

Let $\mathcal{V}$ be a set of quantum states, $\mathcal{M}$ a set of POVMs, and $\mathcal{D}$ a probability distribution over $\mathcal{M}$. For every quantum state $\rho \in \mathcal{V}$ we define a function $q_\rho : \mathcal{M} \to [0, 1]$ that maps POVM measurements to the probability of obtaining the first outcome when measuring the state $\rho$

$$q_\rho(E) = \mathrm{Tr}(E\rho). \tag{3.1}$$

In the problem of PAC learning quantum state the function $q_\rho$ is the target function.

Given a set of quantum states $\mathcal{V}$ and measurements $\mathcal{M}$ we define the concept class $Q$ as the set of all functions $q_\rho$ corresponding to quantum states in $\mathcal{V}$:

$$Q(\mathcal{V}) = \{q_\rho\}_{\rho \in \mathcal{V}}.$$

Given a target quantum state $\rho$ let $\mathrm{EX}(q_\rho, \mathcal{D})$ be the example oracle that returns examples $(E, \mathrm{Tr}(E\rho))$ with probability $\mathcal{D}$. Aaronson proved the following theorem:

**Theorem 1.** *Let $\mathcal{V}$ be a set of $n$-qubit quantum states, let $\mathcal{M}$ be a set of $n$-qubit measurements operators, let $\mathcal{D}$ be a probability distribution over elements of $\mathcal{M}$, and let $\mathrm{EX}(q_\rho, \mathcal{D})$ be an example oracle. The concept class $Q = \{q_\rho\}_{\rho \in \mathcal{V}}$ is PAC-learnable. That is, keeping fixed the error parameters $\varepsilon, \eta, \gamma, \delta > 0$ with $\gamma\varepsilon \geq 7\eta$, for every target concept $q_\rho$ there exists an algorithm that, with probability $1 - \delta$ , when querying the example oracle*

$$m \geq m_Q = \frac{K}{\gamma^2\varepsilon^2}\left(\frac{n}{\gamma^2\varepsilon^2}\log^2\frac{1}{\gamma\varepsilon} + \log\frac{1}{\delta}\right)$$

*times, returns an hypothesis $q_\sigma$ such that:*

$$\Pr_{E\sim\mathcal{D}}\left[|q_\sigma(E) - q_\rho(E)| > \gamma\right] \leq \varepsilon.$$

We give a brief intuition of the proof. The proof combines two known results from quantum information and learning theory. The first step is to invoke a result by Bartlett and Long [BL98] that bounds the learnability of real-valued function in terms of a parameter called the *fat-shattering* dimension. The second step is to note

that the fat-shattering dimension of the hypothesis class $Q(\mathcal{V})$ can be interpreted as the minimum size of a quantum state that encodes, in a probabilistic fashion, a classical bit-string. Ambainis et al. [Amb+02] gave a precise bound to this quantity and Aaronson makes use of this result to bound the sample complexity of quantum states.

## 3.4 Time complexity of learning quantum states

Theorem 1 gives condition for when the minimisation of the empirical risk guarantees good generalisation performance. Based on this notion we can make explicit the conditions under which learning is also computationally efficient:

**Definition 4** (Condition for efficient learnability)**.** *Let $\mathcal{V}$ be a set of $n$-qubit quantum states, let $\mathcal{M}$ be a set of $n$-qubit measurements operators. The concept class $Q$ is efficiently PAC learnable if, for every target concept $q_\rho : \mathcal{M} \to [0,1]$ with $\rho \in \mathcal{V}$, keeping fixed the error parameter $\eta > 0$, there exists an algorithm $L$ running in* $\mathrm{poly}(n, 1/\eta)$ *that when querying the example oracle $m \geq m_Q$ times, generates a hypothesis state $\sigma$ that satisfies the following program:*

$$\begin{aligned} |\operatorname{Tr}(E_i\sigma) - \operatorname{Tr}(E_i\rho)| \leq \eta \quad \textit{for all } i \in [m], \\ \sigma \succeq 0, \\ \operatorname{Tr}(\sigma) = 1. \end{aligned} \tag{3.2}$$

*where by $\sigma \succeq 0$ we denote the positive semidefiniteness of $\sigma$.*

In general the problem of minimising the empirical risk for the learnability of quantum states amounts to solving an approximate SDP feasibility problem. These convex problems have solutions that are known to be efficiently computable in the dimensions of the input matrices. Standard algorithms include interior point methods [NN94; Ali95] or the ellipsoid method [GLS12] and there even exist solvers specifically designed for quantum state learnability problems [Haz08]. But in the case of quantum systems the dimensions of the input matrices scale exponentially with the system size and therefore PAC learning quantum states is, in general, a hard problem.

In the following sections, we show that the set of stabiliser states is PAC learnable under Pauli measurement. We begin by reviewing the theory of stabilisers.

## 3.5 Stabiliser states

A *stabiliser group* is an abelian subgroup of the Pauli group $\mathcal{P}_n$. A *stabiliser code* is the eigenspace of eigenvalue $+1$ of a stabiliser group. Every element in the stabiliser code is a *stabiliser state*. Stabiliser codes where introduced by Gottesman and play a central role in the theory of error correcting codes [Got96; Got97; Got98]. For self-contained and mathematical introductions to the theory and properties of stabilisers we recommend [GMC14; Haa16].

More specifically, we say that a vector $|\psi\rangle$ is *stabilised* by $P \in \mathcal{P}_n$ if $P|\psi\rangle = |\psi\rangle$. The vectors stabilised by all the elements of a subgroup $\mathcal{S}$ of $\mathcal{P}_n$ form a subspace $V_{\mathcal{S}}$. $\mathcal{S}$ is called the *stabiliser* of $V_{\mathcal{S}}$ whose size is $|V_{\mathcal{S}}| = 2^n/|\mathcal{S}|$. Every vector in $V_{\mathcal{S}}$ is a stabiliser state. When a stabiliser contains $2^n$ elements then $|V_{\mathcal{S}}| = 1$ and the state stabilised is unique.

The only vector stabilised by $-I$ and by two anticommuting operators $P$ or $Q$ is the zero vector (proof: $|\psi\rangle = PQ|\psi\rangle = -QP|\psi\rangle = -|\psi\rangle$). It is a known fact that in order for $\mathcal{S}$ to stabilise a non trivial subspace, then $\mathcal{S}$ must be Abelian and not include $-I$. This implies that $\mathcal{S}$ cannot contain elements with phase $\pm i$ (proof: if $iP \in \mathcal{S}$ then $(iP)^2 = -I$).

A stabiliser state can be efficiently represented by its generating set. An important result that makes use of this efficient representation is the Gottesman-Knill theorem [Got98]. The theorem proves that circuits composed by elements of the normaliser of the Pauli group, i.e. the Clifford group, can be simulated efficiently (in a strong sense) on a classical computer.

The density matrix of every stabiliser state can be expressed in terms of its stabilisers. In order to see that, first note that the operator $(I + S)/2$ when $S$ is a Pauli operator, is a projection onto the $+1$ eigenspace of $S$. Therefore if a stabiliser has generators $S_1, \ldots, S_n$ then the density matrix for that state is

$$\rho = \frac{1}{2^n}\prod_{i=1}^{n}(I + S_i) = \frac{1}{2^n}\sum_{a_1,\ldots,a_n\in\{0,1\}} S_1^{a_1}\cdots S_n^{a_n} = \frac{1}{2^n}\sum_{S\in\mathcal{S}} S. \tag{3.3}$$

When we do not have access to the full generating set but only to a subset $\mathcal{L}$ with dimension $|\mathcal{L}| = \ell < n$ we can still construct the projector to the corresponding subspace as $J = \frac{1}{2^\ell}\prod_{i=1}^{\ell}(I + S_i)$. In this case, however, the state is not pure and the density matrix corresponds to the projector up to a normalising constant. We thus get for $\ell < n$:

$$\rho = \frac{1}{2^n}\prod_{i=1}^{\ell}(I + S_i) = \frac{1}{2^n}\sum_{S\in\langle\mathcal{L}\rangle} S. \tag{3.4}$$

We note how this expression is still a valid quantum state because $\mathrm{Tr}(\rho) = 1$ and $\rho \succeq 0$ (proof: $\rho$ is equal to a projector up to a normalising constant).

We now prove an easy but useful lemma. In the following we assume that $\rho$ is a stabiliser state, $P_i$ is a Pauli measurement and $S_i$ a stabiliser of $\rho$. We construct the POVM elements $E_i^{(1)}$ and $E_i^{(2)}$ of the observable $P_i$ by noting that $E_i^{(1)} + E_i^{(2)} = I$ and $E_i^{(1)} - E_i^{(2)} = P_i$. The POVM element $E_i^{(1)}$ can be then written as $E_i^{(1)} = (I + P_i)/2$. Because we always take the first element $E_i^{(1)}$ of each POVM in the following we take $E_i^{(1)} = E_i$ and denote $E_i$ as the POVM associated to $P_i$.

**Lemma 1.** *Let $E = (I+P)/2$ be a POVM measurement associated to a Pauli operator $P$ and $\rho$ an $n$-qubit stabiliser state then $\mathrm{Tr}(E\rho)$ can only take the following values $\{0, 1/2, 1\}$ and:*

$$\begin{cases} \text{if } \mathrm{Tr}(E\rho) = 1 \text{ then } P \text{ is a stabiliser of } \rho; \\ \text{if } \mathrm{Tr}(E\rho) = 1/2 \text{ then neither } P \text{ nor } -P \text{ is a stabiliser of } \rho; \\ \text{if } \mathrm{Tr}(E\rho) = 0 \text{ then } -P \text{ is a stabiliser of } \rho. \end{cases}$$

*Proof.* By using the representation in Eq. 3.3 we can write $\mathrm{Tr}(E\rho) = \frac{1}{2^n} \mathrm{Tr}\left(\sum_{i=1}^{2^n} ES_i\right)$. Recalling that all Pauli matrices are traceless apart from the identity we obtain:

$$\mathrm{Tr}(E\rho) = \frac{1}{2^{n+1}} \left(2^n + \mathrm{Tr}\left(\sum_{S_i \in \mathcal{S} \setminus I} PS_i\right)\right).$$

The lemma follows by noting that $S_i^2 = I$ and $\mathrm{Tr}(S_i) = 0$ for every $S_i \neq I$ and by observing that because $S_i \neq S_j$ for every $i \neq j$ we can only have at most one non-zero element in the sum. □

## 3.6 Learning stabiliser states

In this section, we prove that stabiliser states under Pauli measurements are efficiently PAC learnable. The proof is constructive and gives an efficient algorithm for learning stabiliser states. The algorithm is composed of two subroutines 'Learning' (Algorithm 1) and 'Predictions' (Algorithm 2). In the Learning subroutine we construct an hypothesis state that minimises the program in Eq. 3.2. The construction is implicit (it would take exponential time to write down the hypothesis) and given in terms of a

generating set that is identified using information in the training set. In the Predictions subroutine we compute new predictions from the hypothesis. Both algorithms run in polynomial time in the number of qubits.

**Theorem 2** (Stabiliser states are efficiently PAC-learnable)**.** *Let $\mathcal{V}$ be the set of stabiliser states on n-qubits, let $\mathcal{M}$ be the set of measurements associated to the Pauli group, and let $\mathcal{D}$ be a probability distribution over elements of $\mathcal{M}$. The concept class $Q = \{q_\rho : \mathcal{M} \to [0,1]\}_{\rho \in \mathcal{V}}$ is efficiently PAC-learnable with respect to $\mathcal{D}$ and we say that the stabiliser states are efficiently PAC-learnable with respect to the Pauli group.*

*Proof.* We begin by proving that it is possible to efficiently identify a generating set in $T$. The constructive procedure is summarised in Algorithm 1. Let $T$ be a training set. Recall that for every measurement $E_i$ in $T$ such that $\mathrm{Tr}(E_i\rho) = 1$ there is an associated stabiliser element $P_i = 2E_i - I$. Thanks to Lemma 1 we can identify which measurements, if any, in $T$ correspond to a stabiliser measurement of the state. After the first stabiliser measurement has been identified, and placed on a list $\mathcal{L}$, Algorithm 1 checks whether any new $E_i$ such that $\mathrm{Tr}(E_i\rho) = 1$ can be generated from $\mathcal{L}$. At the end of the process the algorithm returns a list of independent generators $\mathcal{L} = \{S_1, \ldots, S_l\}$. Based on this information our knowledge of the state can be summarised in the following state:

$$\sigma = \frac{1}{2^n} \sum_{S_i \in \langle \mathcal{L} \rangle} S_i. \tag{3.5}$$

By using Lemma 1 it is easy to see how $\sigma$ respects all the inequalities in Eq. 3.2. Because the state is also a normalised projector we have that $\sigma \succeq 0$. Note that a simple sum of the known stabilisers would have also satisfied the inequalities in Eq. 3.2 but, in general, it would not be positive semidefinite.

It remains to give an efficient algorithm to determine whether a new example is independent of the list of generators $\mathcal{L}$ collected so far. The procedure is summarised in Algorithm 2. This is necessary to predict the value of $\mathrm{Tr}(E'_i\rho)$. We do that below using a variant of the check matrix method described in Section 10.5.1 of [NC10]. With this technique every element of $P \in \mathcal{P}_n$, where $P = P^1 \otimes \cdots \otimes P^n$, is mapped to a $2n+1$ dimensional row vector $r_P \in \{0,1\}^{2n+1}$. The vector $r_P$ is defined in the

**Algorithm 1** Learning

**Input:** training set $T = \{(E_i, \mathrm{Tr}(E_i\rho))\}_{i\in[m]}$ where $E_i = (P_i + I)/2$
**Output:** list of generators $\mathcal{L}$ contained in $T$

1: **for** $k = 1$ to $m$ **do**
2: **if** $\mathrm{Tr}(E_k\rho) = 1$ or $\mathrm{Tr}(E_k\rho) = 0$ and $E_k$ is not generated by $\mathcal{L}$ **then**
3: add $\mathrm{Tr}(P_k\rho)P_k$ to $\mathcal{L}$
4: **end if**
5: **end for**

following way:

$$r_P(0) = \begin{cases} 0 & \text{if} \quad \mathrm{sgn}(P) = +1 \\ 1 & \text{if} \quad \mathrm{sgn}(P) = -1 \end{cases}$$

$$r_P(i) = \begin{cases} 0 & \text{if} \quad P^i = Z \\ 1 & \text{if} \quad P^i \in \{X, Y\} \end{cases} \qquad \forall i \in \{1, \ldots, n\}$$

$$r_P(i) = \begin{cases} 0 & \text{if} \quad P^{i-n} = X \\ 1 & \text{if} \quad P^{i-n} \in \{Y, Z\} \end{cases} \qquad \forall i \in \{n+1, \ldots, 2n\},$$

where $\mathrm{sgn}(P') = +1$ if the overall sign of $P^1 \cdots P^n$ is positive and $\mathrm{sgn}(P') = -1$ otherwise. As an example,

$$-XYZY \to r(-XYZY) = [\,1 \,|\, 1\,1\,0\,1 \,|\, 0\,1\,1\,1\,]\,.$$

By checking whether the set of unsigned binary vectors $\{r_{S_1}, \ldots, r_{S_l}\}$ is linearly independent we can determine if the corresponding Pauli operators are also independent. We can use Gaussian elimination to perform this operation at a cost of $\mathcal{O}(n^3)$.

Algorithm 2 computes the expected value of a new measurement $E'$ using $\sigma$. Note that because from the generating set $\mathcal{L}$ we can construct up to $2^l$ elements we cannot write down the full hypothesis state $\sigma$ efficiently. But there is no need to construct this state explicitly. By using a technique developed by Aaronson and Gottesman to keep track of the evolution of a row vector [Got96; AG04] we can make use of the information contained in $\sigma$ using only the generators.

For every new measurement $E'$ we want to determine whether $E'$ commutes with the elements of $\mathcal{L}$ and whether it can be generated by $\mathcal{L}$. Both tasks can be accomplished efficiently using the check-matrix representation [NC10]. However, because the check matrix representation does not allow us to predict the sign, we are left with determining whether it is the operator $P'$ or $-P'$ that can be generated

**Algorithm 2** Predictions

**Input:** set of known stabiliser generators $\mathcal{L} = \{S_i\}$, new measurement $E' = (I + P')/2$
**Output:** prediction $\mathrm{Tr}(E'\rho)$

**if** $[P', S_i] = 0\ \forall i$ and $P'$ is generated by $\mathcal{L}$ **then**
    solve for $c_i$ equation $\sum_{i=1}^{l} c_i r_{S_i} = r_{P'}$ and determine $\mathrm{sgn}(P')$ with Eq. 3.6
    **if** $\mathrm{sgn}(P') = 1$ **then**
        $\mathrm{Tr}(E'\rho) = 1$
    **else if** $\mathrm{sgn}(P') = -1$ **then**
        $\mathrm{Tr}(E'\rho) = 0$
    **end if**
**else**
    $\mathrm{Tr}(E'\rho) = 1/2$
**end if**

with the elements of $\mathcal{L}$ (recall that $E' = (P' + I)/2$). This can be accomplished in the following way. Because in the check vector representation matrix multiplication between operators corresponds to addition modulo 2 and we know that $P'$ is generated by $\mathcal{L}$ we can write:

$$\sum_{i=1}^{l} c_i r_{S_i} = r_{P'}$$

where $c_i \in \{0, 1\}$ and the addition is done modulo 2. This corresponds to a system of linear equations that can be solved efficiently. Once we have found the right vector $c$ we can multiply the relevant operators (an efficient algorithm is described in [AG04]) to determine the sign:

$$\mathrm{sgn}(P') = \mathrm{sgn}(S_1^{c_1} \cdots S_n^{c_n}). \tag{3.6}$$

Algorithm 2 describes how to perform the prediction of the expected value of a new measurement $E'$.

The computational cost of Algorithm 1 and 2 is dominated by the cost of determining whether the stabiliser measurements in the training set are linearly independent. In the worst case scenario of a training set composed by $m$ stabiliser measurements, the linear independence must be checked $m$ times at a cost of $n^3$ per operation. Therefore, the overall time complexity of learning stabiliser states is $O(mn^3)$. □

We remark that Algorithms 1 and 2 are exact in the sense that the difference between the true and predicted value of the expected measurement outcomes is 0. For this reason there is no $\eta$ dependency in the running time of the learning algorithm.

## 3.7 Experimental learning of stabiliser states

In this section we develop an experimentally testable case of Theorem 1. We focus exclusively on the theoretical aspects of the problem and do not discuss the details of the experimental apparatus. These are presented in [Roc+19].

Our work addresses several experimentally relevant challenges. First, we identify a class of states, a class of measurements, and two learning distributions that can be implemented in a photonic system and under which learning is challenging. Second, we discuss how the experimental estimates of the expected values affect the learning theorem. Third, through numerical simulations we estimate a scaling constant that affects the number of experimental measurements to be performed.

Specifically, we study the learnability of a particular class of stabiliser states known as Greenberger-Horne-Zeilinger (GHZ) states [GHZ89]. We test the model on two photonic systems able to encode from 2 to 6 qubits. The experimental data show that noisy GHZ states are PAC learnable.

### 3.7.1 GHZ states and learning distributions

Although Theorem 1 can be applied to any state and under any probability distribution, it is interesting to test its predictions in non-trivial settings. If, for example, one were to take the uniform distribution over all possible measurement bases, with high probability no measurement drawn from this distribution would be able to distinguish the state from the completely mixed one (the expected value of an exponentially high fraction of the measurements would be equal to 1/2).

We choose to study the learnability of GHZ states. An $n$-qubit GHZ state is defined as

$$|\mathrm{GHZ}_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right) \tag{3.7}$$

Due to experimental noise, we are effectively testing the learnability of a mixed state and not of a perfect GHZ state. The advantage of using states that are close to GHZs, that are known to admit an efficient stabiliser representation, is the possibility of clearly identifying a set of measurements and a probability distribution that make the predictions of theorem 'interesting' in the sense that they cannot be reproduced using the completely mixed state as hypothesis.

We use two different experimental setups: the first generates states with up to 4 qubits and can be measured with every Pauli measurement, the second generates states with up to 6 qubits but can only be measured in $X$ and $Y$. We conisder different learning distributions for the two experimental setups, $\mathcal{D}_{(\mathrm{I})}$ and $\mathcal{D}_{(\mathrm{II})}$. The

distribution $\mathcal{D}_{(\mathrm{I})}$ is uniform over the set of stabiliser measurements of the GHZ state minus the identity matrix. The distribution $\mathcal{D}_{(\mathrm{II})}$ is uniform over the set of stabiliser measurements in $X$ and $Z$ of the GHZ state minus the identity matrix.

There are $2^n$ different stabilisers for an $n$-qubit stabiliser state. Because one of the stabilisers is always the identity (whose eigenvalue is 1 for every state) we chose not to include this measurement in those sampled by $\mathcal{D}$.

Under these distributions the completely mixed state is never a good hypothesis (unless $\gamma > 0.5$) because the stabiliser measurements performed on the GHZ state will always return 1 as an outcome. On the completely mixed state the same measurements will output 1 or 0 with equal probability.

### 3.7.2 Learning theorems and experimental data

In the case of learning with experimental data we have to take into account two factors that can invalidate Theorem 1: noise in the measurements and the lack of access to the true value of $\mathrm{Tr}(E\rho)$. Both issues can be positively addressed. We examine the noise problem first. As discussed in [Aar07], if the noise that corrupts $E$ to $E'$ is governed by a known probability distribution such as a Gaussian, then $E'$ is still just a POVM, so Theorem 1 applies directly. If the noise is adversarial, then we can also apply Theorem 1 directly, provided we have an upper bound on $|\mathrm{Tr}\,(E_i\rho) - \mathrm{Tr}\,(E'_i\rho)|$.

We now proceed to discuss the second issue. Theorem 1 is given in terms of expected values but in an experiment these values remain inaccessible and can only be estimated by averaging the outcomes of many measurements. We therefore use a variant of Theorem 1 that can be applied to single outcomes (Theorem 1.3 in [Aar07]).

**Theorem 3.** *Let $\rho$ be an $n$-qubit state, let $\mathcal{D}$ be a distribution over two-outcome measurements, and let $\mathcal{E} = (E_1, \ldots, E_m)$ consist of $m$ measurements drawn independently from $\mathcal{D}$. Suppose we are given bits $B = (b_1, \ldots, b_m)$, where each $b_i$ is 1 with independent probability $\mathrm{Tr}\,(E_i\rho)$ and 0 with probability $1 - \mathrm{Tr}\,(E_i\rho)$. Suppose also that we choose a hypothesis state $\sigma$ to minimize the quadratic functional*

$$f(\sigma) = \sum_{i=1}^{m} \left(\mathrm{Tr}\,(E_i\sigma) - b_i\right)^2. \tag{3.8}$$

*Then there exists a positive constant $K$ such that*

$$\Pr_{E\in\mathcal{D}}\left[|\mathrm{Tr}\,(E\sigma) - \mathrm{Tr}\,(E\rho)| > \gamma\right] \leq \varepsilon$$

*with probability at least $1 - \delta$ over $\mathcal{E}$ and $B$, provided that*

$$m \geq \frac{K}{\gamma^4\varepsilon^2}\left(\frac{n}{\gamma^4\varepsilon^2}\log^2\frac{1}{\gamma\varepsilon} + \log\frac{1}{\delta}\right). \tag{3.9}$$

**Algorithm 3** Find minimum $m$ that allows to PAC-learn $\rho$

**Input:** quantum state $\rho$, number of qubits $n$, distribution $\mathcal{D}_{(I)}$, error parameters $\epsilon, \gamma, \delta$, number of different training sets used for the estimate $i_{\text{MAX}}$
**Output:** minimum value of $m$ that satisfies the conditions of Theorem 1

$m = 1$
**repeat**
  $\delta_{est} = 0$
  **for** $i = 1 \dots i_{MAX}$ **do**
    Generate training set $T = \{(E_i, \text{Tr}(E_i\rho))\}_{i\in[m]}$ with random measurements drawn from $\mathcal{D}_{(I)}$
    $\sigma = \text{Hazan}(T, n)$
    **for** every $E \in \mathcal{V}$ **do**
      **if** $|\text{Tr}(E\sigma) - \text{Tr}(E_i\rho)| > \gamma$ **then**
        $\epsilon_{\text{est}} += 1/|\mathcal{V}|$
      **end if**
    **end for**
    **if** $\epsilon_{\text{est}} > \epsilon$ **then**
      $\delta_{est} += 1/i_{\text{MAX}}$
    **end if**
  **end for**
  $m = m + 1$
**until** $\delta_{est} < \delta$

For convenience, we work with the experimental estimates of the expected values and not at the level of single measurement outcomes. We now show that the Theorem 3 still holds if, rather than considering single measurement outcomes, we work with the estimated expected values of $\text{Tr}\left(E_i\rho\right) \approx \sum_{j=1}^{s} b_i^{(j)}/s$ where each $b_i^{(j)}$ is 1 with independent probability $\text{Tr}\left(E_i\rho\right)$ and 0 with probability $1 - \text{Tr}\left(E_i\rho\right)$. To establish the equivalence it suffices to show that the $\sigma$ that minimises $f = \sum_{i=1}^{m}(\text{Tr}(E_i\sigma) - b_i)^2$ also minimises $f' = \sum_{i=1}^{m'}\left(\text{Tr}\left(E_i\sigma\right) - \text{Tr}\left(E_i\rho\right)\right)^2$ where $m = m's$.

For an integer $s$, let $[s]$ denote the set $\{1, \dots, s\}$. If we assume that there exist $s$ different measurements $\{b_i^{(j)}\}_{j\in[s]}$ of each operator $E_i$ we can rewrite $f$ by grouping

together measurement outcomes that correspond to a single POVM:

$$\begin{aligned}\sum_{i=1}^{m}(\mathrm{Tr}(E_i\sigma)-b_i)^2 &= (\mathrm{Tr}(E_1\sigma)-b_1^{(1)})^2+(\mathrm{Tr}(E_1\sigma)-b_1^{(2)})^2+\cdots+(\mathrm{Tr}(E_{m'}\sigma)-b_{m'}^{(s)})^2\\ &= \sum_{i=1}^{m'}\left[s\left(\mathrm{Tr}(E_i\sigma)\right)^2+\sum_{j=1}^{s}(b_i^{(j)})^2-2\,\mathrm{Tr}(E_i\sigma)\sum_{j=1}^{s}b_i^{(j)}\right]\\ &= \sum_{i=1}^{m'}s\left[\left(\mathrm{Tr}(E_i\sigma)\right)^2+\sum_{j=1}^{s}(b_i^{(j)})^2/s-2\,\mathrm{Tr}(E_i\sigma)\sum_{j=1}^{s}b_i^{(j)}/s\right].\end{aligned}$$

Equivalently $f'$ can be expressed as:

$$\begin{aligned}\sum_{i=1}^{m'}\left(\mathrm{Tr}\left(E_i\sigma\right)-\mathrm{Tr}\left(E_i\rho\right)\right)^2 &= \sum_{i=1}^{m'}\left(\mathrm{Tr}\left(E_i\sigma\right)-\sum_{j=1}^{s}b_i^{(j)}/s\right)^2\\ &= \sum_{i=1}^{m'}\left[\left(\mathrm{Tr}\left(E_i\sigma\right)\right)^2+\left(\sum_{j=1}^{s}b_i^{(j)}/s\right)^2-2\,\mathrm{Tr}\left(E_i\sigma\right)\sum_{j=1}^{s}b_i^{(j)}/s\right].\end{aligned}$$

The minimum of $f(\sigma)$ is found for:

$$\frac{df(\sigma)}{d\sigma}=\sum_{i=1}^{m'}\left[\frac{d\,\mathrm{Tr}(E_i\sigma)^2}{d\sigma}-2\frac{d\,\mathrm{Tr}(E_i\sigma)}{d\sigma}\sum_{j=1}^{s}b_i^{(j)}/s\right]=0.$$

Equivalently, we get for $f'$:

$$\frac{df'(\sigma)}{d\sigma}=\sum_{i=1}^{m'}\left[\frac{d\,\mathrm{Tr}\left(E_i\sigma\right)^2}{d\sigma}-2\frac{d\,\mathrm{Tr}\left(E_i\sigma\right)}{d\sigma}\sum_{j=1}^{s}b_i^{(j)}/s\right]=0.$$

It is easy to see how $f$ and $f'$ are minimised by the same $\sigma$.

### 3.7.3 Numerical simulations

In order to identify the linear scaling of the sample complexity of quantum states we wish to estimate the minimum number of measurements $m$ that allows us to PAC-learn a state $\rho$ with accuracy parameters $\epsilon$, $\gamma$ and success probability $1-\delta$. Algorithm 3 is used for such estimate. At each iteration of $i$ the algorithm generates a set of measurements drawn from either $\mathcal{D}_{(I)}$ or $\mathcal{D}_{(II)}$. We give the pseudocode for the case of $\mathcal{D}_{(I)}$. The support of $\mathcal{D}_{(I)}$ is the set $\mathcal{V}$ of stabiliser measurements of the state minus the identity operator. Because each stabiliser state has $2^n$ stabiliser measurements we have $|\mathcal{V}|=2^n-1$. The case for $\mathcal{D}_{(II)}$ is identical apart for the support of $\mathcal{D}_{(II)}$ that is now the set $\mathcal{W}$ of the stabiliser measurements on $X$ and $Z$ of the state minus the identity operator.

**Algorithm 4** Hazan

**Input:** training set $T = \{(E_i, \mathrm{Tr}(E_i\rho))\}_{i\in[m]}$, Hilbert space dimension $N = 2^n$, and maximum number of iterations $k_{\mathrm{MAX}}$
**Output:** hypothesis state $\sigma$

1: Initialise $\sigma_0 = I/N$
2: **for** $k = 1$ **to** $k_{\mathrm{MAX}}$ **do**
3: **begin**
4: Compute the smallest eigenvector $v_k$ of $\nabla f(\sigma_k)$
5: Let $\alpha = \frac{1}{k}$
6: Update $\sigma_{k+1} = \sigma_k + \alpha_k(v_k v_k^T - \sigma_k)$
7: **end**

We minimise the function $f$ over the positive semidefinite matrices of unit trace with a variant of the Frank-Wolfe algorithm [FW56] developed by Hazan [Haz08]. All our simulations are performed using 300 iterations of the Hazan algorithm.

As discussed, the problem of learning quantum states can be cast as a convex program. In the formulation given in Eq. 3.8 the goal is to minimise the objective function $f(\sigma) = \sum_{i=1}^{m}(\mathrm{Tr}(E_i\sigma) - \mathrm{Tr}(E_i\rho))^2$ over the positive semidefinite matrices of unit trace. Because both the space of positive semidefinite matrices of unit trace and the objective function are convex, we are dealing with a constrained convex optimisation problem. A polynomial time algorithm for this class of problems is the Frank-Wolfe algorithm [FW56] for optimising a single function over the bounded positive semidefinite cone. In our simulations we use an extension of this work, developed by Hazan [Haz08], specifically designed for learning quantum states with the procedure described in Theorem 1. We note that Hazan's algorithm works for every quantum state and every set of measurements and does not exploit the GHZ structure of the states we discuss in this section.

We can compute analytically step 4 by using that $\frac{\partial \mathrm{Tr}(F(\mathbf{X}))}{\partial \mathbf{X}} = f(\mathbf{X})^T$, where $f$ is the scalar derivative of $F$, and the hermiticity of the measurement operators $E_i$

$$\begin{aligned}\nabla f(\sigma_k) &= \frac{\partial f(\sigma_k)}{\partial \sigma_k} \\ &= 2\sum_{i=1}^{m}(\mathrm{Tr}(E_i\sigma_k) - \mathrm{Tr}(E_i\rho))E_i^T \\ &= 2\sum_{i=1}^{m}(\mathrm{Tr}(E_i\sigma_k) - \mathrm{Tr}(E_i\rho))E_i.\end{aligned}$$

### 3.7.4 Experimental demonstration of GHZ learnability

There are several methods to produce GHZ states [Hua+11; Lei+04; Gao+10; Bar+05b; Bar+05a] in photonic systems. In order to scale up to 6 qubits we use two different approaches: the first one aims to increase the number of degrees of freedom per photon while the second one exploits an increasing number of photons. In setup (I) we generate 2-photon states, encoding up to 4 qubits, and perform a full set of measurements in the computational basis. In setup (II) we generate four-photon states, able to encode up to 6 qubits, but we are limited to only $X$ and $Z$ measurements.

We begin our experimental analysis with a full characterisation of the PAC-learnability of a 4-qubit GHZ state generated with setup (I). The complete set of measurements available with setup (I) allows us to compare the quality of the hypothesis $\sigma$ not only in terms of the learning theorem but also from a tomographic perspective. The results, presented in Fig. 3.1, show that, by increasing the number of measurements in the training set, the hypothesis $\sigma$ is getting closer, in terms of fidelity, to the ideal state and to the experimental state (right panel). In the same figure it is possible to see that the predictions (left panel, red dots) obtained by minimising $f(\sigma)$ are always better than those obtained by taking the completely mixed state (black line) as hypothesis. This confirms that the distributions we selected are 'interesting' from a learning perspective because it is not possible to make good predictions using random guessing.

Still using GHZ states generated from setup (I) we test the dependency of the measurement complexity on the error parameters $\epsilon$, $\delta$, $\gamma$. This kind of test is necessary in order to ensure that the hardness of the learning problem used in the experimental demonstration of the theorem is representative of a typical learning scenario. The numerical simulations on the scaling of the error parameters are shown in Fig. 3.2 and indicate that, as expected from Eq. 3.9, the hardness of the learning problem does not change abruptly with the error parameters (unless they introduce pathological cases; for example, for $\gamma > 0.5$ random guessing becomes a good prediction strategy).

We demonstrate the linear scaling of Theorem 1 over a GHZ of the type described in Eq. 3.7 and generated by exploiting setup (II). Our algorithm takes as input the error parameters $\epsilon, \gamma, \delta$ and, for a given $n$, outputs the minimum $m$ such that a training set that respects Eq. 3.2 is generated with probability $p = 1 - \delta$.

We present the results in Fig. 3.3 for both numerical and experimental data. The experimental data demonstrate that quantum states are PAC-learnable. A linear fit performed on the experimental data returns a slope value of 1.1. This implies that the value of the scaling constant $K$ in Eq. 3.9, left undetermined in Theorem 1,

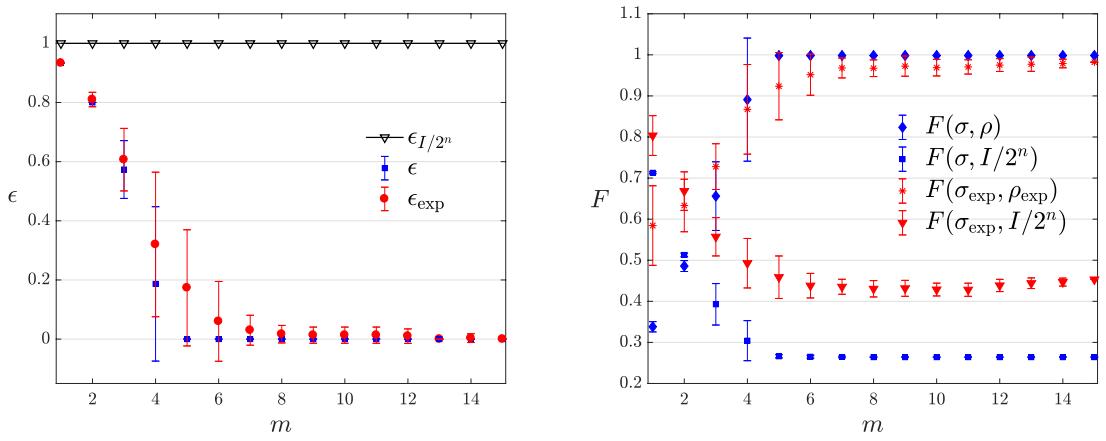



Figure 3.1: **Learning of a 4-qubit GHZ state.** Numerical simulations (blue curves) and experimental data (red curves) of the learning of the state $(|0000\rangle + |1111\rangle)/\sqrt{2}$. The subscript $[\,\cdot\,]_{\text{exp}}$ denotes experimental data. Quantities without subscript are obtained through numerical simulations. **(left)** The probability $\epsilon$ of predicting a measurement outcome with less than $\gamma = 0.1$ accuracy. The black line represents the predictions made using the completely mixed state as hypothesis. Clearly, the informed predictions are always better than a random guess. **(right)** The fidelity $F = \text{Tr}(\sqrt{\sigma^{1/2}\rho\sigma^{1/2}})$ between the hypothesis state $\sigma$ reconstructed by the PAC-learning algorithm and $\rho$ and between $\sigma$ and the completely mixed state $I/2^n$, that is, the starting guess of the optimisation algorithm. The learning distribution $\mathcal{D}_{(\text{I})}$ is uniform over the set of stabiliser measurements of the state minus the identity matrix. Error bars show the standard deviation for an average of 20 different, randomly generated, training sets.

is compatible with learning in an experimental setting. The values obtained from the linear fit in Fig. 3.3 show that learning a 20-qubit state would require $\sim 23$ measurements. Notice that a 20-qubit stabiliser state has 1048576 stabilisers and that the learning algorithm does not exploit the group structure of the state. In this sense the algorithm 'learns' that the state can be represented using only the generators of the group.

The high variance around $m = 4$ in Fig. 3.1 can be explained in the following way: each datapoint is obtained by averaging over a number of different configurations sampled from $\mathcal{D}_{(I)}$. It is then likely to sample a configuration that includes 2 generators and 2 other stabilisers that can be obtained by the product of the generators. It is easy to see how the information content of such a configuration is less than the one where 4 independent stabilisers are sampled. This will in turn limit the ability of $\sigma$ to output good predictions and will generate the high variance in the data.

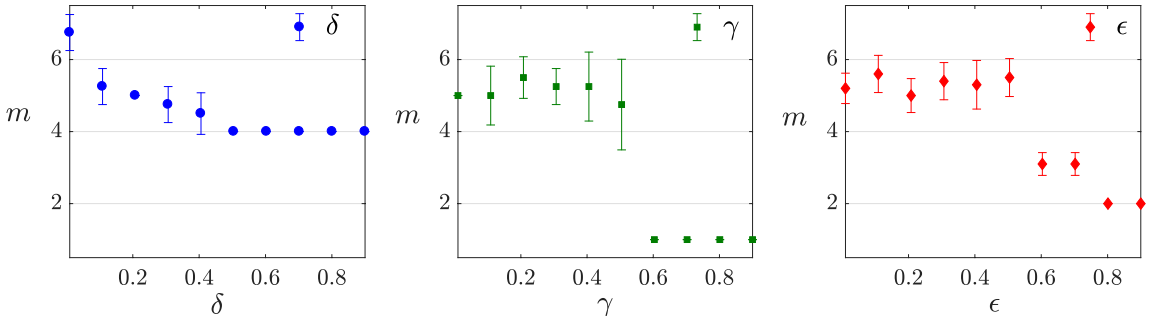



Figure 3.2: **Measurement complexity of error parameters.** Dependence of $m$ on the error parameters for learning 4-qubit GHZ states generated with setup (I). Learning is performed under the distribution $\mathcal{D}_{(\mathrm{I})}$ and each data-point is an average over 4 different GHZ states. When a given error parameter is changed the other ones are kept constant at the following values $\delta = 0.1$, $\gamma = 0.1$, and $\epsilon = 0.05$ **(left)** Scaling of $\delta$. **(center)** Scaling of $\gamma$. **(right)** Scaling of $\epsilon$.

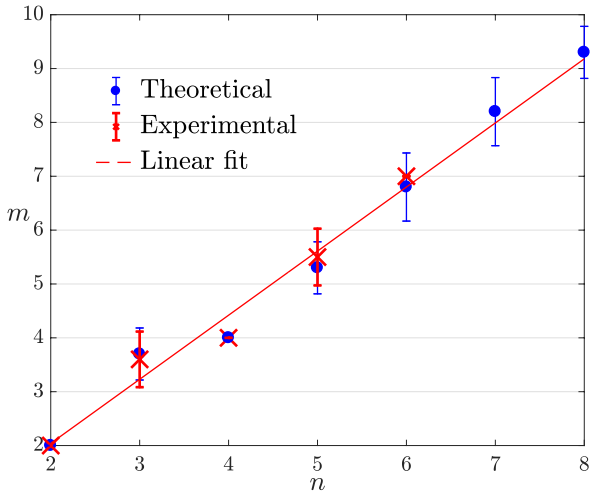



Figure 3.3: **Experimental demonstration of Theorem 1.** Scaling of size of the training set $m$ required to learn a GHZ state as a function of the number of qubits $n$. Experimental data-points (red crosses) are obtained using the experimental setup (II). Each data-point is obtained using 50 different, randomly generated sets of measurement configurations drawn from $\mathcal{D}_{(\mathrm{II})}$. Error bars show the standard deviation for an average of 10 different runs of the algorithm to estimate $m$. The red line is a linear fit on the experimental data-points with equation $m = 1.19n - 0.34$. The learning parameters are $\epsilon = 0.15$, $\gamma = 0.2$ and $\delta = 0.2$.

# Chapter 4

# Learning efficient representations of quantum states

The PAC learnability of stabiliser states discussed in Chapter 3 underscored the importance of finding an efficient representation of the elements in the hypothesis space in order to have computationally efficient learning algorithms. Whilst good representations are important for every learning task, they are particularly relevant in the case of quantum states where the number of parameters in the wave function scales exponentially with the number of particles in the system. In the case of stabilisers, describing the states in terms of the generators of the group provides an efficient description.

The task of finding efficient representations of quantum states is of interest not only in the context of learning but, more generally, in any quantum-mechanical calculation as the exponential scaling of the wave function makes hard to compute quantities of physical interest like ground state energies. This problem is commonly referred to in the literature as the *quantum many-body problem*. We recall that a representation of an $n$ qubit quantum state is efficient if it can be specified using $\mathsf{poly}(n)$ parameters.

Traditionally, the quantum many-body problem has been solved in particular settings leveraging a combination of approximation techniques (such as perturbation theory or semiclassical approaches) and the use of efficient representations that, by exploiting structural properties of the wave-function, can reduce the complexity of the problem. Standard techniques for tackling the quantum many-body problem are *quantum Monte Carlo* (QMC) methods, that allow one to sample exactly from many-body states free of the sign problem [NU98; GKW16; Suz93], and *tensor networks* (TN), that give efficient representation of low-dimensional states satisfying the area law for entanglement [VMC08; Orú14].

Finding an efficient representation of a quantum state is a difficult task as it requires an understanding, and the existence, of a structural property of the system, such as the entanglement structure for tensor networks, that can be leveraged by the representation. When no such property is known it is conceivable to use parametric representations that 'learn' from the data through an optimisation procedure. This is the strategy used by *variational methods*, a standard technique for computing ground state energies that employs a parametric form of the wave function that is commonly known as *ansatz*. The form of the ansatz is usually inspired by a physical property of the system that consequently limits the class of quantum states that can be represented efficiently. For example, the *density matrix renormalization group* is effective under the assumption that the state can be efficiently described by a matrix product state [Whi92; Sch11].

Representations developed for machine learning tasks, such as support vector machines or artificial neural networks, rely on minimal assumptions and provide architectures that are easy to optimise. For these reasons, machine learning methods have been recently introduced to tackle a variety of tasks in quantum information processing that involve the manipulation of quantum states. These techniques offer greater flexibility and, potentially, better performance, with respect to traditional methods.

## 4.1 Overview of our results

Research efforts have mainly focused on using the modelling flexibility and representational power of various forms of *artificial neural networks* (ANNs). Representations of quantum states based on neural networks are called *neural network quantum states*.

In this chapter, we discuss neural network based machine learning techniques that can find efficient representations of quantum states from measurement data. In particular, we focus on *unsupervised learning* methods that do not require a labelled training set and can be used to simulate the outcome of a measurement performed on the state. These types of algorithms are known as *generative models*.

Our main contribution is a novel representation of the probability distribution of a quantum state based on *variational autoencoders* (VAEs). A VAE is a type of generative model based on layered neural networks independently developed by Kingma and Welling [KW13] and Rezende, Mohamed, and Wierstra [RMW14]. We study the representational capabilities of VAEs and address several open questions with neural network quantum states. First, by means of numerical simulations we

study how the depth of the network affects the ability to compress quantum many-body states. More specifically, we benchmark the performance of deep networks on states where no efficient classical description is known, finding that depth systematically improves the quality of the reconstruction for states that can be efficiently constructed with a quantum computer. Surprisingly, the same does not apply for hard states that cannot be efficiently constructed by means of a quantum process. Here, depth does not improve the reconstruction accuracy. Our results suggest that neural networks are able to capture correlations in states that are provably hard to sample from for classical computers but not for quantum ones. This might signal, as discussed in [Lev+19], that unique quantum features such as entanglement are well represented by deep neural networks.

Second, we show that VAEs can learn efficient representations of computationally tractable states and can reduce the number of parameters required to represent an hard quantum state up to a constant factor. However low, this compression level might enable to characterise quantum states of a size expected on near term quantum devices.

## 4.2 Related work

Neural networks can be used to represent a quantum state $|\psi\rangle = \sum_i \psi_i |i\rangle$ in two different ways. First, in a supervised model a neural network learns from a training set of tuples $(|i\rangle, \psi_i)$ how to associate a basis element with a complex amplitude. Once the model is trained the network returns the amplitude $\psi_i$ for a given basis element $|i\rangle$. Second, in an unsupervised model the network learns from a set of measurement outcomes how to reproduce the corresponding measurement statistics (this setting is similar to quantum state tomography). In the simplest case the network sees measurements in only one basis and the goal is to approximate the corresponding probability distribution. If the state has only positive real entries representing the probability distribution corresponds to representing the whole state.

Neural network quantum states were introduced by Carleo and Troyer [CT17] using a type of neural network architecture known as *restricted Boltzmann machine* (RBM) [Smo86]. In a RBM the samples are modelled through a Boltzmann distribution of a pair-wise interacting Ising model and through training the model learns a set of couplings that reproduces the given measurement statistics. This clear physical interpretation of RBMs in terms of spin models contributed to make them a popular model for quantum systems.

RBMs have been successfully applied to a variety of physical problems, ranging from strongly correlated spins [CT17; DLD17], and fermions [Nom+17] to topological phases of matter [DLS16; Gla+17; KPB17]. Theoretical analysis of the representational power of RBMs has been conducted in a series of works [GD17; Che+17; HM17b; DLD17; Cla17]. Gao and Duan, in particular, showed that RBMs cannot efficiently encode every quantum state [GD17]. More specifically, they proved that deep Boltzmann machines (DBMs) with complex weights, a multilayer variant of RBMs, can efficiently represent most physical states. Although this result is of great theoretical interest, the practical application of complex-valued DBMs in the context of unsupervised learning has not yet been demonstrated due to a lack of efficient methods to sample efficiently from DBMs when the weights are complex-valued.

Particularly relevant to our purposes are the works by Torlai et al. [Tor+17] and Carrasquilla et al. [Car+19b]. Both papers develop tomographic schemes based on generative models that are then tested on states for which there are known efficient representations such as *matrix product states* (MPS) [Per+06]. Although it is notable that these models can learn an efficient representation solely from data, it remains unclear how they behave on states where no efficient classical description is available.

One of the aspects of neural network states investigated by our work is the role played by network depth in the representational capability of the model. In the machine learning literature several recent papers have shown that depth significantly improves the representational capability of networks for some classes of functions (such as compositional functions) [MLP16; Tel16; ES16]. All these results are based on showing that ad-hoc choices of the weights of a deeper network cannot be approximated by a shallow one. It remains unclear what is the role of depth in typical cases. Some preliminary work in this direction, supporting the notion that depth increases exponentially the expressivity of the network, has been conducted by Raghu et al. [Rag+17].

## 4.3 Generative modelling with VAEs

In the learning models discussed in the previous chapters we made no assumptions on the underlying data distribution and considered only the problem of inferring an accurate predictor for the labels. This approach is known as *discriminative learning*. Generative models take a different approach to classification problems and seek to directly model the underlying data distribution using a specific parametric form. The

learning task becomes to optimise the parameters of the model and it is known as *parametric density estimation*.

Generative models can also be applied in unsupervised settings, i.e. when the training points are not labelled. In this case the goal is to model an unknown probability distribution from a training set of sample data. More specifically, we are concerned with the following task: given an i.i.d. training set $T = \{x^{(1)}, \ldots, x^{(n)}\}$ sampled according to an unknown pdf, find a distribution that models the data. In addition, we seek models that can generate new samples from the distribution.

We assume a model $\mathcal{P}_X$ with parametric form specified by a parameter $\theta$ and define the log likelihood of $T$ as

$$\mathcal{L}(x^{(1)}, \ldots, x^{(n)}; \theta) = \log \mathcal{P}_X(x^{(1)}, \ldots, x^{(n)}; \theta) = \sum_{i=1}^{n} \log \mathcal{P}_X(x^{(i)}; \theta),$$

where we used the assumptions that the training set is i.i.d. There are two main ways of designing generative models. In the Bayesian approach the parameters of the model are treated as a random variable and Bayes' theorem is used to model the posterior distribution of the parameters given the data. For an introduction to a Bayesian perspective to inference problems we recommend the textbook by MacKay [Mac03]. In the frequentist approach, which we use in the following discussion, we seek a set of parameters that maximises the likelihood of seeing the training data. This approach is known as *maximum likelihood estimation* (MLE). Popular modern generative models that perform MLE are deep Boltzmann machines [SH09], flow-based generative models [DKB14], autoregressive models [OKK16], and VAEs [KW13].

In this chapter, we use VAEs to represent quantum states. VAEs were originally introduced in the context of variational inference as an alternative strategy to Markov chain Monte Carlo (MCMC) to approximate posterior densities for Bayesian models [KW13; RMW14]. Although VAEs were developed for a variational inference problem, they can be discussed entirely in the framework of MLE.

The starting point of a VAE model is to assume the data generating process involves an unobserved continuous random variable $Z$, known as a *latent variable*, with pdf $\mathcal{P}_Z$. $\mathcal{P}_Z$ is usually taken to be from a parametric family of distributions such that its density is differentiable almost everywhere with respect to some parameters $\theta$ and $z$. From a coding perspective it is possible to interpret the unobserved variable $Z$ as a latent space or code, $\mathcal{P}_{Z|X}$ as a probabilistic *encoder* and $\mathcal{P}_{X|Z}$ as a *decoder*, since given a value $z$ it outputs a distribution over the possible corresponding values of $X$.

A schematic of a VAE is shown in Fig. 4.1. In a VAE the encoder projects the input in the latent space and the decoder reconstructs the input from the latent

representation. Once the network is trained the encoder can be dropped and, by generating samples in the latent space, it is possible to sample according to the original distribution. In graph theoretic terms, the graph representing a network with a given number of layers is a *blow up* of a directed path on the same number of vertices. Such a graph is obtained by replacing each vertex of the path with an independent set of arbitrary but fixed size. The independent sets are then connected to form complete bipartite graphs.

A key element of the model is to determine the conditional distribution $\mathcal{P}_{Z|X}$. This can be achieved using Bayes' theorem

$$\mathcal{P}_{Z|X}(z|x;\theta) = \frac{\mathcal{P}_{X|Z}(x|z;\theta)\mathcal{P}_Z(z;\theta)}{\mathcal{P}_X(x;\theta)} = \frac{\mathcal{P}_{X|Z}(x|z;\theta)\mathcal{P}_Z(z;\theta)}{\int \mathcal{P}_{X,Z}(x,z;\theta)dz}.$$

The marginalisation over $z$ at the denominator is usually intractable because the space is too large. *Variational methods* solve this problem by approximating $\mathcal{P}_{Z|X}$ with a simpler distribution $\mathcal{Q}_{Z|X}$ parameterised by a set of parameters $\phi$ and seek to minimise a distance between $\mathcal{P}_{Z|X}$ and $\mathcal{Q}_{Z|X}$. It is customary to use the relative entropy between $\mathcal{P}_{Z|X}$ and $\mathcal{Q}_{Z|X}$ as distance between the distributions. We recall that *relative entropy* or *Kullback-Leibler divergence* between $\mathcal{P}_{Z|X}$ and $\mathcal{Q}_{Z|X}$ is defined as:

$$D_{KL}(\mathcal{P}_{Z|X}||\mathcal{Q}_{Z|X}) = \mathbb{E}_{Z\sim\mathcal{P}_{Z|X}}\left[\log\frac{\mathcal{P}_{Z|X}(Z|X)}{\mathcal{Q}_{Z|X}(Z|X)}\right].$$

The relative entropy satisfies the *Gibbs's inequality* $D_{KL}(\mathcal{P}_{Z|X}||\mathcal{Q}_{Z|X}) \geq 0$ where

$$D_{KL}(\mathcal{P}_{Z|X}||\mathcal{Q}_{Z|X}) = 0 \text{ if and only if } \mathcal{P}_{Z|X} = \mathcal{Q}_{Z|X}.$$

In general

$$D_{KL}(\mathcal{P}_{Z|X}||\mathcal{Q}_{Z|X}) \neq D_{KL}(\mathcal{Q}_{Z|X}||\mathcal{P}_{Z|X}).$$

We are now ready to analyse the likelihood of each datapoint $x^{(i)}$ individually. Recall that when a function $f$ is concave (as is the case for the logarithm), Jensen's inequality holds

$$f(\mathbb{E}_{X\sim\mathcal{P}_X}[X]) \geq \mathbb{E}_{X\sim\mathcal{P}_X}[f(X)]$$

We can apply Jensen's inequality to the density of each datapoint $x^{(i)}$

$$\begin{aligned}
\log \mathcal{P}_X(x^{(i)};\theta) &= \log \int \mathcal{P}_{X|Z}(x^{(i)}|z;\theta)\mathcal{P}_Z(z;\theta)dz \\
&= \log \int \mathcal{P}_{X|Z}(x^{(i)}|z;\theta)\mathcal{P}_Z(z;\theta)\frac{\mathcal{Q}_{Z|X}(z|x^{(i)};\phi)}{\mathcal{Q}_{Z|X}(z|x^{(i)};\phi)}dz \\
&= \log \mathbb{E}_{Z\sim\mathcal{Q}_{Z|X}}\left[\frac{\mathcal{P}_{X|Z}(x^{(i)}|Z;\theta)\mathcal{P}_Z(Z;\theta)}{\mathcal{Q}_{Z|X}(Z|x^{(i)};\phi)}\right] \\
&\geq \mathbb{E}_{Z\sim\mathcal{Q}_{Z|X}}\left[\log\frac{\mathcal{P}_{X|Z}(x^{(i)}|Z;\theta)\mathcal{P}_Z(Z;\theta)}{\mathcal{Q}_{Z|X}(Z|x^{(i)};\phi)}\right] \\
&= \mathbb{E}_{Z\sim\mathcal{Q}_{Z|X}}\left[\log\mathcal{P}_{X|Z}(x^{(i)}|Z;\theta)\right] - D_{KL}\left[\mathcal{Q}_{Z|X}||\mathcal{P}_Z\right] \\
&= J(\theta,\phi,x^{(i)}),
\end{aligned}$$

where $\log \mathcal{P}_X(x^{(i)};\theta) \geq J(\theta,\phi,x^{(i)})$ is known as the *evidence lower bound* (ELBO).

In variational inference the lower bound provided by the ELBO is used to maximise the log likelihood. The first term in the ELBO $\mathbb{E}_{Z\sim\mathcal{Q}_{Z|X}}[\log\mathcal{P}_{X|Z}(x^{(i)}|Z;\theta)]$, known as *reconstruction loss*, is the expected negative log-likelihood of the $i$-th data-point. It favours choices of $\theta$ and $\phi$ that lead to more faithful reconstructions of the input. The second term $D_{\mathrm{KL}}(\mathcal{Q}_{Z|X}||\mathcal{P}_Z)$ is known as *regularisation loss* and is the Kullback-Leibler divergence between the encoder's distribution $\mathcal{Q}_{Z|X}$ and the prior on $Z$. It can be interpreted as a regularisation term that pushes the approximate posterior to be close to the prior $\mathcal{P}_Z$.

In order to compute analytically the Kullback-Leibler divergence in the ELBO we take $\mathcal{P}_Z(z;\theta) = \mathcal{N}(z;0,I)$ to be a Gaussian with zero mean and unit variance and $\mathcal{Q}_{Z|X}(z|x^{(i)};\phi) = \mathcal{N}(z;\mu^{(i)},\sigma^{(i)}I)$ to be a Gaussian with mean $\mu^{(i)}$ and variance $\sqrt{\sigma^{(i)}}$. For simplicity we also take $\mathcal{P}_{X|Z}$ to be a multivariate Gaussian (in case of real-valued data) or Bernoulli (in case of binary data). The maximisation of the likelihood is then performed on the parameters of these distributions. The intuition of VAEs is to use the well developed optimisation machinery of neural networks to model the parameters of the distributions (both the encoder and the decoder). For example, for the posterior $Q_{Z|X}(z|x^{(i)};\phi) = \mathcal{N}(z;\mu_\phi,\sigma_\phi I)$ both $\mu_\phi$ and $\sigma_\phi$ are computed using fully connected feed-forward neural networks.

For completeness we remind that a fully connected feed-forward neural network (FFNN) with $L$ layers is a map $f_{NN}:\mathbb{R}^n\to\mathbb{R}^m$ given by

$$f_{NN}(x) = g_L\,\sigma\,g_{L-1}\,\sigma\cdots\sigma\,g_1(x)$$

with affine linear maps $g_l:\mathbb{R}^{r_{l-1}}\to\mathbb{R}^{r_l}$ defined via $g_l = W_l x + b_l$ with $W_l\in\mathbb{R}^{r_l\times r_{l-1}}$, $b_l\in\mathbb{R}^{r_l}$, $r_0 = n$, $r_L = m$ and $r_l\in\mathbb{N}$ for $1\leq l\leq L-1$ is called the *width* of the $i$-th

*hidden layer*. The number $L$ is called the *depth* of the network. The nonlinear function $\sigma$ acts component-wise and is called the *activation function*. Common activation functions are the rectified linear $\sigma(x) = max\{0, x\}$ (and the corresponding neurons are known as *rectified linear units* (ReLU)) and the hyperbolic tangent $\sigma(x) = \tanh(z)$. The textbook by Goodfellow, Begio, and Courville gives a thorough introduction to artificial neural networks [GBC16].

As is customary in machine learning, $J(\theta, \phi, x^{(i)})$ is maximised using gradient ascent. This requires computing the gradient of $J(\theta, \phi, x^{(i)})$. Assuming Gaussian forms for $\mathcal{Q}_{Z|X}$ and $\mathcal{P}_Z$, $D_{KL}\left[\mathcal{Q}_{Z|X}||\mathcal{P}_Z\right]$ can be usually computed in closed form. Computing the expectation term can be problematic as it might have very high variance when approximated with a standard Monte Carlo estimator [PBJ12]. The way VAEs circumvent the problem is by using the 'reparametrisation trick' which amounts to a change of variable using a differentiable transformation. More details are given in [KW13].

## 4.4 Encoding quantum probability distributions with VAEs

We use VAEs to encode the probability distribution arising from taking the square moduli of the amplitudes of a quantum state in a given basis. Let us consider an $n$-qubit quantum state $|\psi\rangle$, with respect to a basis $\{|b_i\rangle\}_{i=1,\dots,2^n}$. We can write the probability distribution corresponding to $|\psi\rangle$ as $p(b_i) = |\langle b_i|\psi\rangle|^2$. If we consider the computational basis, we can write $|\psi\rangle = \sum_{i=1}^{2^n} \psi_i |i\rangle$, where each basis element corresponds to an $n$-bit string. A VAE can be trained to generate basis elements $|i\rangle$ according to the probability $p(i) = |\langle i|\psi\rangle|^2 = |\psi_i|^2$.

We note that, in principle, it is possible to encode a full quantum state (phase included) in a VAE. This requires samples taken from more than one basis and a network structure that can distinguish among the different inputs.

We approximate the true posterior distribution across measurement outcomes in the latent space $Z$ with a multivariate Gaussian, having diagonal covariance structure, zero mean and unit standard deviation. The training set consists of a set of basis elements generated according to the distribution associated with a quantum state. Following training, the variables $Z$ are sampled from a multivariate Gaussian and used as the input to the decoder. By taking samples from this Gaussian as input, the decoder is able to generate strings corresponding to measurement outcomes that closely follow the distribution of measurement outcomes used to train the network.

We trained all our networks using the tensorflow r1.3 framework on a single NVIDIA K80 GPU. We trained using backpropagation and the Adam optimiser with initial learning rate of $10^{-3}$ [KB14]. All the hidden layers used leaky rectified linear units (LReLU) [MHN13]. A LReLU is equivalent to a standard ReLU for $x \geq 0$ but for $x \leq 0$ we take $y = \ell x$ where $\ell$ is the leak which we set to 0.2. For the final layer we used sigmoid activation functions.

Training involves optimising two objectives: the reconstruction loss and the regularisation loss. We used a warm up schedule on the regularisation objective by increasing a weight on the regularisation loss from 0 to 0.85 linearly during training [Søn+16]. This turned out to be critical, especially for hard states. A consequence of this approach is that the model does not learn the distribution until close to the end of training, irrespective of the number of training iterations. Each network was trained using $50,000$ batches of 1000 samples each. Each sample consists of a binary string representing a measurement outcome.

Following training, the state was reconstructed from the VAE decoder by drawing $100(2^n)$ samples from a multivariate Gaussian with zero mean and unit variance. The samples were decoded by the decoder to generate measurement outcomes in the form of binary strings. The relative frequency of each string was recorded and used to reconstruct the learned distribution which was compared to the true distribution to determine its fidelity.

In all experiments the number of nodes in the latent layer is the same as the number of qubits. Using fewer or more nodes in this layer resulted in worse performance. The number of nodes in the hidden layers is determined by the number of layers and the compression $C$ defined by $m/2^n$ where $n$ is the number of qubits and $m$ is the number of parameters in the decoder. In all cases the encoder has the same number of hidden layers and nodes in each layer as the decoder.

We compress the VAE representation of a quantum state by removing neurons from each hidden layer of the VAE. When $n$ is small achieving a high level of compression caused instabilities in the network (*i.e.* the reconstruction accuracy became more dependent on the weight initialisation). In this respect we note that, by restricting the number of neurons in the penultimate layer, we are effectively constraining the number of possible basis states that can be expressed in the output layer and, as a result, the number of configurations the VAE can sample from. This can be shown noting that the activation functions of the penultimate layer generate a set of linear inequalities that must be simultaneously satisfied. A geometric argument that involves how many regions of an $n$-dimensional space can be separated by $m$ hyperplanes leads us to

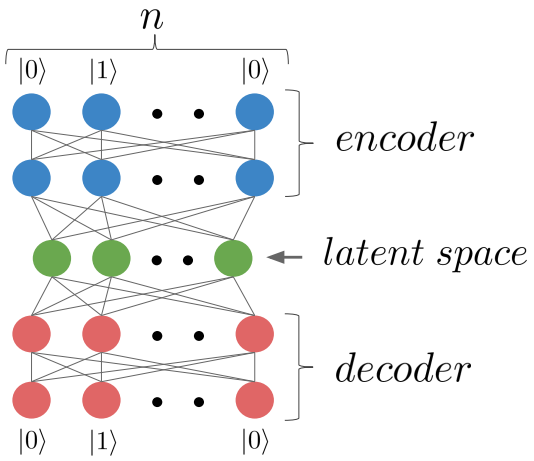



Figure 4.1: **Encoding quantum probability distributions with VAEs.** A VAE can be used to encode and then generate samples according to the probability distribution of a quantum state. Each dot corresponds to a neuron and neurons are arranged in layers. Input (top), latent, and output (bottom) layers contain $n$ neurons. The number of neurons in the other layers is a function of the compression and the depth. Layers are fully connected with each other with no intra layer connectivity. The network has three main components: the encoder (blue neurons), the latent space (green), and the decoder (red). Each edge of the network is labelled by a weight $\theta$. The total number of weights $m$ in the decoder corresponds to the number of parameters used to represent a quantum state. The network can approximate quantum states using $m < 2^n$ parameters. The model is trained using a dataset consisting of basis elements drawn according to the probability distribution of a quantum state. Elements of the basis are presented to the input layer on top of the encoder and, during the training phase, the weights of the network are optimised in order to reconstruct the same basis element in the output layer.

conclude that, to have full expressive capability, the penultimate layer must include at least $n$ neurons. Similar arguments have been discussed in [HH91] for multilayer perceptrons.

## 4.5 Hard and easy quantum states

In this section we introduce a method to classify quantum states based on the hardness of sampling their probability distribution in a given basis. We use this framework to assess the power of deep neural network models at representing quantum many-body wave-functions.

We now proceed to define two concepts that we will frequently use throughout the paper and form the basis of our classification method: *reconstruction accuracy*

and *compression*. Let $\rho$ and $\sigma$ be $n$–qubit quantum states. We say that $\sigma$ is a good representation of $\rho$ if the fidelity $F = \mathrm{Tr}(\sqrt{\rho^{1/2}\sigma\rho^{1/2}}) \geq 1 - \epsilon$ for an $\epsilon > 0$. This accuracy metric cannot be immediately applied to the analysis of VAEs, that can only encode the probability distribution associated to a state. We now show that the fidelity can be expressed in terms of the probability distributions over a measurement that maximally distinguishes the two states. Let $E = \{E_i\}$ be a POVM measurement. Then, using a result by Fuchs and Caves [FC94] we can write

$$F = \min_{E} \sum_{i} \sqrt{\mathrm{Tr}(E_i\rho)\mathrm{Tr}(E_i\sigma)}, \tag{4.1}$$

where the minimum is taken over all possible POVMs. Note that $p(i) = \mathrm{Tr}(E_i\rho)$ and $q(i) = \mathrm{Tr}(E_i\sigma)$ are the probabilities of measuring the state $\rho$ and $\sigma$, respectively, in outcome labelled by $i$ and $\sum_i \sqrt{p(i)q(i)}$ is the Bhattacharyya coefficient between the two distributions.

Eq. 4.1 allows us to define a measure of complexity of a state based on the hardness of approximating the probability distribution $p(i)$ with a classical sampler $q(i)$. Because $F = \sum_i \sqrt{p(i)q(i)}$ is expressed in terms of the probability distributions $p(i)$ and $q(i)$ we can use samples of $p(i)$ and $q(i)$ to estimate $F$. If we assume that sampling from the approximating distribution $q(i)$ is at most as hard as sampling from $p(i)$, we can use the hardness of sampling $p(i)$ as a proxy for the complexity of the state.

Throughout the paper, unless where explicitly mentioned, we work with states that have only positive, real entries in the computational basis. In this case, it is easy to see that the Bhattacharyya coefficient between the distributions reduces to the fidelity and, hence, measurements in the $Z$ basis minimise Eq 4.1.

In order to connect the above definition of state complexity with VAEs, we introduce the *compression factor*. Given an $n$-qubit state that is represented by a VAE with $m$ parameters in the decoder, the compression factor is $C = m/2^n$. We say that a state $\rho$ is *exponentially compressible* if there exists a network that approximates $\rho$ with high accuracy using $m = O(\mathrm{poly}(n))$ parameters.

Once a network is trained, the cost of generating a sample is proportional to the number of parameters in the network. In this sense the complexity of a state is parametrised by the number of parameters used by a neural network representation. Based on these observations, we define *easy states* those that can be represented with high accuracy and exponential compression and *hard states* those that can be represented with high accuracy using at least $\Omega(\exp(n))$ parameters. The last category includes: $(i)$ states that can be efficiently sampled with a quantum computer, but

are conjectured to have no classical algorithm to do so (note that our definition of hardness relies on conjectures on the hardness of certain computational tasks); (*ii*) states that cannot be efficiently obtained on a quantum computer starting from some fixed product input state (*e.g.* random Haar states).

Under this definition, states that admit an efficient classical description (such as stabilizer states or MPS with low bond dimension) are easy, because we know that $O(\text{poly}(n))$ parameters are sufficient to specify the state. Specifically, for the class of easy states we consider separable states obtained by taking the tensor product of $n$ different 1-qubit random states. More formally, we consider states of the form $|\tau\rangle = \bigotimes_{i=1}^{n} |r_i\rangle$ where $|r_i\rangle$ are random 1-qubit states. These states can be described using only $2n$ parameters.

Among the class of hard states of the first kind, we study the learnability of a type of hard distributions introduced in [Fef14] which can be sampled exactly on a quantum computer. These distributions are conjectured to be hard to approximately sample from with a classical computer – the existence of an efficient sampler would lead to the collapse of the Polynomial Hierarchy under some natural conjectures described in [Fef14; AA11].

Finally, for the second class of hard states, we consider random pure states. These are generated by normalising a $2^n$ dimensional complex vector drawn from the unit sphere according to the Haar measure.

### 4.5.1 States that are classically hard to sample from

We study the learnability of a special class of hard states introduced by Fefferman and Umans [FU15] which is produced by a certain quantum computational process which exhibits quantum 'supremacy'. A quantum computation achieves quantum supremacy if it cannot be performed in a reasonable amount of time on an existing classical supercomputer. For a review of quantum supremacy proposals we recommend [HM17a]. More specifically, we consider quantum supremacy proposals where a quantum circuit generates a state whose probability distribution in some basis is — asymptotically — hard to sample from on a classical computer (modulo some very plausible computational complexity assumptions). To demonstrate quantum supremacy one only requires quantum gates to operate within a certain fidelity without full error-correction. This makes efficient sampling from such distributions feasible to execute on near-term quantum devices and opens the search for possibilities to look for practically-relevant decision problems.

To construct a distribution, one starts from an encoding function $h : [m] \to \{0,1\}^N$. The function $h$ performs an efficient encoding of its argument and is used to construct the following so-called *Efficiently Specifiable* polynomial on $n$ variables:

$$Q(X_1, \ldots, X_N) = \sum_{z \in [m]} X_1^{h(z)_1} \ldots X_N^{h(z)_N}, \tag{4.2}$$

where $h(z)_i$ means that we take only the $i$-th bit, and $m$ is an arbitrary integer. In the following, we pick $h$ to be related to the permanent. More specifically, $h : [0, n! - 1] \to \{0,1\}^{n^2}$ maps the $i$-th permutation (out of $n!$) to a string which encodes its $n \times n$ permutation matrix in a natural way resulting in a $N$-coordinate vector, where $N = n^2$. To encode a number $A \in [0, n! - 1]$ in terms of its permutation vector, we first represent $A$ in a factorial number system. The resulting vector $A'$ is an $N$-coordinate vector which identifies a particular permutation $\sigma$.

With the above encoding, our polynomial $Q$ has have the form

$$Q(X_1, \ldots, X_N) = \sum_{z \in [n! - 1]} X_1^{h(z)_1} \ldots X_N^{h(z)_N}. \tag{4.3}$$

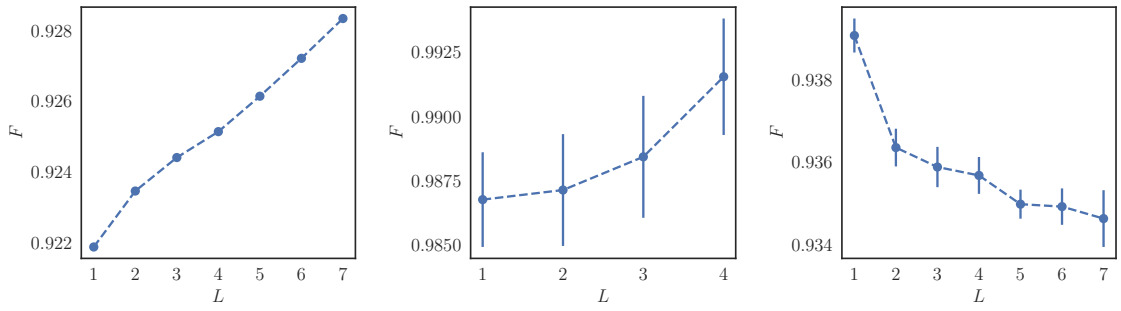
Fix some number $L$ and consider the following set of vectors $y = (y_1, \ldots, y_N) \in [0, L-1]^N$ (i.e. each $y_j$ ranges between 0 and $L-1$). For each $y$ construct another vector $Z_y = (z_{y_1}, \ldots, z_{y_N})$ constructed as follows: each $z_{y_j}$ corresponds to a complex $L$-ary root of unity raised to power $y_j$. For instance, pick $L = 4$ and consider $y' = (1, 2, 3, 0, 2, 3, 0, 4)$. Then the corresponding vector $Z_{y'} = (w^1, w^2, w^3, w^0, w^2, w^3, w^0, w^4)$, where $w = e^{2\pi i/4}$ (for an arbitrary $L$ it is $e^{2\pi i/L}$).

Using the definition of $Q$ we can construct, keeping fixed the value of $L$, each element of the 'hard' distribution $\mathcal{D}_{Q,L}$

$$\Pr_{\mathcal{D}_{Q,L}}[y] = \frac{|Q(Z_y)|^2}{L^N n!}. \tag{4.4}$$

A quantum circuit which performs sampling is remarkably easy. It amounts to applying the quantum Fourier transform to a uniform superposition which was transformed by $h$ and measuring in the standard basis (see Theorem 4 of Section 4 of [FU15]).

Classical sampling of distributions based on the above Efficiently Specifiable polynomial is believed to be hard, in particular because it contains the problem of computing the permanent of a matrix. Note that computing a multiplicative approximation of the permanent of a matrix with integer entries is #P-hard in the worst-case and computing the permanent is #P-hard on average (for more details see [AA11]). Fefferman and Umans conjecture that an approximate average-case




(a) (b) (c)

Figure 4.2: **Depth affects the learnability of hard quantum states.** Fidelity as a function of the number of layers in the VAE decoder for (a) an 18-qubit hard state that is easy to generate with a quantum computer, (b) random 18-qubit product states that admit efficient classical descriptions and (c) random 15-qubit pure states. Error bars for (b) and (c) show the standard deviation for an average of 5 different random states. The compression level $C$ is set to $C = 0.5$ for (a) and (c) and $C = 0.015$ for (b) where $C$ is defined by $\frac{m}{2^n}$ where $m$ is the number of parameters in the VAE decoder and $n$ is the number of qubits. We use a lower compression rate for product states because, due to their simple structure, even a 1 layer network achieves almost perfect overlap. Plot (b) makes use of up to 4 layers in order to avoid the saturation effects discussed in the main text.

solution to any Efficiently Specifiable polynomial is also #P-hard and thus the existence of an efficient classical sampler — for distributions based on the polynomial — would imply a collapse of the Polynomial Hierarchy to the third level (see Section 5 and 6 of [FU15] for detailed proof).

## 4.6 The role of depth in compressibility

Classically, depth is known to play a significant role in the representational capability of a neural network. Recent results, such as the ones by Mhaskar, Liao, and Poggio [MLP16], Telgarsky [Tel16], and Eldan and Shamir [ES16] showed that some classes of functions can be approximated by deep networks with the same accuracy as shallow networks but with exponentially fewer parameters.

The representational capability of networks that represent quantum states remains largely unexplored. Some of the known results are only based on empirical evidence and sometimes yield unexpected results. For example, Morningstar and Melko [MM17] showed that shallow networks are more efficient than deep ones when learning the energy distribution of a 2-dimensional Ising model.

In the context of the learnability of quantum states, Gao and Duan [GD17] proved that DBMs can efficiently represent some states that cannot be efficiently represented by shallow networks (*i.e.* states generated by polynomial depth circuits or $k$-local Hamiltonians with polynomial size gap) using a polynomial number of hidden units. However, there are no known methods to sample efficiently from DBMs when the weights include complex-valued coefficients.

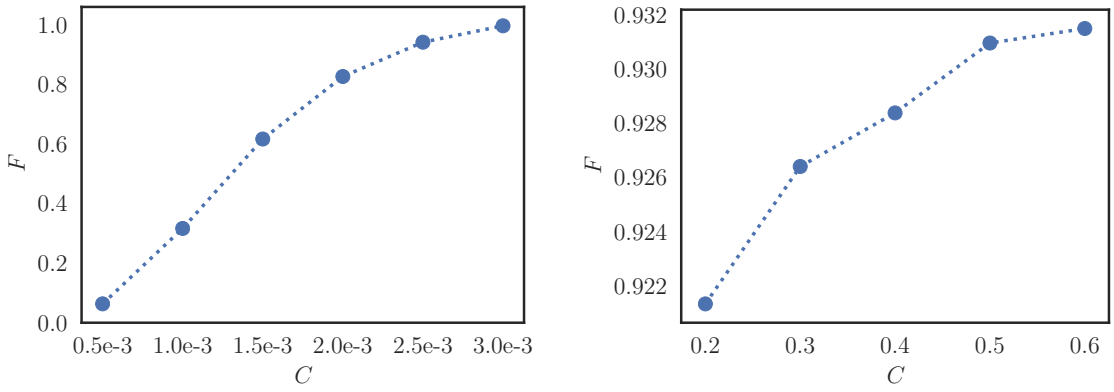
We benchmark with numerical simulations the role played by depth in compressing states of different levels of complexity. We focus on three different states: an easy state (the completely separable state discussed in the previous section), a hard state (specified through the procedure defined by Fefferman and Umans), and a random pure state.

Our results are presented in Fig. 4.2. Here, by keeping the number of parameters in the decoder constant, we determine the reconstruction accuracy of networks with increasing depth. Remarkably, depth affects the reconstruction accuracy of hard quantum states. This might indicate that VAEs are able to capture correlations in hard quantum states. As a sanity check we notice that the network can learn correlations in random product states and that depth does not affect the learnability of random states.

## 4.7 Efficient representation of quantum states

In this section we focus our attention onto two questions: can VAEs find efficient representations of easy states? What level of compression can we obtain for hard states? Through numerical simulations we show that VAEs can learn to efficiently represent some easy states (that are challenging for standard methods) and achieve good levels of compression for hard states. Remarkably, our methods allow to compress up to a factor 5 the hard quantum states introduced in [FU15]. We remark that the exponential hardness cannot be overcome for general quantum states and our methods achieve only a factor improvement on the overall complexity.

We test the performance of the VAE representation on two classes of states: the hard states that can be constructed efficiently with a quantum computer introduced by Fefferman and Umans [FU15] and states that can be generated with a long-range Hamiltonian dynamics, as found for example in experiments with ultra-cold ions [Ric+14]. The states generated through this evolution are highly symmetric physical states. However, due to the bond dimension increasing exponentially with the evolution time, these states are particularly challenging for MPS methods. An




(a) (b)

Figure 4.3: **VAEs can learn efficient representation of easy states and can be used to characterize hard states.** Fidelity as a function of compression $C = m/2^n$ for (a) an 18-qubit state generated by evolving $2^{-n/2}\sum_i |i\rangle$ using the long-range Hamiltonian time evolution for a time $t = 20$ and (b) an 18-qubit hard state generated according to [FU15]. Figure (a) shows that the VAE can learn to represent efficiently with almost perfect accuracy easy states that are challenging for MPS. Figure (b) shows that hard quantum states can be compressed with high reconstruction accuracy up to a factor 5. The decoder in (a) has 1 hidden layer to allow for greater compression without incurring in the saturation effects discussed in main text. The decoder in (b) has 6 hidden layers in order to maximise the representational capability of the network.

interesting question is to understand whether neural networks are able to exploit these symmetries and represent these states efficiently.

We consider the evolution $|\Psi(t)\rangle = e^{-i\mathcal{H}t}|\Psi(t=0)\rangle$ of a product state $|\Psi(t=0)\rangle = 2^{-n/2}\sum_i |i\rangle$ through the Hamiltonian

$$\mathcal{H} = \sum_{i<j} V(i,j)\left(\sigma_i^x\sigma_j^x + \sigma_i^y\sigma_j^y\right),$$

where $V(i,j) = 1/|i-j|^{3/4}$ is a long-range two-body interaction. At long propagation times $t \gg 1$, the resulting states are highly entangled, and are, for example, challenging for MPS-based tomography [Cra+10]. To assess the ability of VAE to compress highly entangled states, we focus on the task of reconstructing the outcomes of experimental measurements in the computational basis. In particular, we generate samples distributed according to the probability density $|\Psi_i(t)|^2$, and reconstruct this distribution with our generative deep models.

Results are displayed in Fig. 4.3. For states obtained through the long-range Hamiltonian evolution we achieve with almost maximum reconstruction accuracy

compression levels of up to $C \approx 10^{-3}$. This corresponds to a number of parameters $m = \mathcal{O}(100) \ll 2^{18}$ which implies that the VAE has learned an efficient representation of the state.

In the case of hard states, we can reach a compression of 0.2, corresponding to a factor 5 reduction in the number of parameters required to represent the state. Note that the entanglement properties of hard states are likely to make them hard to compress for tensor network states. For example, if one wanted to compress an 18 qubits state using MPS (a type of tensor network that is known to be efficiently contractable) we have found that the estimated bond dimension to reconstruct this state is $D \approx 460$. This number is obtained computing the largest bipartite entanglement entropy ($S$), and estimating the bond dimension with $D \approx 2^S$. Considering that an MPS has $D^2$ variational parameters (in the best case), this would yield about $2 \times 10^5$ variational parameters required to represent those hard states. The resulting MPS compressing factor is then about 1.23, a significantly lower figure with respect to the compression factor of 5 obtained with VAEs. We note that this calculation only shows that the entanglement structure of hard states is not well modelled by MPS. Other types of tensor networks might be more amenable to the specific structure of these states but it is unlikely these models will be computationally tractable.

Although limited, the levels of compression we achieve for hard states could play a role in experiments aimed at characterising early quantum devices. In this setting, a quantum machine with a handful of noisy qubits performs a task that is not reproducible even by the fastest supercomputer. Our methods might allow to characterise the result of a computation by reducing the complexity of the problem. Potentially, a neural network approach to characterisation can be accomplished by compressing a trusted initial state into a VAE whose parameters are then evolved according to a set of rules specified by the quantum circuit. By comparing the experimental distribution with the one sampled with the VAE it is then possible to determine whether the device is faulty.

# Chapter 5

# Quantum PAC learnability of DNFs

In the previous chapters, we studied the learnability of quantum mechanical systems using classical algorithms. In this chapter, we discuss the learnability of classes of functions using quantum algorithms and under the assumption that the training data come in quantum superposition. Our primary interest is in determining whether quantum computers operating on quantum data can perform efficiently learning tasks that are hard for classical learners. We emphasise that the notion of hardness we use in this chapter is in the computational complexity sense, i.e. a *hard* problem is one for which no polynomial time algorithm is known. The broader question of whether quantum algorithms can solve more efficiently learning problems that have already polynomial time algorithms is certainly relevant for practical applications of quantum computers but will not be discussed here.

All the problems in this chapter are addressed within the framework of computational learning theory and, in particular, the PAC model. The PAC model was extended to the quantum case by Bshouty and Jackson [BJ98]. There are two additions with respect to the classical version. First, the learning algorithm can be quantum. Second, the training set comes in quantum superposition. As for the classical PAC model there are two questions that are relevant for the performance of a learning algorithm: sample complexity and time complexity. The focus of this chapter is to understand whether quantum computation confers any advantage in terms of time complexity for learning problems.

The question of whether quantum examples confer any advantage in terms of sample complexity was resolved by Arunachalam and de Wolf who proved that, in the distribution free setting, the quantum PAC model has only a constant factor advantage in terms of sample complexity with respect to its classical analogue [AW17].

Although quantum examples are no more powerful than classical ones in terms of sample complexity, certain results suggest that a classical/quantum separation exists when considering the time complexity of learning problems. For example, when learning with respect to the uniform distribution, the class of polynomial-size DNF formulae [BJ98] and $k$-juntas [AS07] under the uniform distribution are known to be efficiently quantum PAC-learnable. In the classical setting, in both these cases, the current best known algorithms run in quasi-polynomial time (assuming $k = \omega(1)$). Information-theoretic lower bounds are known in more restricted models, such as the statistical query model, which suggest these classes cannot be learnt in polynomial time [Blu+94].

In the context of learning in the presence of noise, Cross, Smith, and Smolin proved that parity functions under the uniform distribution can be efficiently learned using a quantum example oracle [CSS15]. Classically, the problem is widely believed to require subexponential, but superpolynomial, time [BKW03; Lyu05]. The result proved in [CSS15] was extended to linear functions and to more complex error models by Grilo, Kerenidis, and Zijlstra [GKZ17].

In this chapter, we study the quantum PAC learnability of the class of boolean functions that can be expressed as polynomial size formulae in disjunctive normal form (DNF). Whether DNF are learnable in polynomial time, or not, is one of the central unresolved questions in the PAC learning framework. Currently, the best classical algorithm for this problem has running time $2^{\tilde{O}(n^{1/3})}$ [KS01]. A number of variants of this problem have been studied and it has been shown that learning DNFs is possible, for example, in models with access to membership query [Jac94; AFK13], to quantum examples [BJ98], or when the examples are generated by a uniform random walk on the Boolean hypercube [BJ98].

## 5.1 Overview of our results

We prove that DNF formulae under constant-bounded product distributions can be learned in polynomial time in the quantum PAC model. Our proof builds on the work by Feldman for learning DNFs under the product distribution using membership queries [Fel12]. Feldman's proof is in turn based on a result by Kalai, Samordintsky, and Teng that shows that DNFs can be approximated by heavy low-degree Fourier coefficients alone [KST09] . Notably, Feldman's result also applies to learning settings where the examples are drawn from a product distribution, i.e. a distribution that factorises over the elements of the input vector.

The only part of Feldman's algorithm that makes use of membership queries is the subroutine that approximates the Fourier spectrum of $f$. The approximation is obtained using the *Kushilevitz-Mansour* (KM) algorithm [KM93], for the case of uniform distributions, and the *extended Kushilevitz-Mansour* (EKM) algorithm [KST09], for the case of product distributions. Bshouty and Jackson showed that it is possible to approximate the Fourier coefficients of $f$ using quantum Fourier sampling. This technique, introduced in [BV97], allows one to sample efficiently from the distribution defined by $\hat{f}^2$ defined by the squared Fourier coefficients using the quantum Fourier transform (QFT).

In order to extend the result of Bshouty and Jackson to product distributions, it is sufficient to find a quantum technique to sample according to the squared coefficients of a Fourier transform defined over an inner product where each term is weighted according to the product distribution. Bahdur [Bah61] and Furst, Jackson, and Smith [FJS91] showed that the Fourier transform can be extended to product distributions. The resulting function is known as the *$\mu$–biased Fourier transform*.

In this work we introduce the *$\mu$–biased quantum Fourier transform*. We show the validity of our construction in two steps. First, we explicitly construct a unitary operator that implements the single qubit transform. Then, we argue that this construction can be efficiently implemented on a quantum circuit with logarithmic overhead. By exploiting the factorisation of product distributions, we show how to build an $n$-qubit transform as a tensor product of $n$ single qubit transforms. Our algorithm does not require prior knowledge of the parameter $\mu$ that characterises the product distribution. This can be estimated efficiently via sampling and we show that the error introduced with this procedure can be bounded.

The main technical contribution of this chapter is a quantum algorithm to approximate the heavy, $\mu$–biased, low-degree Fourier spectrum of $f$ for constant-bounded product distributions without using membership queries (recall that membership queries are necessary in Feldman's classical algorithm). This can be interpreted as a quantum version of the EKM algorithm for approximating the low-degree Fourier coefficients of $f$. We provide rigorous upper bounds on the scaling of the algorithm using the Dvoretzky–Kiefer–Wolfowitz theorem, a concentration inequality that bounds the number of samples required to estimate a probability distribution in infinity norm. The learnability of DNFs under the product distribution immediately follows from an application of the quantum EKM algorithm to Corollary 5.1 in [Fel12].

## 5.2 Related work

The best known classical algorithm for learning DNF has running time $2^{\tilde{O}(n^{1/3})}$ [KS01]. Two cases in which it is possible to show DNF learnability, in the classical PAC model, under specific assumptions are particularly relevant to our discussion. First, when the distribution is uniform a quasi-polynomial $n^{O(\log(n))}$ algorithm is known [Ver90]. Second, in the membership query model, where the learner can query an oracle for a value of the unknown function at a given point in the domain, Jackson gave a polynomial time algorithm for DNFs that works over both the uniform and product distributions [Jac97].

The learnability of DNF formulae under the uniform distribution using a quantum example oracle was first studied by Bshouty and Jackson, who, in the same paper, also introduced the quantum PAC model [BJ98]. Their approach to learning DNF was built on the *harmonic sieve* algorithm developed in [Jac97]. Jackson's algorithm exploits a property of DNF known as concentration of Fourier spectrum. More specifically, Jackson used the fact that for every $s$-term DNF and for every probability distribution $\mathcal{D}$, there exists a parity $\chi_a$ such that $|\mathbb{E}_{\mathcal{D}}[f\chi_a]| \geq 1/(2s+1)$. This implies that for every $f$ and $\mathcal{D}$ there exists a parity that weakly approximates $f$. In the harmonic sieve algorithm, the boosting algorithm of Freund is then used to turn the weak learner into a strong one [Fre95]. The only part of the harmonic sieve algorithm that requires membership queries is the KM algorithm used to find the weakly approximating parity function. Bshouty and Jackson consider the setting where the examples are given by a quantum example oracle and replace the KM algorithm with quantum Fourier sampling.

Jackson, Tamon, and Yamakami studied the learnability of DNFs in the quantum membership model [JTY02] (where the quantum example oracle is replaced by an oracle that returns $f(x)$ for a given $x$). By using the quantum Goldreich-Levin algorithm developed by Adcock and Cleve [AC02], they were able to obtain a better bound on the query complexity with respect to the best classical algorithm. We recall that the classical KM algorithm can be derived from the Goldreich-Levin theorem, an important result that reduces the computational problem of inverting a one-way function to the problem of predicting a given hard-predicate associated with that function [GL89]. The result in [AC02] shows that this reduction can be obtained more efficiently when considering quantum functions and quantum hard–predicates. A different quantum implementation of the Goldreich-Levin algorithm was given in [MO10].

The class of constant depth circuits ($\mathsf{AC}^0$) is also known to have a concentrated Fourier spectrum [LMN89]. Classical lower bounds proved the hardness of learning $\mathsf{AC}^0$ circuits based on the hardness of factoring integers [Kha93]. Because quantum computers can factor integers efficiently, this bound does not hold for quantum learning algorithms and it was hoped that by using the same Fourier sampling techniques developed for DNF it would have been possible to quantum learn $\mathsf{AC}^0$ efficiently. Recently, Arunachalam, Grilo, and Aarthi proved that $\mathsf{AC}^0$ circuits are not efficiently quantum PAC learnable under the assumption that learning with errors is hard for quantum computers [AGS19].

## 5.3 Fourier analysis over the Boolean cube

We revise some standard notions of analysis of Boolean functions. For an introduction to the subject we recommend the notes by de Wolf [Wol08] and the textbook by O'Donnel [ODo14].

Let $x \in \{-1,1\}^n$ and let $f$ and $g$ be real-valued functions defined over the Boolean hypercube $f, g : \{-1,1\}^n \to \mathbb{R}$. The space of real functions over the Boolean hypercube is a vector space with inner product $\langle f, g\rangle = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x)g(x) = \mathbb{E}[f \cdot g]$ where the expectation is taken uniformly over all $x \in \{0,1\}^n$. A *parity function* $\chi_a : \{-1,1\}^n \to \{-1,1\}$ labels a $x \in \{-1,1\}^n$ according to a characteristic vector $a \in \{0,1\}^n$ and is defined as $\chi_a(x) = (-1)^{a\cdot x}$ where $a \cdot x = \sum_{i=1}^n a_i x_i$. The set of parity functions $\{\chi_a\}_{a\in\{0,1\}^n}$ forms an orthonormal basis for the space of real-valued functions over the Boolean hypercube. This fact implies that we can uniquely represent every function $f$ as a linear combination of parities, the *Fourier transform* of $f$. The linear coefficients, known as the *Fourier coefficients*, are given by the projections of the function into the parity base and are denoted with $\hat{f}(a) = \langle f, \chi_a\rangle = \mathbb{E}[f(x)\chi_a(x)]$. The set of Fourier coefficients is called the *Fourier spectrum* of $f$ and is denoted by $\hat{f}$, which can also be seen as a $2^n$ dimensional vector in $\mathbb{R}^{2^n}$. For a set $S \subseteq \{0,1\}^n$, $\hat{f}(S)$ denotes the vector of all Fourier coefficients with indices in $S$. The *degree* of a Fourier coefficient $\hat{f}(a)$ is $\|a\|_0$. Let $B_d = \{a \in \{0,1\}^n \mid \|a\|_0 \leq d\}$. We denote by $\hat{f}(B_d)$ the vector of all degree–$\leq d$ coefficients of $f$. The squared Fourier coefficients are related by Parseval's identity $\mathbb{E}[f^2] = \sum_a \hat{f}(a)^2 = \|\hat{f}\|_2^2$. This implies that for any $f : \{-1,1\}^n \to [-1,1]$, $\sum_a \hat{f}(a)^2 \leq 1$. The equality holds if $f$ is Boolean–valued, i.e. if $f : \{-1,1\}^n \to \{-1,1\}$ and therefore $\hat{f}^2$ forms a probability distribution. Let $\gamma > 0$, we say that a Fourier coefficient $\hat{f}(a)$ is $\gamma$-*heavy* if it has large magnitude $|\hat{f}(a)| > \gamma$. By Parseval's the number of $\gamma$-heavy Fourier coefficient is at most $\|\hat{f}\|_2^2/\gamma$.

The Fourier spectrum of a function $f$ can be approximated using the KM algorithm in $\ell_\infty$ norm. The KM algorithm, based upon a celebrated result by Goldreich and Levin [GL89], requires membership query access to $f$ (*i.e.* it requires an oracle that for every $x \in \{-1,1\}^n$ returns $f(x)$).

**Theorem 4** (KM algorithm)**.** *Let $f : \{-1,1\}^n \to [-1,1]$ be a real–valued function and let $\epsilon > 0$, $\delta > 0$. Then, there exists an algorithm with oracle access to $f$ that, with probability at least $1-\delta$, returns a succinctly represented vector $\tilde{f}$ such that $\|\hat{f} - \tilde{f}\|_\infty \leq \epsilon$ and $\|\tilde{f}\|_0 \leq 4/\epsilon^2$. The algorithm runs in $\tilde{O}(n^2 \log(1/\delta)/\epsilon^6)$ time and makes $\tilde{O}(n \log(1/\delta)/\epsilon^6)$ queries to $f$.*

### 5.3.1 $\mu$–biased Fourier analysis

A *product distribution* $\mathcal{D}_\mu$ over $\{-1,1\}^n$ is characterised by a real vector $\mu \in (-1,1)^n$. Such a distribution $\mathcal{D}$ assigns values to each variable independently, so for $x \in \{-1,1\}^n$ we have $\mathcal{D}_\mu(x) = \prod_{i:x_i=1}(1+\mu_i)/2 \prod_{i:x_i=-1}(1-\mu_i)/2$ and $\mathbb{E}_\mu[x_i] = \mu_i$. Notice that for $\mu = 0$ one recovers the uniform distribution. We say that the distribution $\mathcal{D}_\mu$ is *c-bounded*, or *constant-bounded*, if $\mu \in [-1+c, 1-c]^n$, where $c \in (0,1]$.

Bahdur [Bah61] and Furst, Jackson, and Smith [FJS91] showed that the Fourier transform can be extended to product distributions. The resulting function is known as the *$\mu$–biased Fourier transform*. The book by O'Donnell contains a brief introduction to $\mu$–biased Fourier analysis and its applications [ODo14]. For an inner product $\langle f, g \rangle_\mu = \mathbb{E}_\mu[f(x)g(x)]$, the set of functions $\{\phi_{\mu,a} \,|\, a \in \{0,1\}^n\}$, where $\phi_{\mu,a}(x) = \prod_{i:a_i=1}(x_i - \mu_i)/\sqrt{1-\mu_i^2}$ forms an orthonormal basis for the vector space of real–valued functions on $\{-1,1\}^n$. In this way every function $f : \{-1,1\}^n \to \mathbb{R}$ can be represented as $f(x) = \sum_{a \in \{0,1\}^n} \hat{f}_\mu(a)\phi_{\mu,a}(x)$, where $\hat{f}_\mu(a) = \mathbb{E}_\mu[f(x)\phi_{\mu,a}(x)]$. For vectors of $\mu$–biased Fourier coefficients we extend the same notation introduced for standard Fourier coefficients. Parseval's identity extends to product distributions $\mathbb{E}_\mu[f^2] = \sum_a \hat{f}_\mu(a)^2 = \|\hat{f}_\mu\|_2^2$. This implies that for any $f : \{-1,1\}^n \to [-1,1]$, $\sum_a \hat{f}_\mu(a)^2 \leq 1$.

The KM algorithm has been extended to product distributions in [Bel91; Jac97; KST09]. We follow the presentation of Feldman and give the version presented in [KST09].

**Theorem 5** (EKM algorithm)**.** *Let $f : \{-1,1\}^n \to [-1,1]$ be a real–valued function and let $\epsilon > 0$, $\delta > 0$, $\mu \in (-1,1)^n$. Then, there exists an algorithm with oracle access to $f$ that, with probability at least $1-\delta$, returns a succinctly represented vector $\tilde{f}_\mu$*

*such that* $\|\hat{f}_\mu - \tilde{f}_\mu\|_\infty \leq \epsilon$ *and* $\|\tilde{f}_\mu\|_0 \leq 4/\epsilon^2$. *The algorithm runs in time polynomial in* $n$,$1/\epsilon$ *and* $\log(1/\delta)$.

## 5.4 The quantum PAC model

Let $f$ be a Boolean-valued function $f : \{-1,1\}^n \to \{-1,1\}$ and $\mathcal{D}$ a probability distribution over $\{-1,1\}^n$. In the PAC model the learning algorithm is classical and has access to an example oracle $\mathrm{EX}(f,\mathcal{D})$ that returns an example $(x, f(x))$ where $x$ is randomly sampled from $\mathcal{D}$. In the *quantum PAC* model the learning algorithm is quantum and has access to a *quantum example oracle* $\mathrm{QEX}(f,\mathcal{D})$. The action of a $\mathrm{QEX}(f,\mathcal{D})$ oracle for a target function $f$ and probability distribution $\mathcal{D}$ is

$$\mathrm{QEX}(f,\mathcal{D}) : \ |0\ldots0,0\rangle \to \sum_x \sqrt{\mathcal{D}(x)}\,|x, f(x)\rangle\,.$$

It is useful to compare the power of a quantum example oracle and a quantum membership oracle. Recall that a quantum membership oracle acts as $O_f : |i\rangle\,|0\rangle \to |i\rangle\,|f(i)\rangle$. It was proven that quantum membership queries are strictly more powerful than quantum example queries that is, a quantum example oracle can be simulated using membership queries but the converse is not true [BJ98]. It is instructive to see how to simulate a quantum example oracle using quantum membership queries. First, note that given a (continuous) probability distribution whose density is efficiently integrable, there exists an efficient technique developed by Grover and Rudolph to generate a quantum superposition which approximates the distribution [GR02]. In order to reach the target state, the technique exploits a sequence of controlled rotations whose magnitudes are proportional to the probabilities of two equal partitions of the probability space. Each controlled rotation can be efficiently implemented because the density is efficiently integrable over the entire probability space.

**Lemma 2.** *Let* $\mathcal{D}$ *be a continuous probability distribution and let* $\mathcal{D}^*$ *be a discretisation of* $\mathcal{D}$ *over* $\{0,1\}^n$. *If there exists an efficient classical algorithm to compute* $\int_a^b \mathcal{D}(x)dx$ *for every* $a, b \in \mathbb{R}$ *then, there exists an efficient quantum algorithm that returns the quantum state*

$$|\psi\rangle = \sum_i \sqrt{\mathcal{D}^*(i)}\,|i\rangle\,.$$

Starting from the state $|0\ldots0,0\rangle$, use Lemma 2 on the first register to obtain $\sum_x \sqrt{\mathcal{D}(x)}\,|x,0\rangle$. Finally, an application of the quantum membership oracle returns the state $\sum_x \sqrt{\mathcal{D}(x)}\,|x, f(x)\rangle$.

An alternative way to construct a quantum example oracle is through a *quantum random access memory* (QRAM). A QRAM is a quantum procedure that allows one to encode in superposition $N$ data points into $\log(N)$ qubits in time $O(\log(N))$. More specifically, let $\{m_j\}$ be the content of a memory structure with $N$ elements. The action of the QRAM on a state $\sum_i \alpha_i |i, 0\rangle$ is

$$\text{QRAM} : \sum_i \alpha_i |i, 0\rangle \to \sum_i \alpha_i |i, m_j\rangle .$$

Note that because the QRAM can load data in logarithmic time, it is possible to build a superposition that encodes a Boolean function supported on $\{-1, 1\}^n$ in polynomial time. A possible implementation of the QRAM is presented in [GLM08b]. The idea is to use a tree-structure where the nodes are qutrits (quantum systems with tree levels) and the leaves encode the entries of the function. By traversing the tree in quantum superposition it is possible to return a quantum state that encodes a superposition of the elements of the image of $f$ in $O(\log(N))$ time.

Note that the tree requires $O(N)$ qutrits. This exponential scaling in terms of physical resources is one of the main reasons why it is still unclear whether a QRAM can be built [Aar15]. The issue can be related to whether the exponential number of components needs to be continuously 'active'. If all the nodes are active during a query then error correcting the tree would require exponential physical resources and therefore it would be impractical to build the device in an experimental setting. The proponents of the QRAM claim that at every QRAM query only $O(\log(N))$ components need to be active while the others can be considered as 'non-active' and error free [GLM08b; GLM08a]. Whether this assumption holds in real settings is unclear. Arunachalam et al. [Aru+15] addressed this question and showed that, under a given error model, algorithms that require to query the memory a polynomial number of times (like the quantum linear system algorithm [HHL09]) might not require fault-tolerant components. However, for superpolynomial query algorithms, like Grover's search, the QRAM requires error-corrected components.

## 5.5 DNF learnability and Fourier spectrum

A *DNF formula* $f : \{-1, 1\}^n \to \{-1, 1\}$ is a disjunction of terms where each term is a conjunction of Boolean literals and a literal is either a variable or its negation (*e.g.*, $f(x) = (x_2 \wedge x_3 \wedge \neg x_1) \vee (\neg x_4 \wedge x_1)$). The *size* $s$ of a DNF is the number of terms.

An important property of DNFs is that their Fourier spectrum concentrates on a small set of coefficients. Therefore, by learning the heavy Fourier coefficients it

is possible to learn the function. The connection between Fourier spectrum and learnability was first established by Linial, Mansour, and Nisian for $\mathsf{AC}^0$ (polynomial size constant depth circuits) [LMN89]. Mansour gave the first algorithm specifically exploiting the Fourier concentration property for learning DNFs [Man95]. For a survey of Fourier based techniques in learning theory we recommend [Man94].

The harmonic sieve algorithm by Jackson also exploits the concentration of Fourier spectrum of DNFs [Jac97]. This algorithm requires a complicated boosting procedure. Our proof of the learnability of DNFs under the product distribution builds on an algorithm by [Fel12] that greatly simplified the learnability of DNFs. At the core of Feldman's algorithm lies a result by Kalai, Samorodnitsky, and Teng [KST09] that shows that DNFs can be approximated by heavy low-degree Fourier coefficients alone. More formally, they proved that, for any $s$-term DNF $f$ and for every function $g : \{-1,1\}^n \to [-1,1]$, the distance between $f$ and $g$ (measured as $\mathbb{E}_\mu[|f(x) - g(x)|]$) is $\mathbb{E}_\mu[|f(x)-g(x)|] \leq (2s+1) \cdot \|\hat{f} - \hat{g}\|_\infty$. This fact gives a direct learnability condition and avoids an involved boosting procedure to turn a weak learner into a strong one (as in the harmonic sieve algorithm by [Jac97]). Feldman further refined this fact about DNFs.

**Theorem 6** (Theorem 3.8 in [Fel12])**.** *Let $c \in (0,1]$ be a constant, $\mu$ be a $c$-bounded distribution and $\epsilon > 0$. For an integer $s > 0$ let $f$ be an $s$-term DNF. For $d = [\log(s/\epsilon)/\log(2/(2-c))]$ and every bounded function $g : \{-1,1\}^n \to [-1,1]$,*

$$\mathbb{E}_\mu[|f(x) - g(x)|] \leq (2 \cdot (2-c)^{d/2} \cdot s + 1) \cdot \left\| \hat{f}_\mu(B_d) - \hat{g}_\mu(B_d) \right\|_\infty + 4\epsilon.$$

By this theorem the learnability of DNF reduces to constructing a $g$ that approximates the heavy low-degree Fourier spectrum of $f$. This is exactly the approach followed by Feldman that we now proceed to sketch.

The first step of the procedure is to run the EKM algorithm to estimate the heavy Fourier spectrum of $f$. The EKM algorithm returns a succinct representation of $\hat{f}$ and the learner selects only the coefficients that have degree $\leq d$. This is the only step of the algorithm that requires membership queries and is the subroutine that will be replaced by the quantum EKM algorithm that will be derived in Section 5.7.

Once the learner has estimated the Fourier spectrum of $f$, it proceeds with the construction of $g$. The procedure is simple and based on an iterative process. Note that by Parseval

$$\mathbb{E}_\mu[(f-g)^2] = \sum_b (\hat{f}_\mu(b) - \hat{g}_\mu(b))^2 = \|\hat{f}_\mu - \hat{g}_\mu\|_2^2. \tag{5.1}$$

Suppose there exists an $a$ such that $|\hat{f}_\mu(a) - \hat{g}_\mu(a)| \geq \gamma$. It is possible to construct a $g'$ such that $g'$ is closer than $g$ to $f$ in $l_2$ norm with the following rule:

$$g' = g + (\hat{f}_\mu(a) - \hat{g}_\mu(a))\phi_{\mu,a}.$$

Then by Eq. 5.1 we have that

$$\begin{aligned}\mathbb{E}_\mu[(f-g')^2] &= \sum_{b\neq a}(\hat{f}_\mu(b) - \hat{g}_\mu(b))^2 \\ &= \mathbb{E}_\mu[(f-g)^2] - (\hat{f}_\mu(a) - \hat{g}_\mu(a))^2 \\ &\leq \mathbb{E}_\mu[(f-g)^2] - \gamma^2.\end{aligned}$$

The problem with this procedure is that the function $g'$ might have value outside $[-1,1]$ but Feldman showed that the function can be adjusted to the right range and made closer to $f$ in $\ell_2$ distance by cutting-off all the values outside of $[-1,1]$.

Once a precision has been reached such that an application of Theorem 6 gives $\mathbb{E}_\mu[|f(x) - g(x)|] \leq \epsilon$, the algorithm outputs $\text{sign}(g)$ as hypothesis. From this, we get the following in regards to learning $f$,

$$\Pr_\mu[f \neq \text{sign}(g)] \leq \mathbb{E}_\mu[|f-g|] \leq \epsilon.$$

The running time of all the above operations is polynomial in $n$ and inverse polynomial in the error parameters resulting in the following corollary

**Corollary 7** (Corollary 5.1 in [Fel12])**.** *Let $f$ compute an $s$-term DNF. Let $c \in (0,1]$ be a constant and let $\mathcal{D}_\mu$ be a $c$-bounded probability distribution. Let $\text{EX}(f,\mu)$ be an example oracle and $\text{MQ}(f)$ a membership oracle. Then, there exists an algorithm with $\text{EX}(f,\mu)$ and $\text{MQ}(f)$ access that efficiently PAC learns $f$ over $\mathcal{D}_\mu$.*

Finally, we note that the requirement of $c$-bounded distributions is imposed in order to control the magnitude of modulus of the $\mu$-biased Fourier basis $\{|\phi_{\mu,a}|\}$ that, otherwise, would diverge for $\mu$ close to $+1$ or $-1$.

## 5.6 Quantum $\mu$–biased Fourier transform

In this section we introduce the $\mu$–biased quantum Fourier transform and show how this can be used to derive a quantum algorithm for sampling from the probability distribution defined by the Fourier coefficients of the $\mu$–biased transform. We recall that the $\mu$–biased Fourier transform is defined as

$$f(x) = \sum_{a\in\{0,1\}^n} \hat{f}_\mu(a)\phi_{\mu,a}(x), \tag{5.2}$$

where $\phi_{\mu,a}(x) = \prod_{i:a_i=1}(x_i - \mu_i)/\sqrt{1-\mu_i^2}$, $\hat{f}_\mu(a) = \mathbb{E}_\mu[f(x)\phi_{\mu,a}(x)]$, and $\mathcal{D}_\mu(x) = \prod_{i:x_i=1}(1+\mu_i)/2\prod_{i:x_i=-1}(1-\mu_i)/2$. Our construction of the $n$-qubit $\mu$–biased QFT exploits a fundamental property of product distributions, namely that the orthonormal basis $\{\phi_{\mu,a}\}$ it defines can be factorised on the individual bits. This fact allows us to give an explicit form of the $n$-qubit transform as a tensor product of $n$ single qubit transforms. We begin by constructing the single qubit transform. Later we will show how to construct efficiently an $n$-qubit transform out of $n$ single qubit ones. In the following we assume that the function $f : \{-1,1\}^n \to \{-1,1\}$ is Boolean–valued. Our results can be extended to real–valued functions over the Boolean hypercube using a discretisation procedure. As shown in [BJ98] the error induced by the approximation can be controlled.

Let $b \in \{-1,1\}$ and $v \in \{0,1\}$. The action of the single qubit $\mu$-biased QFT can be explicitly constructed (the normalisation follows from noticing that $\{\phi_{\mu,v}\}$ forms an orthonormal basis)

$$H_\mu |b\rangle = \sum_{v\in\{0,1\}} \sqrt{\mathcal{D}_\mu(b)}\phi_{\mu,v}(b) |b\rangle . \tag{5.3}$$

Here we defined $H_\mu$ as the single qubit $\mu$–biased QFT operator whose description in the computational basis is readily given by:

$$H_\mu = \begin{bmatrix} \sqrt{\mathcal{D}_\mu(-1)}\phi_{\mu,0}(-1) & \sqrt{\mathcal{D}_\mu(1)}\phi_{\mu,0}(1) \\ \sqrt{\mathcal{D}_\mu(-1)}\phi_{\mu,1}(-1) & \sqrt{\mathcal{D}_\mu(1)}\phi_{\mu,1}(1) \end{bmatrix}.$$

By taking the functional forms of $\mathcal{D}_\mu(x)$ and $\phi(x)$ we can write

$$H_\mu = \begin{bmatrix} \sqrt{\frac{1-\mu}{2}} & \sqrt{\frac{1+\mu}{2}} \\ -\frac{(1+\mu)\sqrt{1-\mu}}{\sqrt{2-2\mu^2}} & -\frac{(-1+\mu)\sqrt{1+\mu}}{\sqrt{2-2\mu^2}} \end{bmatrix}.$$

It is easy to verify that this matrix is unitary and positive semidefinite. We also note that, as consequence of the Solovay-Kitaev theorem [Kit97], it is possible to approximate $H_\mu$ to accuracy $\epsilon$ (in operator norm) using $\Theta(\log^c(1/\epsilon)$ gates from a fixed finite set of universal gates ($c$ is a constant approximately equal to 2.)

We can construct the extension of the $\mu$–biased QFT to the case of $n$ qubits by taking the tensor product of $n$ single qubit operators. Let $x \in \{-1,1\}^n$ and $a \in \{0,1\}^n$, if we denote as $a_i \in \{0,1\}$ the $i$-th bit of $a$, $\mathcal{D}_{\mu_i}(x)$ as the probability associated to the $i$-th bit, and $\phi_{\mu,a_i}(x)$ its respective basis element, we can write:

$$H_\mu \otimes \cdots \otimes H_\mu |x\rangle = \sum_{a_1} \cdots \sum_{a_n} \prod_{i=1}^{n} \sqrt{\mathcal{D}_{\mu_i}(x)}\phi_{\mu_i,a_i}(x) |a_1\rangle \ldots |a_n\rangle .$$

By exploiting the product structure of $\mathcal{D}_\mu$ and $\{\phi_{\mu,a}\}$ that is, $\mathcal{D}_\mu(x) = \prod_i \mathcal{D}_{\mu_i}(x)$ and $\{\phi_{\mu,a}(x) = \prod_i \phi_{\mu_i,a_i}(x)\}$ we can write the $n$ qubit $\mu$–biased QFT as:

$$H_\mu^n |x\rangle = \sum_{a\in\{0,1\}^n} \sqrt{\mathcal{D}_\mu(x)}\phi_{\mu,a}(x) |a\rangle . \tag{5.4}$$

We remark that it is possible to construct the $n$ qubit transform only because the product distribution and the $\{\phi_{\mu,a}\}$ basis factorises. Without this factorisation we could still write Eq. 5.4 but we would not know how to implement this transformation efficiently on a quantum computer (the Solovay-Kitaev theorem guarantees that only single qubit unitaries can be efficiently approximated by a universal set of gates).

Finally, we note that the construction of the $\mu$-biased transform assumes knowledge of the vector $\mu$. It is possible to estimate $\mu_i$ for each $i$ using random samples from $\mathcal{D}_\mu$. In Section 5.8, we prove that the error introduced by this approximation can be controlled if $\mathcal{D}_\mu$ is $c$-bounded.

As a simple application of the $\mu$–biased QFT, we show how to sample from the probability distribution defined by the coefficients of the single bit $\mu$–biased Fourier transform (recall that Parseval's equality holds in the $\mu$–biased setting).

**Lemma 3** ($\mu$–biased quantum Fourier sampling)**.** *Let $f : \{-1,1\} \to \{-1,1\}$ be a Boolean–valued function. Then, there exists a quantum algorithm with quantum membership oracle $O_f$ access that returns $v \in \{0,1\}$ with probability $\hat{f}_\mu^2(v)$. The algorithm requires exactly 1 $O_f$ query and 3 gates.*

*Proof.* Let $f'(x) = (1 - f(x))/2$ be the truth table representation of $f(x)$ with $(-1)^{f'(x)} = f(x)$. Apply Lemma 2 to get $\sum_b \sqrt{\mathcal{D}_\mu(b)} |b\rangle$. By querying the quantum membership oracle $O_{f'}$ (given access to $O_f$ this is equivalent to a relabelling of the qubits) one can make a phase query and obtain $\sum_b \sqrt{\mathcal{D}_\mu(b)} f(b) |b\rangle$ (note that $|f(b)| = 1$ and therefore the state is still normalised). Finally, applying the $\mu$–biased QFT results in

$$\begin{aligned}\sum_b \sqrt{\mathcal{D}_\mu(b)} f(b) \left( \sum_v \sqrt{\mathcal{D}_\mu(b)}\phi_{\mu,v}(b) |v\rangle \right) &= \sum_{b,v} \mathcal{D}_\mu(b) f(b) \phi_{\mu,v}(b) |b\rangle \\ &= \sum_v \hat{f}_\mu(v) |v\rangle .\end{aligned}$$

Measuring the state, one obtains $v$ with probability $\hat{f}_\mu^2(v)$. □

In order to use this result in the context of quantum PAC learning, we need to replace the membership oracle $O_f$ with a quantum example oracle. The following lemma, that extends Lemma 1 in [BJ98] to the $\mu$-biased case, serves this purpose. Differently from Lemma 3 we present directly the $n$-dimensional case.

**Lemma 4.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean-valued function. Then, there exists a quantum algorithm with quantum example oracle* $\mathrm{QEX}(f,\mu)$ *access that returns $a \in \{0,1\}^n$ with probability $\hat{f}_\mu^2(a)/2$. The algorithm requires exactly* 1 QEX *query and $O(n)$ gates.*

*Proof.* Let $f'(x) = (1 - f(x))/2$ be the truth table representation of $f(x)$ with $(-1)^{f'(x)} = f(x)$. Given access to $\mathrm{QEX}(f,\mu)$ it is always possible to construct an oracle for $\mathrm{QEX}(f',\mu)$ (this is equivalent to a relabelling of the qubits). Apply $\mathrm{QEX}(f',\mu)$ on a $|0,\ldots,0,0\rangle$ to get $\sum_x \sqrt{\mathcal{D}_\mu(x)}\,|x, f'(x)\rangle$. Then apply $H_\mu^n$ on the first register:

$$\sum_{x\in\{-1,1\}^n} \sum_{a\in\{0,1\}^n} \sqrt{\mathcal{D}_\mu(x)}\sqrt{\mathcal{D}_\mu(x)}\phi_{\mu,a}(x)\,|a, f'(x)\rangle\,.$$

An application of the standard QFT on the second register gives:

$$\begin{aligned}
&\sum_{x,a,z} \frac{1}{\sqrt{2}}(-1)^{f'(x)z}\mathcal{D}_\mu(x)\phi_{\mu,a}(x)\,|a,z\rangle \\
&\qquad = \frac{1}{\sqrt{2}}\left(\sum_a \hat{f}_\mu(a)\,|a,1\rangle + \sum_a \mathbb{E}_\mu[\phi_{\mu,a}(x)]\,|a,0\rangle\right) \\
&\qquad = \frac{1}{\sqrt{2}}\left(\sum_a \hat{f}_\mu(a)\,|a,1\rangle + \sum_a \mathbb{E}_\mu[\phi_{\mu,a}(x)\phi_{\mu,0}(x)]\,|a,0\rangle\right) \\
&\qquad = \frac{1}{\sqrt{2}}\left(\sum_a \hat{f}_\mu(a)\,|a,1\rangle + |0\ldots0,0\rangle\right),
\end{aligned}$$

where we used the orthonormality of the $\{\phi_{\mu,a}\}$ basis and $\phi_{\mu,0}(x) = 1$. Measuring the first register we obtain $|a,1\rangle$ with probability $\hat{f}_\mu^2(a)/2$. □

## 5.7 Quantum computation of $\mu$–biased Fourier spectrum

In this section we give a quantum algorithm to approximate the $\mu$-biased Fourier spectrum of a function in $\ell_\infty$ norm. This can be interpreted as a quantum version of the EKM algorithm. As a simple application of the quantum EKM algorithm we obtain the learnability of DNFs under product distributions in the quantum PAC model.

In order to bound the number of samples required to estimate a probability distribution in $\ell_\infty$ norm we use the Dvoretzky-Kiefer-Wolfowitz (DKW) theorem (see Appendix B for details). To make notation consistent, we write $\|F(x) - F_m(x)\|_\infty$ instead of $\max_{x\in\{-1,1\}^n} |F(x) - F_m(x)|$.

**Lemma 5.** *Let $f$ be a probability distribution over $\{-1,1\}^n$ and let $\tau > 0$, $\delta > 0$. Then, there exits an algorithm that with probability $1-\delta$ and for $m = O(\log(1/\delta)/\tau^2)$ outputs $f_m$ such that $\|f - f_m\|_\infty \leq \tau$.*

*Proof.* Let $\{e_1, \ldots, e_{2^n}\}$ be an ordering of elements of the Boolean hypercube $\{-1,1\}^n$. We have that

$$\begin{aligned}\|f - f_m\|_\infty &= \max_{\{e_1,\ldots,e_{2^n}\}} |f(e_i) - f_m(e_i)| \\ &= \max_{\{e_1,\ldots,e_{2^n}\}} |F(e_{i+1}) - F_m(e_{i+1}) - (F(e_i) - F_m(e_i))|.\end{aligned}$$

An application of the triangle inequality gives

$$\begin{aligned}&\max_{\{e_1,\ldots,e_{2^n}\}} |F(e_{i+1}) - F_m(e_{i+1}) - (F(e_i) - F_m(e_i))| \\ &\qquad \leq \max_{\{e_1,\ldots,e_{2^n}\}} |F(e_{i+1}) - F_m(e_{i+1})| + \max_{\{e_1,\ldots,e_{2^n}\}} |F(e_i) - F_m(e_i)| \\ &\qquad \leq 2\,\|F - F_m\|_\infty.\end{aligned}$$

By Theorem 11 we have that, with probability $1-\delta$,

$$\Pr(\|F - F_m\|_\infty \geq \gamma) \leq 2e^{-2m\gamma^2}.$$

Let $\gamma = \tau/2$, then

$$\Pr(\|f - f_m\|_\infty \leq \tau) \leq 1 - 2e^{-m\tau^2/2},$$

from which it is easy to see that $m = O(\log(1/\delta)/\tau^2)$. □

The combined application of Lemma 4 and Lemma 5 allows us to prove the following result:

**Theorem 8** (Quantum EKM algorithm)**.** *Let $f : \{-1,1\}^n \to \{-1,1\}$ be a Boolean–valued function and let $\epsilon > 0$, $\delta > 0$, $\mu \in (-1,1)^n$. Then, there exists a quantum algorithm with $\mathrm{QEX}(f,\mu)$ access that, with probability at least $1-\delta$, returns a succinctly represented vector $\tilde{f}_\mu$, such that $\|\hat{f}_\mu - \tilde{f}_\mu\|_\infty \leq \epsilon$ and $\|\tilde{f}_\mu\|_0 \leq 4/\epsilon^2$. The algorithm requires $O(\log^2(1/\delta)/\epsilon^8)$ $\mathrm{QEX}(f,\mu)$ queries and $O(n\log^2(1/\delta)/\epsilon^8)$ gates.*

*Proof.* We begin by estimating the $a$'s corresponding to the $\epsilon/2$-heavy Fourier coefficients of $f$. Let $\{p(a) = |\hat{f}_\mu(a)|^2\}$ be the probability distribution defined by the $\mu$–biased Fourier coefficients of $f$. Lemma 4 gives a procedure that, with 1 $\mathrm{QEX}(f,\mu)$ query and $O(n)$ gates, measures $|a,1\rangle$ with probability $q(a,1) = |\hat{f}_\mu(a)|^2/2$ and $|0\ldots0,0\rangle$ with probability $q(0,0) = 1/2$. Applying Lemma 5 on the distribution $q$ with $\tau = \epsilon^2/8$ we obtain that $O(\log(1/\delta)\epsilon^4)$ samples are required to have an estimate

$\|q-\tilde{q}\|_\infty \leq \epsilon^2/8$ with high probability. This implies that $\|\hat{f}_\mu^2 - \tilde{f}_\mu^2\|_\infty \leq \epsilon^2/4$. By selecting the characteristic vectors that correspond to coefficients such that $|\tilde{f}_\mu(a)|^2 > \epsilon^2/2$ (and discarding the element $|a,0\rangle$) we can output a list of $a$'s such that, with probability $\geq 1-\delta$, all the corresponding Fourier coefficients have $|\hat{f}_\mu(a)| > \epsilon$ and there are no coefficients such that $|\hat{f}_\mu(a)| \leq \epsilon/2$. By Parseval's equality this implies that the list may contain at most $4/\epsilon^2$ elements.

The final step requires the estimation of the Fourier coefficients. For a given $a$, the Fourier coefficient $\hat{f}_\mu(a) = \mathbb{E}_\mu[f(x)\chi_a(x)]$ can be obtained by sampling using the $\mathrm{QEX}(f,\mu)$ oracle to simulate $\mathrm{EX}(f,\mu)$ (to get an example $(x, f(x))$ it would suffice to measure a state prepared with $\mathrm{QEX}(f,\mu)$) in time $O(\log(1/\delta)/\epsilon^2)$ (the number of examples required for the estimate is a standard application of the Hoeffding bound).

The total number of examples required to estimate all the $\epsilon/2$-heavy Fourier coefficients of $f$ is $O(t\log^2(1/\delta)/\epsilon^8)$ by the union bound, where $t$ is the number of $\epsilon/2$-heavy Fourier coefficients. Because by Parseval's $t \leq 4/\epsilon^2$ we have that the final algorithm requires $O(\log^2(1/\delta)/\epsilon^8)$ $\mathrm{QEX}(f,\mu)$ queries and $O(n\log^2(1/\delta)/\epsilon^8)$ gates. □

Theorem 8 can be straightforwardly used in the method developed by [Fel12, Corollary 5.1] to obtain the learnability of DNF under product distributions.

**Corollary 9.** *Let $f$ compute an $s$-term DNF. Let $c \in (0,1]$ be a constant, let $\mathcal{D}_\mu$ be a $c$-bounded probability distribution and let $\mathrm{QEX}(f,\mu)$ be a quantum example oracle. Then, there exists a quantum algorithm with $\mathrm{QEX}(f,\mu)$ access that efficiently PAC learns $f$ over $\mathcal{D}_\mu$.*

We recall that the collection of the heavy Fourier coefficients of the DNF $f$ is the only step of Feldman's algorithm that requires MQ. The remaining of the algorithm makes use of the coefficients to construct a function $g$ that approximates $f$.

## 5.8 Error analysis

In the previous sections we assumed that the vector $\mu$ parametrising the product distribution was given to the learner. Here we prove that, if $\mathcal{D}_\mu$ is $c$-bounded, it is possible to estimate $\mu$ introducing an error that can be made small at a cost that scales polynomially in $n$. We recall that $\mu \in [-1+c, 1-c]^n$, $c \in (0,1]$, and $\mu_i = \mathbb{E}_\mu[x_i]$. By the Hoeffding bound we have that, with probability $1-\delta$, it is possible to approximate $\mu_i$ to $\epsilon$ accuracy $|\mu_i - \tilde{\mu}_i| \leq \epsilon$ using $O(\log(1/\delta)/\epsilon^2)$ samples.

We want to estimate the error introduced by approximating $H_\mu^n$ with $H_{\tilde{\mu}}^n$ (note that the $\mu$-biased QFT is now parametrised by $\tilde{\mu}$) in terms of the operator norm. Let $A$ be an operator, the operators norm $\|A\|$ is defined as:

$$\|A\| = \sup_{|\psi\rangle \neq 0} \frac{\|A|\psi\rangle\|}{\||\psi\rangle\|}.$$

The error analysis then requires to bound the quantity:

$$\left\|H_\mu^n - H_{\tilde{\mu}}^n\right\| \leq \gamma.$$

In order to prove the bound we introduce a useful lemma:

**Lemma 6.** *Let $A = A_n \cdots A_1$ be a product of unitary operators $A_j$. Assume that for every $A_j$ there exits an approximation $\tilde{A}_j$ such that $\|A_j - \tilde{A}_j\| \leq \epsilon_j$. The follow inequality holds*

$$\left\|A_n \cdots A_1 - \tilde{A}_n \cdots \tilde{A}_1\right\| \leq \sum_j \epsilon_j.$$

*Proof.* We prove by induction. The base step follows from the assumptions. For the inductive step let $X_k = A_k \cdots A_1$ and $\tilde{X}_k = \tilde{A}_k \cdots \tilde{A}_1$. Because the inductive hypothesis holds we have

$$\|X_k - \tilde{X}_k\| \leq \sum_{j=1}^{k} \epsilon_j.$$

By making use of the triangular inequality, the induction hypothesis, and noting that the product of unitaries is unitary we have

$$\begin{aligned}
\left\|A_{k+1}X_k - \tilde{A}_{k+1}\tilde{X}_k\right\| &= \left\|A_{k+1}\left(X_k - \tilde{X}_k\right) + \left(A_{k+1} - \tilde{A}_{k+1}\right)\tilde{X}_k\right\| \\
&\leq \left\|A_{k+1}\left(X_k - \tilde{X}_k\right)\right\| + \left\|\left(A_{k+1} - \tilde{A}_{k+1}\right)\tilde{X}_k\right\| \\
&= \|A_{k+1}\|\left\|X_k - \tilde{X}_k\right\| + \left\|A_{k+1} - \tilde{A}_{k+1}\right\|\left\|\tilde{X}_k\right\| \\
&\leq \sum_{j=1}^{k} \epsilon_j + \epsilon_{k+1} \\
&= \sum_{j=1}^{k+1} \epsilon_j.
\end{aligned}$$

□

Let $H_j = I \otimes \cdots I \otimes H_{\mu_j} \otimes I \otimes \cdots I$ and $\tilde{H}_j = I \otimes \cdots I \otimes H_{\tilde{\mu}_j} \otimes I \otimes \cdots I$. By Lemma 6 we have that

$$\left\|H_\mu^n - H_{\tilde{\mu}}^n\right\| \leq \sum_{j=1}^{n} \left\|H_j - \tilde{H}_j\right\|. \tag{5.5}$$

The bound on $\left\|H_j - \tilde{H}_j\right\|$ can be simplified using the following property of the operator norm $\|A \otimes B\| = \|A\|\|B\|$,

$$\begin{aligned}\left\|H_j - \tilde{H}_j\right\| &= \left\|I \otimes \cdots I \otimes (H_{\mu_j} - H_{\tilde{\mu}_j}) \otimes I \otimes \cdots I\right\| \\ &= \|I\| \cdots \|I\| \left\|(H_{\mu_j} - H_{\tilde{\mu}_j})\right\| \|I\| \cdots \|I\| \\ &= \left\|H_{\mu_j} - H_{\tilde{\mu}_j}\right\|.\end{aligned}$$

The problem of bounding Eq. 5.5 is then equivalent to bounding $\|H_{\mu_i} - H_{\tilde{\mu}_i}\|$. Let $|\psi\rangle = \sum_{x\in\{-1,1\}} \alpha_x |x\rangle$, we have that

$$\|(H_{\mu_i} - H_{\tilde{\mu}_i})|\psi\rangle\| = \left\| \sum_{x\in\{-1,1\}} \sum_{a\in\{0,1\}} \left(\sqrt{\mathcal{D}_{\mu_i}(x)}\phi_{\mu_i,a}(x) - \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,a}(x)\right) \alpha_x |a\rangle \right\|$$

where $\phi_{\mu,a}(x) = \prod_{i:a_i=1}(x_i - \mu_i)/\sqrt{1-\mu_i^2}$ and $\mathcal{D}_\mu(x) = \prod_{i:x_i=1}(1+\mu_i)/2 \prod_{i:x_i=-1}(1-\mu_i)/2$. We have to estimate the following quantity for a generic $a$, $x$

$$\begin{aligned}S &= \left|\sqrt{\mathcal{D}_{\mu_i}(x)}\phi_{\mu_i,a}(x) - \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,a}(x)\right| \\ &= \left|\frac{(x_i-\mu_i)\sqrt{1-\tilde{\mu}_i^2}\sqrt{\mathcal{D}_{\mu_i}(x)} - (x_i-\tilde{\mu}_i)\sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}}{\sqrt{1-\mu_i^2}\sqrt{1-\tilde{\mu}_i^2}}\right|.\end{aligned}$$

Recall that for every $i$ it holds $1-\mu_i^2 \geq c^2$, $1 - m\tilde{u}_i^2 \geq c^2$, $|\mu_i - \tilde{\mu}_i| \leq \epsilon$, $x_i \in \{-1,1\}$. By the triangle inequality we have that

$$\begin{aligned}S &\leq \frac{1}{c^2}\left|(x_i-\mu_i)\sqrt{1-\tilde{\mu}_i^2}\sqrt{\mathcal{D}_{\mu_i}(x)} - (x_i-\tilde{\mu}_i)\sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right| \\ &= \frac{1}{c^2}\Big|(x_i-\mu_i)\left(\sqrt{1-\tilde{\mu}_i^2}\sqrt{\mathcal{D}_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right) \\ &\qquad + (\tilde{\mu}_i-\mu_i)\sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\Big| \\ &\leq \frac{1}{c^2}\Bigg(\left|(x_i-\mu_i)\left(\sqrt{1-\tilde{\mu}_i^2}\sqrt{\mathcal{D}_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right)\right| \\ &\qquad + \left|(\tilde{\mu}_i-\mu_i)\sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right|\Bigg) \\ &\leq \frac{1}{c^2}\left((2-c)\left|\left(\sqrt{1-\tilde{\mu}_i^2}\sqrt{\mathcal{D}_{\mu_i}(x)} - \sqrt{1-\mu_i^2}\sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right)\right| + \epsilon\right) \\ &\leq \frac{1}{c^2}\left((2-c)\left(\left|\sqrt{\mathcal{D}_{\mu_i}(x)} - \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right| + \left|\sqrt{1-\mu_i^2} - \sqrt{1-\tilde{\mu}_i^2}\right|\right) + \epsilon\right).\end{aligned}$$

If we note that

$$\left|\sqrt{\mathcal{D}_{\mu_i}(x)} - \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\right| = \left|\frac{\tilde{\mu}_i - \mu_i}{2(\sqrt{\mathcal{D}_{\mu_i}(x)} + \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)})}\right| \leq \frac{\epsilon}{\sqrt{8c}}$$

and

$$\left|\sqrt{1-\mu_i^2}-\sqrt{1-\tilde{\mu}_i^2}\right| = \left|\frac{\tilde{\mu}_i^2-\mu_i^2}{\sqrt{1-\mu_i^2}+\sqrt{1-\tilde{\mu}_i^2}}\right| \leq \frac{\epsilon}{2c}$$

we have

$$S \leq t\epsilon, \tag{5.6}$$

where $t = ((2-c)(\frac{1}{\sqrt{8c}}+\frac{1}{2c})+1)/c^2$. By making use of Eq. 5.6 and noting that $\sum_x |\alpha_x| \leq 1$ we have

$$\begin{aligned}\|H_{\mu_i} - H_{\tilde{\mu}_i}\| &\leq \sum_{x,a} \left\| \left(\sqrt{\mathcal{D}_{\mu_i}(x)}\phi_{\mu_i,a}(x) - \sqrt{\mathcal{D}_{\tilde{\mu}_i}(x)}\phi_{\tilde{\mu}_i,a}(x)\right)\alpha_x |a\rangle \right\| \\ &\leq t\epsilon\left(2\sum_x |\alpha_x|\right) \\ &\leq 2t\epsilon.\end{aligned}$$

From which it follows that

$$\|H_\mu^n - H_{\tilde{\mu}}^n\| \leq 2nt\epsilon. \tag{5.7}$$

Eq. 5.7 guarantees that if one can approximate every $\mu_i$ with linear precision, *i.e.* with $\epsilon = O(1/n)$, it is possible to control the approximation error. Recall that by using the Hoeffding bound we can approximate $\mu_i$ to linear precision using $m = O(n^2 \log(1/\delta))$ examples.

Going beyond errors arising from the approximation of the vector $\mu$, we might assume that the quantum example oracle $\mathrm{QEX}(f, \mathcal{D})$ returns a state $|\Psi\rangle$ such that $\||\Psi\rangle - \sum_x \sqrt{\mathcal{D}(x)}\,|x, f(x)\rangle\| \leq \epsilon$ for a small $\epsilon > 0$. In this case, determining whether it is still possible to learn the function efficiently is contingent on a specific form of the error. For example, Grilo, Kerenidis, and Zijlstra showed that linear functions are still efficiently quantum learnable when the error distributions have support on a small interval around 0 [GKZ17].

# Chapter 6

# Conclusions

In this thesis we studied three questions that are relevant for assessing the potential and the limitations of algorithmic models of quantum systems. First, in Chapter 3 we worked on quantifying the computational resources required to learn quantum states and showed that, in the PAC framework, stabiliser states are efficiently learnable. Second, in Chapter 4 we tackled the problem of how to find an efficient representation of a quantum state from data and showed that VAEs can encode with a polynomial number of parameters states that are easy to simulate classically. Third, in Chapter 5 we turned to identifying learning problems that can only be solved efficiently by a quantum computer, and proved that the learnability of DNFs under product distributions provides such an example.

We conclude by discussing future directions in Section 6.1 and reflecting on the differences between scientific theories and algorithmic models in Section 6.2.

## 6.1 Future work

A general goal for future work is to combine ideas from each one of the three chapters in order to develop quantum learning algorithms with provable guarantees that are capable of computing efficiently some quantity of physical interest from training data. For example, can we train a quantum classifier for predicting the entanglement entropy of a state? En route to the final task there are several intermediate issues to be addressed:

- The quantum PAC model relies on the unrealistic assumption of access to a quantum state that encodes the domain set in a superposition whose amplitudes are the square root of the learning distribution. Are there interesting quantum learning models that do not require such strong form of access to the training

data? This question is particularly relevant in light of the results by Tang [Tan18] and Rudi et al. [Rud+18] who showed that, if there exists a classical data structure that grants the ability to sample entries of the training set according to their $\ell_2$ norm (an assumption that is comparable to preparing the training set in quantum superposition) then, it is possible to obtain classical algorithms with runtimes comparable to those of their quantum counterparts (note that at the moment the degree of the polynomials that upper bound the runtime of the quantum algorithms is significantly lower than their classical counterparts).

- Prior knowledge, also known as *inductive bias*, can dramatically increase the effectiveness of a learning algorithm. Contrary to common tasks in machine learning, such as image classification or language translation, in physics many properties of the system under study are well understood and can be hard-wired into the algorithm to increase its performance. In the context of neural-network quantum states, encoding symmetries is also important to have a better correspondence between the model and the real system. For a review of results in this direction we recommend [Car+19a].

- The maximum prediction accuracy of a supervised learning algorithm is determined by the number of examples in the training set. This implies that after a target accuracy has been reached, performing further computations does not improve the outcome. Hinging on this intuition, methods such as divide and conquer [ZDW13] (i.e. distributing portions of the training data onto separate machines, each solving a smaller learning problem, and then combining individual predictors into a joint one) or Nyström methods [RCR15] (i.e. constraining the learning problem to a small set of candidate predictors, obtained by randomly sampling directions in a larger, universal hypothesis space ) can greatly reduce the computational cost while keeping the statistical performance of the learned estimator essentially unaltered. It is therefore important to ask, are quantum algorithms still superior over classical ones when convergence rates are taken into account? If not, is it possible to design quantum algorithms with a better error dependency whilst maintaining a favourable scaling in the dimensions of the training set?

- Noise can play different, potentially beneficial, roles in learning problems. In a classical setting, it has been shown that noise can alleviate two of the most common model-fitting issues: local optima and generalisation performance.

Perturbing gradients can help with the former by 'jumping out' of local optima [Nee+15], whereas perturbing training inputs or outputs can improve the latter [Bis95]. The possibility of exploiting advantageously the effects of noise is particularly interesting in the context of quantum computation, especially for the first generation of non fault-tolerant quantum hardware. A separation between classical and quantum for a noisy learning problem (i.e. the learnability of parity functions under noise) was proved in [CSS15] and generalised to a more complex error model in [GKZ17]. Can we identify new, noisy, problems that only a learner equipped with quantum resources can solve?

## 6.2 Perspective: algorithmic models in physics

The mathematical modelling of observational data is a foundational component of every physical theory. Consider an apple falling from a tree. A successful theory must be capable of formulating quantitative temporal predictions on the position and momentum of the apple, given its initial location in the crown. But a physical theory is more than just a prediction system. It provides a human interpretable *explanation* of the rules governing the interaction of the variables and parameters of the model. In the case of the apple, Newtonian mechanics provides an explanation of its fall in terms of the concept of mass, gravitational force, and the second law of motion.

Although determining what constitutes a scientific explanation is still subject to an ongoing debate [Woo17], explanations appear to play an important role in the data collection process and in conjecturing new domains where the theory is applicable (Deutsch provides a fascinating and all-encompassing account of the power of explanations in [Deu11]). Coming back to the case of the apple, the same explanation used to model its fall can be applied to describe the motion of planets and, remarkably, to predict the existence of previously unobserved celestial bodies. This was the case of Neptune, which was discovered in 1846 after mathematical calculations predicted its existence.

Since their introduction as tools of scientific inquiry, computers have taken an important role in science. Computers are used to collect and analyse vast quantities of data, to perform complex symbolic calculations, and to simulate the behaviour of physical systems, often replacing experiments. But one area where computers have yet to establish a significant presence is in the process of scientific discovery. The development of new theories has not yet seen significant levels of automation. With the advent of effective algorithms to identify patterns and make predictions based

solely on data this may change. In addition to the various algorithms discussed in this thesis, there exist direct attempts at mimicking the process of scientific discovery using learning algorithms [SL09; Ite+18].

When, if ever, would we be able to call the outcomes of these algorithms 'theories'? Assuming that an algorithm can achieve the same predictive power of a theory, the problem becomes to determine whether an algorithmic model 'explains'. In science does it make sense to have models with excellent predictive power but that do not explain, in the sense of Newton's second law explaining the fall of the apple?

A similar dilemma arises in statistics where the essence of the debate was captured in a well-known essay by Breiman [Bre+01] (later developed by Norvig [Nor12]). In this paper, Breiman describes two major schools, or cultures, of designing statistical models. The first, and the one that more closely resembles the current scientific method, is the *data modelling* culture. In this framework, the analysis of data starts from a model that determines how the input variables interact with the model parameters to produce the response variables. The model, calibrated and validated with hypothesis tests, is then used for prediction tasks or to gain some information about the process that associates inputs and outputs. In science this form of modelling, whose ultimate product is a concise set of equations, has been tremendously successful, to a point of being referred to as 'unreasonably effective' in an essay by Wigner [Wig90].

The second, is the *algorithmic modelling* culture, which Breiman identifies with machine learning. Here, the algorithmic process that maps inputs and outputs is treated almost like a black box whose sole purpose is to replicate the input output relationship using some training data (note that some form of prior information is always present and, as formalised in the 'no free lunch' theorems by Wolpert, it is in fact required in order to generalise beyond the training data [Wol96]). In this modelling culture, the focus is not on extracting information on the relationships between inputs and outputs (although this is possible in certain cases) but to guarantee the best possible predictive accuracy.

As in the case of science, algorithmic models in statistics may lack human interpretability (Lipton analyses the different, and often conflicting, notions of interpretability in [Lip16]). Breiman addresses this point by noting that the explanations offered by a statistical model are more about the model itself than about the phenomenon under study (if a 'true' explanation exists at all) and suggests that we should be satisfied with the idea of modelling the input output relation in a way that generalises well.

The problem of whether algorithmic models provide satisfactory explanations is likely to continue to be subject to an intense debate. Notwithstanding, machine learning methods are making their way into physics and it is important to develop a theoretical understanding of their modelling capabilities.

# Appendix A

# Group Theory

The following presentation is adapted from [Lan02]. Let $S$ be a set. A mapping $S \times S \to S$ is called a multiplication rule. If $x, y$ are elements of $G$, then the image of the pair $(x, y)$ is called their *product* and will be denoted by $xy$.

**Definition 5.** *A group is a set $G$ with a multiplication rule that satisfies the following properties:*

- *(associative law) If $x, y, z$ are elements of $G$ then $x(yz) = (xy)z$.*
- *(identity element) There exists an element $e \in G$ such that $ex = xe = x$ for all $x \in G$. Such element is called the identity element of the group.*
- *(existence of inverse) For every $x \in G$ there exists an element $x^{-1} \in G$ such that $xx^{-1} = x^{-1}x = e$. Such element is called an inverse for $x$.*

*The group is said to be abelian if the rule of composition respects the following additional property:*

- *(commutativity) For every $x, y \in G$ then $xy = yx$.*

Let $G$ be a group. If the number of elements in the underlying set is finite then the group is said to be a *finite group*. The order of a finite group is the number of elements in the underlying set and is denoted as $|G|$.

**Definition 6.** *A subgroup $H$ of a group $G$ is a subset of $G$ that contains the identity element for $G$ and is closed under multiplication and inverse.*

Let $S$ be a subset of $G$. The set $S$ is a *generator* for the group $G$ if every element of $G$ can be expressed as product of elements of $S$. If $G$ is *generated* by $S$ we write $G = \langle S \rangle$. Given $x_1, \ldots, x_n \in G$, these elements generate a subgroup $\langle x_1, \ldots, x_n \rangle$. A set of generators is said to be *independent* or *minimal* if any proper subset of the generating set generates a strictly smaller subgroup.

**Lemma 7.** *Let $G$ be a finite group and let $\{x_1, \ldots, x_n\} \subseteq G$ be a set of independent generators. Then $n \leq \log_2(|G|)$.*

*Proof.* Let $H = \langle x_1, \ldots, x_n \rangle$ be the subgroup generated by $x_1, \ldots, x_n \in G$. Consider $x \in H$ and $y \notin H$. Then $xy \notin H$ because otherwise $x^{-1}xy = y \in H$ which we know is false by assumption. Thus, for each $x \in H$ there is an element $fy$ such that $fy \in \langle x_1, \ldots, x_n, y \rangle$ but not in $\langle x_1, \ldots, x_n \rangle$. Thus adding $y$ to $H$ at least doubles the size of the group being generated. Therefore we conclude that $n \leq \log_2(|G|)$. $\square$

# Appendix B

# Concentration of measure

The phenomenon of *concentration of measure* refers to the concentration of the sum of bounded, independent random variables around a certain value, usually the expected valued of the sum. The magnitude of the phenomenon is quantified by a variety of *concentration inequalities* that upper bound the probability that the sum deviates significantly from its mean. The bounds are often exponential as is the case for the two inequalities that we use: Hoeffding's inequality and the Dvoretzky-Kiefer-Wolfowitz inequality. For a comprehensive introduction to the phenomenon of concentration of measure we refer the reader to the textbook by Ledoux [Led01].

We present a version of Hoeffding's inequality adapted from Theorem 2.5 in [McD98].

**Theorem 10** (Hoeffding)**.** *Let $X_1, \ldots, X_m$ be a sequence of i.i.d. random variables, with $a \leq X_k \leq b$ for each $k$. Let $Z = \frac{1}{m}\sum_{k=1}^{m} X_k$ and $\mu = \mathbb{E}[Z]$. Then for any $\epsilon \geq 0$,*

$$\Pr\left(|Z - \mu| \geq \epsilon\right) \leq 2e^{-2m\epsilon^2/(b-a)^2}.$$

Hoeffding's inequality was originally proved for Bernoulli random variables by Chernoff [Che+52]. Hoeffding later extended it to general bounded independent random variables [Hoe63]. Because in computer science it is customary to apply the inequality to Bernoulli random variables, it is referred in the literature both as Chernoff's inequality and Hoeffding's inequality (sometime it is even referenced as the Chernoff-Hoeffding inequality). We refer to the inequality as Hoeffding's inequality in both the Bernoulli and general case.

Let $X_1, \ldots, X_m$ be a sequence of i.i.d. random variables drawn from a distribution $f$ on $\mathbb{R}$ with *Cumulative Distribution Function* (CDF) defined by $F(x) = \sum_{X_i \leq x} f(X_i)$, and let $x_1, \ldots, x_n$ be their realisations. Given a set $A$ the indicator function $\mathbf{1}_A : A \to \{0, 1\}$ takes values $f(x) = 0$ if $x \notin A$ and $f(x) = 1$ if $x \in A$. We denote the empirical probability distribution associated to $f(x)$ as $f_m(x) = \sum_{i=1}^{m} \mathbf{1}_{\{X_i = x\}}/m$ and

its empirical cumulative distribution as $F_m(x) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{X_i \leq x\}}$. The Dvoretzky-Kiefer-Wolfowitz (DKW) theorem bounds the number of samples required to estimate a cumulative distribution in $l_\infty$ norm. The DKW Theorem was first proposed by Dvoretzky-Kiefer-Wolfowitz in 1956 with an almost tight bound [DKW56]. In 1958 Birnbaum and McCarty conjectured that the inequality was tight [BM58]. This conjecture was proved by Massart in 1990 [Mas+90]. We present a version of the DKW theorem adapted from Theorem 11.6 in [Kos07].

**Theorem 11** (Dvoretzky-Kiefer-Wolfowitz)**.** *Let $X_1, \ldots, X_m$ be a sequence of i.i.d. random variables with cumulative distribution $F$ and empirical cumulative distribution defined by $F_m(x) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}_{\{X_i \leq x\}}$. Then for any $\epsilon \geq 0$,*

$$\Pr\left(\sup_{x \in \mathbb{R}} |F(x) - F_m(x)| > \epsilon\right) \leq 2e^{-2m\epsilon^2},$$

*for all $\epsilon > 0$.*

We note that DKW theorem holds also for the case where $F$ is discontinuous (the discontinuities can be infinite but must be countable). Therefore, the DKW theorem can be applied to the case of discrete random variables. A proof of this result is presented in [Kos07].

# Bibliography


[Aar07] S. Aaronson. "The learnability of quantum states". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. Vol. 463. 2088. The Royal Society. 2007, pp. 3089–3114.

[Aar15] S. Aaronson. "Read the fine print". In: *Nature Physics* 11.4 (2015), pp. 291–293.

[Aar18] S. Aaronson. "Shadow tomography of quantum states". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2018, pp. 325–338.

[AA11] S. Aaronson and A. Arkhipov. "The computational complexity of linear optics". In: *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM. 2011, pp. 333–342.

[AG04] S. Aaronson and D. Gottesman. "Improved simulation of stabilizer circuits". In: *Physical Review A* 70.5 (2004), p. 052328.

[AG08] S. Aaronson and D. Gottesman. *Identifying stabilizer states*. [Online; accessed 19-May-2019]. 2008. URL: `http://pirsa.org/08080052/`.

[Aar+18] S. Aaronson et al. "Online learning of quantum states". In: *Advances in Neural Information Processing Systems*. 2018, pp. 8976–8986.

[AC02] M. Adcock and R. Cleve. "A quantum Goldreich-Levin theorem with cryptographic applications". In: *Annual Symposium on Theoretical Aspects of Computer Science*. Springer. 2002, pp. 323–334.

[Ali95] F. Alizadeh. "Interior point methods in semidefinite programming with applications to combinatorial optimization". In: *SIAM Journal on Optimization* 5.1 (1995), pp. 13–51.

[Amb+02] A. Ambainis et al. "Dense quantum coding and quantum finite automata". In: *Journal of the ACM (JACM)* 49.4 (2002), pp. 496–511.

[AB09] M. Anthony and P. L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

[AG18] J. van Apeldoorn and A. Gilyén. "Improvements in quantum SDP-solving with applications". In: *arXiv preprint arXiv:1804.05058* (2018).

[Ape+17] J. van Apeldoorn et al. "Quantum SDP-solvers: Better upper and lower bounds". In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 403–414.

[AK07] S. Arora and S. Kale. “A combinatorial, primal-dual approach to semidefinite programs”. In: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM. 2007, pp. 227–236.

[AGS19] S. Arunachalam, A. B. Grilo, and A. Sundaram. “Quantum hardness of learning shallow classical circuits”. In: *arXiv preprint arXiv:1903.02840* (2019).

[AW17] S. Arunachalam and R. de Wolf. “Optimal Quantum Sample Complexity of Learning Algorithms”. In: *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*. 2017.

[Aru+15] S. Arunachalam et al. “On the robustness of bucket brigade quantum RAM”. In: *New Journal of Physics* 17.12 (2015), p. 123010.

[AS07] A. Atıcı and R. A. Servedio. “Quantum algorithms for learning and testing juntas”. In: *Quantum Information Processing* 6.5 (2007), pp. 323–348.

[AFK13] P. Awasthi, V. Feldman, and V. Kanade. “Learning using local membership queries”. In: *Conference on Learning Theory*. 2013, pp. 398–431.

[Bah61] R. R. Bahadur. “A representation of the joint distribution of responses to n dichotomous items”. In: *Studies in Item Analysis and Prediction* 6 (1961), pp. 158–168.

[Ban+99] K. Banaszek et al. “Maximum-likelihood estimation of the density matrix”. In: *Physical Review A* 61.1 (1999), p. 010304.

[Ban+18] L. Banchi et al. “Modelling non-markovian quantum processes with recurrent neural networks”. In: *New Journal of Physics* 20.12 (2018), p. 123030.

[Bar+05a] M. Barbieri et al. “Polarization-momentum hyperentangled states: Realization and characterization”. In: *Physical Review A* 72.5 (2005), p. 052110.

[Bar+05b] J. T. Barreiro et al. “Generation of hyperentangled photon pairs”. In: *Physical Review Letters* 95.26 (2005), p. 260501.

[BL98] P. L. Bartlett and P. M. Long. “Prediction, learning, uniform convergence, and scale-sensitive dimensions”. In: *Journal of Computer and System Sciences* 56.2 (1998), pp. 174–190.

[BLW96] P. L. Bartlett, P. M. Long, and R. C. Williamson. “Fat-shattering and the learnability of real-valued functions”. In: *Journal of Computer and System Sciences* 52.3 (1996), pp. 434–452.

[BM02] P. L. Bartlett and S. Mendelson. “Rademacher and Gaussian complexities: Risk bounds and structural results”. In: *Journal of Machine Learning Research* 3.Nov (2002), pp. 463–482.

[BPR07] F. Bauer, S. Pereverzev, and L. Rosasco. “On regularization algorithms in learning theory”. In: *Journal of Complexity* 23.1 (2007), pp. 52–72.

[BMM18] M. Belkin, S. Ma, and S. Mandal. “To understand deep learning we need to understand kernel learning”. In: *arXiv preprint arXiv:1802.01396* (2018).

[Bel91] M. Bellare. *The Spectral Norm of Finite Functions*. Tech. rep. Cambridge, MA, USA: Massachusetts Institute of Technology, 1991.

[BV97] E. Bernstein and U. Vazirani. “Quantum complexity theory”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1411–1473.

[BM58] Z. Birnbaum and R. McCarty. “A Distribution-Free Upper Confidence Bound for $\Pr\{Y < X\}$, Based on Independent Samples of $X$ and $Y$”. In: *The Annals of Mathematical Statistics* (1958), pp. 558–562.

[Bis95] C. M. Bishop. “Training with noise is equivalent to Tikhonov regularization”. In: *Neural Computation* 7.1 (1995), pp. 108–116.

[BKW03] A. Blum, A. Kalai, and H. Wasserman. “Noise-tolerant learning, the parity problem, and the statistical query model”. In: *Journal of the ACM (JACM)* 50.4 (2003), pp. 506–519.

[Blu+94] A. Blum et al. “Weakly learning DNF and characterizing statistical query learning using Fourier analysis”. In: *Proceedings of the twenty-sixth annual ACM Symposium on Theory of Computing*. ACM. 1994, pp. 253–262.

[Blu10] R. Blume-Kohout. “Optimal, reliable estimation of quantum states”. In: *New Journal of Physics* 12.4 (2010), p. 043034.

[BS17] F. G. Brandão and K. M. Svore. “Quantum speed-ups for solving semidefinite programs”. In: *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2017, pp. 415–426.

[Bra+17] F. G. Brandão et al. “Exponential quantum speed-ups for semidefinite programming with applications to quantum learning”. In: *arXiv preprint arXiv:1710.02581* (2017).

[Bre+01] L. Breiman et al. “Statistical modeling: The two cultures (with comments and a rejoinder by the author)”. In: *Statistical science* 16.3 (2001), pp. 199–231.

[BJS10] M. J. Bremner, R. Jozsa, and D. J. Shepherd. “Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy”. In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 467.2126 (2010), pp. 459–472.

[BJ98] N. H. Bshouty and J. C. Jackson. “Learning DNF over the uniform distribution using a quantum example oracle”. In: *SIAM Journal on Computing* 28.3 (1998), pp. 1136–1153.

[Buž04] V. Bužek. “Quantum Tomography from Incomplete Data via MaxEnt Principle”. In: *Quantum state estimation*. Springer, 2004, pp. 189–234.

[CT17] G. Carleo and M. Troyer. “Solving the quantum many-body problem with artificial neural networks”. In: *Science* 355.6325 (2017), pp. 602–606.

[Car+19a] G. Carleo et al. “Machine learning and the physical sciences”. In: *arXiv preprint arXiv:1903.10563* (2019).

[Car+19b] J. Carrasquilla et al. “Reconstructing quantum states with generative models”. In: *Nature Machine Intelligence* 1.3 (2019), p. 155.

[Che+17] J. Chen et al. “On the Equivalence of Restricted Boltzmann Machines and Tensor Network States”. In: *arXiv preprint arXiv:1701.04831* (2017).

[Che+52] H. Chernoff et al. “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations”. In: *The Annals of Mathematical Statistics* 23.4 (1952), pp. 493–507.

[Chi+19] N.-H. Chia et al. “Quantum-inspired classical sublinear-time algorithm for solving low-rank semidefinite programming via sampling approaches”. In: *arXiv preprint arXiv:1901.03254* (2019).

[Cil+18] C. Ciliberto et al. “Quantum machine learning: a classical perspective”. In: *Proc. R. Soc. A*. Vol. 474. 2209. The Royal Society. 2018, p. 20170551.

[Cla17] S. R. Clark. “Unifying Neural-network Quantum States and Correlator Product States via Tensor Networks”. In: *arXiv preprint arXiv:1710.03545* (2017).

[Cra+10] M. Cramer et al. “Efficient quantum state tomography”. In: *Nature Communications* 1 (2010), p. 149.

[CSS15] A. W. Cross, G. Smith, and J. A. Smolin. “Quantum learning robust against noise”. In: *Physical Review A* 92.1 (2015), p. 012327.

[DLD17] D.-L. Deng, X. Li, and S. Das Sarma. “Quantum Entanglement in Neural Network States”. In: *Physical Review X* 7.2 (2017), p. 021021.

[DLS16] D.-L. Deng, X. Li, and S. D. Sarma. “Exact Machine Learning Topological States”. In: *arXiv preprint arXiv:1609.09060* (2016).

[Deu11] D. Deutsch. *The beginning of Infinity: Explanations That Transform the World*. Penguin UK, 2011.

[DKB14] L. Dinh, D. Krueger, and Y. Bengio. “Nice: Non-linear independent components estimation”. In: *arXiv preprint arXiv:1410.8516* (2014).

[DKW56] A. Dvoretzky, J. Kiefer, and J. Wolfowitz. “Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator”. In: *The Annals of Mathematical Statistics* (1956), pp. 642–669.

[ES16] R. Eldan and O. Shamir. “The power of depth for feedforward neural networks”. In: *Conference on Learning Theory*. 2016, pp. 907–940.

[FU15] B. Fefferman and C. Umans. “The power of quantum fourier sampling”. In: *arXiv preprint arXiv:1507.05592* (2015).

[Fef14] W. J. Fefferman. “The power of quantum Fourier sampling”. PhD thesis. California Institute of Technology, 2014.

[Fel12] V. Feldman. “Learning DNF Expressions from Fourier Spectrum.” In: *COLT*. Vol. 8. 8.4. 2012, pp. 8–4.

[FW56] M. Frank and P. Wolfe. “An algorithm for quadratic programming”. In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110.

[Fre95] Y. Freund. “Boosting a weak learning algorithm by majority”. In: *Information and Computation* 121.2 (1995), pp. 256–285.

[FC94] C. A. Fuchs and C. M. Caves. “Ensemble-dependent bounds for accessible information in quantum mechanics”. In: *Physical Review Letters* 73.23 (1994), p. 3047.

[FJS91] M. L. Furst, J. C. Jackson, and S. W. Smith. “Improved learning of $AC^0$ functions”. In: *COLT*. Vol. 91. 1991, pp. 317–325.

[Gao+10] W.-B. Gao et al. “Experimental demonstration of a hyper-entangled ten-qubit Schrödinger cat state”. In: *Nature Physics* 6.5 (2010), p. 331.

[GD17] X. Gao and L.-M. Duan. “Efficient representation of quantum many-body states with deep neural networks”. In: *Nature Communications* 8.1 (2017), p. 662. (Visited on 11/20/2017).

[GMC14] H. J. García, I. L. Markov, and A. W. Cross. “On the geometry of stabilizer states”. In: *Quantum Information and Computation* 14 (2014), pp. 683–720.

[GLM08a] V. Giovannetti, S. Lloyd, and L. Maccone. “Architectures for a quantum random access memory”. In: *Physical Review A* 78.5 (2008), p. 052310.

[GLM08b] V. Giovannetti, S. Lloyd, and L. Maccone. “Quantum random access memory”. In: *Physical Review Letters* 100.16 (2008), p. 160501.

[Gla+17] I. Glasser et al. “Neural Networks Quantum States, String-Bond States and chiral topological states”. In: *arXiv preprint arXiv:1710.04045* (2017).

[GL89] O. Goldreich and L. A. Levin. “A hard-core predicate for all one-way functions”. In: *Proceedings of the twenty-first annual ACM symposium on Theory of computing*. ACM. 1989, pp. 25–32.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT press, 2016.

[Got96] D. Gottesman. “Class of quantum error-correcting codes saturating the quantum Hamming bound”. In: *Physical Review A* 54.3 (1996), p. 1862.

[Got97] D. Gottesman. “Stabilizer codes and quantum error correction”. PhD thesis. California Institute of Technology, 1997.

[Got98] D. Gottesman. “The Heisenberg representation of quantum computers”. In: *arXiv preprint arXiv:quant-ph/9807006* (1998).

[GHZ89] D. M. Greenberger, M. A. Horne, and A. Zeilinger. “Going beyond Bell’s theorem”. In: *Bell’s theorem, quantum theory and conceptions of the universe*. Springer, 1989, pp. 69–72.

[GKZ17] A. B. Grilo, I. Kerenidis, and T. Zijlstra. “Learning with errors is easy with quantum samples”. In: *arXiv preprint arXiv:1702.08255* (2017).

[Gro+10] D. Gross et al. “Quantum state tomography via compressed sensing”. In: *Physical Review Letters* 105.15 (2010), p. 150401.

[GLS12] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Vol. 2. Springer science & Business Media, 2012.

[GR02] L. Grover and T. Rudolph. “Creating superpositions that correspond to efficiently integrable probability distributions”. In: *arXiv preprint arXiv:quant-ph/0208112* (2002).

[GKW16] J. Gubernatis, N. Kawashima, and P. Werner. *Quantum Monte Carlo Methods*. Cambridge University Press, 2016.

[Haa16] J. Haah. “Algebraic methods for quantum codes on lattices”. In: *Revista Colombiana de Matemáticas* 50.2 (2016), pp. 299–349.

[Haa+17] J. Haah et al. “Sample-optimal tomography of quantum states”. In: *IEEE Transactions on Information Theory* 63.9 (2017), pp. 5628–5641.

[HHL09] A. W. Harrow, A. Hassidim, and S. Lloyd. “Quantum algorithm for linear systems of equations”. In: *Physical Review Letters* 103.15 (2009), p. 150502.

[HM17a] A. W. Harrow and A. Montanaro. “Quantum computational supremacy”. In: *Nature* 549.7671 (2017), p. 203.

[Haz08] E. Hazan. “Sparse approximate solutions to semidefinite programs”. In: *Latin American Symposium on Theoretical Informatics*. Springer. 2008, pp. 306–316.

[Hoe63] W. Hoeffding. “Probability Inequalities for Sums of Bounded Random Variables”. In: *Journal of the American Statistical Association* 58.301 (1963), pp. 13–30.

[Hra97] Z. Hradil. “Quantum-state estimation”. In: *Physical Review A* 55.3 (1997), R1561.

[HH91] S.-C. Huang and Y.-F. Huang. “Bounds on the number of hidden neurons in multilayer perceptrons”. In: *IEEE Transactions on Neural Networks* 2.1 (1991), pp. 47–55.

[HM17b] Y. Huang and J. E. Moore. “Neural network representation of tensor network and chiral states”. In: *arXiv preprint arXiv:1701.06246* (2017).

[Hua+11] Y.-F. Huang et al. “Experimental generation of an eight-photon Greenberger–Horne–Zeilinger state”. In: *Nature Communications* 2 (2011), p. 546.

[Ite+18] R. Iten et al. “Discovering physical concepts with neural networks”. In: *arXiv preprint arXiv:1807.10300* (2018).

[Jac94] J. Jackson. “An efficient membership-query algorithm for learning DNF with respect to the uniform distribution”. In: *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*. IEEE. 1994, pp. 42–53.

[Jac97] J. C. Jackson. “An efficient membership-query algorithm for learning DNF with respect to the uniform distribution”. In: *Journal of Computer and System Sciences* 55.3 (1997), pp. 414–440.

[JTY02] J. C. Jackson, C. Tamon, and T. Yamakami. “Quantum DNF learnability revisited”. In: *Lecture Notes in Computer Science* (2002), pp. 595–604.

[Jay57] E. T. Jaynes. “Information theory and statistical mechanics. II”. In: *Physical Review* 108.2 (1957), p. 171.

[KST09] A. T. Kalai, A. Samorodnitsky, and S.-H. Teng. “Learning and smoothed analysis”. In: *Foundations of Computer Science, 2009. FOCS’09. 50th Annual IEEE Symposium on*. IEEE. 2009, pp. 395–404.

[KRS18] V. Kanade, A. Rocchetto, and S. Severini. “Learning DNFs under product distributions via *mu*-biased quantum Fourier sampling”. In: *arXiv preprint arXiv:1802.05690* (2018).

[KPB17] R. Kaubruegger, L. Pastori, and J. C. Budich. “Chiral Topological Phases from Artificial Neural Networks”. In: *arXiv preprint arXiv:1710.04713* (2017).

[KV94] M. J. Kearns and U. Vazirani. *An Introduction to Computational Learning Theory*. MIT press, 1994.

[Kha93] M. Kharitonov. “Cryptographic hardness of distribution-specific learning”. In: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*. ACM. 1993, pp. 372–381.

[KB14] D. Kingma and J. Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).

[KW13] D. Kingma and M. Welling. “Auto-encoding variational bayes”. In: *arXiv preprint arXiv:1312.6114* (2013).

[Kit97] A. Y. Kitaev. “Quantum computations: algorithms and error correction”. In: *Russian Mathematical Surveys* 52.6 (1997), pp. 1191–1249.

[Kit+02] A. Y. Kitaev et al. *Classical and Quantum Computation*. 47. American Mathematical Society, 2002.

[KS01] A. R. Klivans and R. Servedio. “Learning DNF in time $2^{\tilde{O}(n^{1/3})}$”. In: *Proceedings of the thirty-third annual ACM symposium on Theory of computing*. ACM. 2001, pp. 258–265.

[KMS17] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz. “Deep learning in color: towards automated quark/gluon jet discrimination”. In: *Journal of High Energy Physics* 2017.1 (2017), p. 110.

[Kos07] M. R. Kosorok. *Introduction to Empirical Processes and Semiparametric Inference*. Springer Science & Business Media, 2007.

[KM93] E. Kushilevitz and Y. Mansour. "Learning decision trees using the Fourier spectrum". In: *SIAM Journal on Computing* 22.6 (1993), pp. 1331–1348.

[Lan02] S. Lang. *Algebra*. Vol. 211. Springer-Verlag, 2002.

[Led01] M. Ledoux. *The Concentration of Measure Phenomenon*. 89. American Mathematical Society, 2001.

[LSW15] Y. T. Lee, A. Sidford, and S. C.-w. Wong. "A faster cutting plane method and its implications for combinatorial and convex optimization". In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 1049–1065.

[Lei+04] D. Leibfried et al. "Toward Heisenberg-limited spectroscopy with multiparticle entangled states". In: *Science* 304.5676 (2004), pp. 1476–1478.

[Lev+19] Y. Levine et al. "Quantum entanglement in deep learning architectures". In: *Physical Review Letters* 122.6 (2019), p. 065301.

[LMN89] N. Linial, Y. Mansour, and N. Nisan. "Constant depth circuits, Fourier transform, and learnability". In: *30th Annual Symposium on Foundations of Computer Science*. IEEE. 1989, pp. 574–579.

[Lip16] Z. C. Lipton. "The mythos of model interpretability". In: *arXiv preprint arXiv:1606.03490* (2016).

[LW86] N. Littlestone and M. Warmuth. *Relating data compression and learnability*. Tech. rep. University of California, Santa Cruz, 1986.

[Low09] R. A. Low. "Learning and testing algorithms for the Clifford group". In: *Physical Review A* 80.5 (2009), p. 052314.

[Lyu05] V. Lyubashevsky. "The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem". In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2005, pp. 378–389.

[MHN13] A. L. Maas, A. Y. Hannun, and A. Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *Proc. ICML*. Vol. 30. 1. 2013.

[Mac03] D. J. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

[Man94] Y. Mansour. "Learning Boolean functions via the Fourier transform". In: *Theoretical Advances in Neural Computation and Learning*. Springer, 1994, pp. 391–424.

[Man95] Y. Mansour. "An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution". In: *Journal of Computer and System Sciences* 50.3 (1995), pp. 543–550.

[Mas+90] P. Massart et al. "The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality". In: *The Annals of Probability* 18.3 (1990), pp. 1269–1283.

[McA99] D. A. McAllester. “Some PAC-Bayesian theorems”. In: *Machine Learning* 37.3 (1999), pp. 355–363.

[McD98] C. McDiarmid. “Concentration”. In: *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998, pp. 195–248.

[MLP16] H. Mhaskar, Q. Liao, and T. Poggio. “Learning functions: When is deep better than shallow”. In: *arXiv preprint arXiv:1603.00988* (2016).

[Mon17a] A. Montanaro. “Learning stabilizer states by Bell sampling”. In: *arXiv preprint arXiv:1707.04012* (2017).

[Mon17b] A. Montanaro. “Quantum circuits and low-degree polynomials over”. In: *Journal of Physics A: Mathematical and Theoretical* 50.8 (2017), p. 084002.

[MO10] A. Montanaro and T. J. Osborne. “Quantum Boolean Functions”. In: *Chicago Journal of Theoretical Computer Science* 2010.1 (2010).

[MM17] A. Morningstar and R. G. Melko. “Deep Learning the Ising Model Near Criticality”. In: *arXiv preprint arXiv:1708.04622* (2017).

[Nee+15] A. Neelakantan et al. “Adding gradient noise improves learning for very deep networks”. In: *arXiv preprint arXiv:1511.06807* (2015).

[Nes08] M. Nest. “Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond”. In: *arXiv preprint arXiv:0811.0898* (2008).

[NN94] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Vol. 13. SIAM, 1994.

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[NU98] M. P. Nightingale and C. J. Umrigar. *Quantum Monte Carlo Methods in Physics and Chemistry*. 525. Springer science & Business Media, 1998.

[Nom+17] Y. Nomura et al. “Restricted Boltzmann machine learning for solving strongly correlated quantum systems”. In: *Physical Review B* 96.20 (2017), p. 205152.

[Nor12] P. Norvig. “Colorless green ideas learn furiously: Chomsky and the two cultures of statistical learning”. In: *Significance* 9.4 (2012), pp. 30–33.

[ODo14] R. O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

[OW16] R. O’Donnell and J. Wright. “Efficient quantum tomography”. In: *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. ACM. 2016, pp. 899–912.

[OKK16] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. “Pixel recurrent neural networks”. In: *arXiv preprint arXiv:1601.06759* (2016).

[Orú14] R. Orús. “A practical introduction to tensor networks: Matrix product states and projected entangled pair states”. In: *Annals of Physics* 349 (2014), pp. 117–158.

[PBJ12] J. Paisley, D. Blei, and M. Jordan. “Variational Bayesian inference with stochastic search”. In: *arXiv preprint arXiv:1206.6430* (2012).

[Pat+18] J. Pathak et al. “Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach”. In: *Physical Review Letters* 120.2 (2018), p. 024102.

[Per+06] D. Perez-Garcia et al. “Matrix product state representations”. In: *arXiv preprint arXiv:quant-ph/0608197* (2006).

[Rag+17] M. Raghu et al. “On the expressive power of deep neural networks”. In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org. 2017, pp. 2847–2854.

[RMW14] D. J. Rezende, S. Mohamed, and D. Wierstra. “Stochastic Backpropagation and Approximate Inference in Deep Generative Models”. In: *International Conference on Machine Learning*. 2014, pp. 1278–1286.

[Ric+14] P. Richerme et al. “Non-local propagation of correlations in quantum systems with long-range interactions”. In: *Nature* 511.7508 (2014), pp. 198–201.

[Roc18] A. Rocchetto. “Stabiliser states are efficiently PAC-learnable”. In: *Quantum Information and Computation* 18.7&8 (2018).

[RBL16] A. Rocchetto, S. C. Benjamin, and Y. Li. “Stabilizers as a design tool for new forms of the Lechner-Hauke-Zoller annealer”. In: *Science Advances* 2.10 (2016), e1601246.

[Roc+18] A. Rocchetto et al. “Learning hard quantum distributions with variational autoencoders”. In: *npj Quantum Information* 4.1 (2018), p. 28.

[Roc+19] A. Rocchetto et al. “Experimental learning of quantum states”. In: *Science Advances* 5.3 (2019), eaau1946.

[RCR15] A. Rudi, R. Camoriano, and L. Rosasco. “Less is more: Nyström computational regularization”. In: *Advances in Neural Information Processing Systems*. 2015, pp. 1657–1665.

[Rud+18] A. Rudi et al. “Approximating Hamiltonian dynamics with the Nyström method”. In: *arXiv preprint arXiv:1804.02484* (2018).

[Rud09] T. Rudolph. “Simple encoding of a quantum circuit amplitude as a matrix permanent”. In: *Physical Review A* 80.5 (2009), p. 054302.

[SH09] R. Salakhutdinov and G. Hinton. “Deep boltzmann machines”. In: *Artificial Intelligence and Statistics*. 2009, pp. 448–455.

[SL09] M. Schmidt and H. Lipson. “Distilling free-form natural laws from experimental data”. In: *Science* 324.5923 (2009), pp. 81–85.

[Sch11] U. Schollwöck. “The density-matrix renormalization group in the age of matrix product states”. In: *Annals of Physics* 326.1 (2011), pp. 96–192.

[SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[Smo86] P. Smolensky. *Information Processing in Dynamical Systems: Foundations of Harmony Theory*. Tech. rep. University of Colorado Boulder, 1986.

[Søn+16] C. K. Sønderby et al. “Ladder variational autoencoders”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 3738–3746.

[Suz93] M. Suzuki. *Quantum Monte Carlo Methods in Condensed Matter Physics*. World Scientific, 1993.

[Tan18] E. Tang. “A quantum-inspired classical algorithm for recommendation systems”. In: *arXiv preprint arXiv:1807.04271* (2018).

[Tel16] M. Telgarsky. “Benefits of depth in neural networks”. In: *arXiv preprint arXiv:1602.04485* (2016).

[Tor+17] G. Torlai et al. “Many-body quantum state tomography with neural networks”. In: *arXiv preprint arXiv:1703.05334* (2017).

[Val84] L. G. Valiant. “A theory of the learnable”. In: *Communications of the ACM* 27.11 (1984), pp. 1134–1142.

[Val02] L. G. Valiant. “Quantum circuits that can be simulated classically in polynomial time”. In: *SIAM Journal on Computing* 31.4 (2002), pp. 1229–1254.

[Vap13] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer science & Business Media, 2013.

[Ver90] K. A. Verbeurgt. “Learning DNF Under the Uniform Distribution in Quasi-Polynomial Time.” In: *COLT*. 1990, pp. 314–326.

[VMC08] F. Verstraete, V. Murg, and J. I. Cirac. “Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems”. In: *Advances in Physics* 57.2 (2008), pp. 143–224.

[Whi92] S. R. White. “Density matrix formulation for quantum renormalization groups”. In: *Physical Review Letters* 69.19 (1992), p. 2863.

[Wig90] E. P. Wigner. “The unreasonable effectiveness of mathematics in the natural sciences”. In: *Mathematics and Science*. World Scientific, 1990, pp. 291–306.

[Wol08] R. de Wolf. “A brief introduction to Fourier analysis on the Boolean cube”. In: *Theory of Computing* (2008), pp. 1–20.

[Wol96] D. H. Wolpert. “The lack of a priori distinctions between learning algorithms”. In: *Neural Computation* 8.7 (1996), pp. 1341–1390.

[Woo17] J. Woodward. “Scientific Explanation”. In: *The Stanford Encyclopedia of Philosophy*. Ed. by E. N. Zalta. Fall 2017. Metaphysics Research Lab, Stanford University, 2017.

[Zha+16] C. Zhang et al. “Understanding deep learning requires rethinking generalization”. In: *arXiv preprint arXiv:1611.03530* (2016).

[ZDW13] Y. Zhang, J. Duchi, and M. Wainwright. “Divide and conquer kernel ridge regression”. In: *Conference on Learning Theory*. 2013, pp. 592–617.

[ZPF16] L. Zhao, C. A. Pérez-Delgado, and J. F. Fitzsimons. “Fast graph operations in quantum computation”. In: *Physical Review A* 93.3 (2016), p. 032314.