Proceedings of Machine Learning Research vol 247:1–24, 2024 37th Annual Conference on Learning Theory
Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 142 (2023)

Provable Advantage in Quantum PAC Learning

Wilfred Salmon WAS29@CAM.AC.UK
DAMTP, Centre for Mathematical Sciences, University of Cambridge

Sergii Strelchuk SS870@CAM.AC.UK
DAMTP, Centre for Mathematical Sciences, University of Cambridge

Department of Computer Science, University of Warwick

Tom Gur TG508@CAM.AC.UK
Department of Computer Science and Technology, University of Cambridge

Editors: Shipra Agrawal and Aaron Roth

Abstract

We revisit the problem of characterising the complexity of Quantum PAC learning, as introduced by Bshouty and Jackson [SIAM J. Comput. 1998, 28, 1136–1153]. Several quantum advantages have been demonstrated in this setting, however, none are generic: they apply to particular concept classes and typically only work when the distribution that generates the data is known. In the general case, it was recently shown by Arunachalam and de Wolf [JMLR, 19 (2018) 1-36] that quantum PAC learners can only achieve constant factor advantages over classical PAC learners.

We show that with a natural extension of the definition of quantum PAC learning used by Arunachalam and de Wolf, we can achieve a generic advantage in quantum learning. To be precise, for any concept class $\mathcal{C}$ of VC dimension d, we show there is an (ϵ, δ)-quantum PAC learner with sample complexity

$$O\left(\frac{1}{\sqrt{\epsilon}}\left[d + \log\left(\frac{1}{\delta}\right)\right]\log^9(1/\epsilon)\right).$$

Up to polylogarithmic factors, this is a square root improvement over the classical learning sample complexity. We show the tightness of our result by proving an $\Omega(d/\sqrt{\epsilon})$ lower bound that matches our upper bound up to polylogarithmic factors.

1. Introduction

Probably approximately correct (PAC) learning Valiant (1984) is a fundamental model of machine learning. One is given a set of functions $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}} = \{f : \mathcal{X} \to \{0,1\}\}$, called a concept class, that encodes the structure of a learning problem (for example, functions that only depend on the hamming weight of their input). Given labelled examples from an unknown concept $c \in \mathcal{C}$, we are tasked with learning an approximation to c.

We model the data that the learning algorithm receives by an unknown probability distribution $\mathcal{D}$ on $\mathcal{X}$, and say that a hypothesis $h : \mathcal{X} \to \{0,1\}$ is ϵ-approximately correct if the probability that it differs from c is at most ϵ. To be precise, a hypothesis $h \in \{0,1\}^{\mathcal{X}}$ is said to be ϵ-approximately correct if $\mathbb{P}_{X\sim\mathcal{D}}\left[h(X) \neq c(X)\right] \leq \epsilon$.

A learning algorithm $\mathcal{A}$ draws independent samples $(X, c(X))$, where X is distributed according to $\mathcal{D}$, and then outputs a hypothesis h. The algorithm $\mathcal{A}$ is an (ϵ, δ)-learner if, with probability at least $1-\delta$ over the random samples, it outputs a ϵ-approximately correct

ISSN 1433-8092

hypothesis.

The amount of "structure" possessed by $\mathcal{C}$ is characterised by its Valiant-Chapernikis (VC) dimension Vapnik and Chervonenkis (1971), denoted d. For a subset $Y \subseteq X$, we define $\mathcal{C}|_Y := \{c|_Y : c \in \mathcal{C}\}$ as the restriction of the concept class to Y. We say that $\mathcal{C}$ shatters Y if $\mathcal{C}|_Y = \{0,1\}^Y$, i.e., if all possible labellings of Y appear in concepts in $\mathcal{C}$. Then, d is the maximum size of a shattered set, that is $d = \max\{|Y| : Y \text{ is shattered by } \mathcal{C}\}$.

Over a period of 27 years Blumer et al. (1989); Hanneke (2016), the exact asymptotic scaling of the minimum number of samples required by an (ϵ, δ)-learner was found to be

$$\Theta\left[\frac{1}{\epsilon}\left(d + \log\left(\frac{1}{\delta}\right)\right)\right], \tag{1}$$

thereby characterising the complexity of classical PAC learning.

In 1995, Bshouty and Jackson Bshouty and Jackson (1995) considered a generalisation of PAC learning to the quantum setting Arunachalam and de Wolf (2017). Here, instead of receiving independent identically distributed samples $(X, C(X))$, one receives independent copies of a quantum state

$$|\psi_c\rangle = \sum_{x \in \mathcal{X}} \sqrt{\mathcal{D}(x)}\, |x\ c(x)\rangle\,, \tag{2}$$

known as a *quantum sample*. In particular, measuring such a state in the computational basis gives a sample $(X, C(X))$. In turn, instead of counting the number of samples, the quantum sample complexity is the number of copies of the state given to the quantum learning algorithm.

The Quantum PAC model is instrumental in understanding the limits of other quantum cryptographic and computational tasks. For instance, in Arunachalam et al. (2021), a connection between differential privacy and PAC learnability of quantum states was established, and recently Cai et al. (2022) used the PAC framework to investigate the complexity of learning parameterised quantum circuits, which are ubiquitous in variational quantum algorithms where they are used for quantum state preparation.

In the special case of quantum PAC learning under the uniform distribution, it has been shown that one can obtain quantum sample complexity advantages in specific learning tasks, such as learning Fourier basis functions Bernstein and Vazirani (1997), DNF formulae Bshouty and Jackson (1995), and k-juntas Atici and Servedio (2007). These advantages rely on Fourier sampling, in which one applies the Hadamard transform on every qubit followed by a measurement of the resulting state in the computational basis. One observes a bit string s with probability given by its squared Fourier coefficient $|\hat{c}_s|^2$ and can thus directly infer properties of the Fourier spectrum of the unknown function. However, such advantages rely on the distributions $\mathcal{D}$ being (approximately) uniform.

The general quantum PAC learning model, with an arbitrary and unknown distribution $\mathcal{D}$, was studied by Arunachalam and de Wolf Arunachalam and de Wolf (2017, 2018), who showed that the quantum sample complexity has exactly the same asymptotic scaling as the classical learning complexity, ruling out everything but constant factor prospective advantages.

Thus, most recent literature has focused identifying advantages only in suitably restricted versions of the quantum PAC model Atici and Servedio (2007); Pirnay et al. (2023). Nevertheless, such models have demonstrated remarkable utility when assessing the complexity of learning quantum states, channels Aaronson (2007); Chung and Lin (2018); Rocchetto (2018), and measurements Padakandla and Magner (2022); Cheng et al. (2015) in quantum theory with lower bounds on query complexity established in Zhang (2010).

Here, we consider a natural and less restrictive version of the quantum PAC learning model. Instead of access to copies of the state $|\psi_c\rangle$, we assume that we have access to the quantum circuit that generates it, similarly in spirit to Kothari and O'Donnell (2023); van Apeldoorn et al. (2023). That is, we assume one has access to a quantum circuit Q_c that generates a quantum sample $|\psi_c\rangle$ (for example, as a decomposition into one and two-qubit gates) and thus can implement Q_c and $Q_c^\dagger$. Given this natural adjustment to the input access of quantum PAC learning algorithms, we can revisit the question of whether strong generic (beyond constant-factor) quantum advantages are possible for quantum PAC learning.

1.1. Our results

In this paper, we show that there is a square root advantage (up to polylogarithmic factors) for quantum PAC learning over classical PAC learning in the full, general model. Our main result (see section 5) is summarised by the following theorem.

> **Theorem 1:** Let $\mathcal{C}$ be a concept class with VC dimension d. Then, for every $\epsilon, \delta > 0$, there exists a (ϵ, δ)-quantum PAC learner for $\mathcal{C}$ that makes at most
>
> $$O\left(\frac{1}{\sqrt{\epsilon}}\left[d + \log\left(\frac{1}{\delta}\right)\right]\log^9(1/\epsilon)\right), \tag{3}$$
>
> calls to an oracle that generates a quantum sample (Q_c), or its inverse ($Q_c^\dagger$).

In comparison, the optimal classical PAC learning complexity (and quantum PAC complexity given access to copies of $|\psi_c\rangle$ Arunachalam and de Wolf (2018)) is given in equation (5). Thus, our upper bound is a square root improvement (up to polylogarithmic factors) over the best possible classical learning algorithm. In fact, we show that this upper bound is essentially tight, up to polylogarithmic factors, as captured by the following theorem.

> **Theorem 2:** Let $\mathcal{C}$ be a concept class with VC dimension d. Then, for a sufficiently small constant $\delta > 0$ and for all $\epsilon > 0$, any quantum (ϵ, δ)-learner for $\mathcal{C}$ makes at least
>
> $$\Omega\left(\frac{d}{\sqrt{\epsilon}}\right) \tag{4}$$
>
> calls to an oracle that generates a quantum sample (Q_c), or its inverse ($Q_c^\dagger$).

1.2. Technical overview

Our starting point is the observation that the lower bound of Arunachalam and de Wolf Arunachalam and de Wolf (2018) implicitly rests on the assumption that a quantum learning algorithm must not depend on the underlying concept, and it can thus be represented by a (concept independent) POVM. They then reduce the problem of PAC learning to that of state discrimination (where the POVM is state-independent). However, if we allow for the common assumption that the algorithm has access to an oracle Q_c generating $|\psi_c\rangle$, the proof of the lower bound no longer holds[1]. If the POVM describing the algorithm calls the oracle, it, *by definition*, depends on the underlying concept. Thus, one cannot reduce the problem to that of state discrimination, where it is assumed that the POVM is independent of the input state.

If one implements Q_c on some physical device (for example, as a series of one and two-qubit gates), it is natural to assume that one can also implement the inverse process $Q_c^\dagger$ (for example, by reversing the order of the gates and replacing each by its inverse). Thus, we argue that if one has access to the state $|\psi_c\rangle$ it is natural to also consider the situation in which one also has access to Q_c and $Q_c^\dagger$. Indeed, this setting has recently received significant attention van Apeldoorn et al. (2023); Haah et al. (2023).

Given access to Q_c and $Q_c^\dagger$, it is tempting to attempt techniques such as Grover search and amplitude amplification, which often achieve quadratic quantum advantages. Consider, for example, the simplest possible concept class $\mathcal{C} = \{0,1\}^{\mathcal{X}}$: the set of all possible classifiers. It is known that a classical worst-case distribution for this class is a "perturbed" delta-function Arunachalam and de Wolf (2018), where there is a marked element $x_0 \in \mathcal{X}$ with probability $\mathcal{D}(x_0) = 1-4\epsilon$, and all other elements have equal probability. Roughly speaking, to (ϵ,δ)-learn $\mathcal{C}$, one must learn a fraction of $3/4$ of the values of c. However, it takes on average $O(1/\epsilon)$ samples to return an x that *isn't* x_0 and thus the classical learning query complexity is $\Omega(|\mathcal{X}|/\epsilon)$. In this case, one could repeatedly run Grover's search, marking any state $|x\ b\rangle$ as good if we have not yet learnt $c(x)$. With Grover search, it only takes $O(1/\sqrt{\epsilon})$ oracle calls to return an x that is not x_0 and thus we see the quantum query complexity is $O(|\mathcal{X}|/\sqrt{\epsilon})$, the desired quadratic improvement. Therefore, we already outperform the lower bound of Arunachalam and de Wolf Arunachalam and de Wolf (2018).

Note that the method above does not immediately generalise to other concept classes. For example, consider the concept class $\mathcal{C} = \{c \in \{0,1\}^{\mathcal{X}} : |c^{-1}(\{1\})| = d\}$, the class of classifiers with exactly d inputs that map to 1, and take $\mathcal{D}$ to be the uniform distribution on $\mathcal{X}$. If $|\mathcal{X}|$ is very large, then most unseen x's will have $c(x) = 0$ and thus the above approach is uninformative. Instead, one should mark a state $|x\ b\rangle$ as good if $b = 1$. In this way, one can search for the inputs $x \in \mathcal{X}$ that have $c(x) = 1$ and hence deduce c. This will also lead to a quadratic quantum advantage.

However, for general concept classes, it is less clear what to search for. One could run the Halving algorithm, where we mark a state $|x\ y\rangle$ as good if the majority of the concepts $h \in \mathcal{C}$ that are consistent with the data so far have $h(x) = 1-y$. In this case, every time the Grover algorithm succeeds, one would eliminate at least half of the concepts in $\mathcal{C}$. However, this leads to a $\log|\mathcal{C}|$ factor in the learning complexity, which can be as large as $d\log|\mathcal{X}|$, i.e., arbitrarily larger than d (the VC dimension of $\mathcal{C}$). Thus, even under the simplifying

1. Since the state $|\psi_c\rangle$ must be produced by some process, this assumption is quite minimal.

assumption of the uniform distribution, it is unclear how to attempt to use Grover's search to obtain a quantum advantage.

Nevertheless, we show that one can achieve a square root quantum advantage in the general case. As a first step, we use the technique of equivalence queries Gluch and Urbanke (2021) (also known as random counterexamples). An equivalence query is an alternative learning oracle to the traditional PAC oracle, in which one submits a candidate hypothesis $h \in \{0,1\}^{\mathcal{X}}$. If $h = c$, then the oracle outputs YES, otherwise it produces a labelled counterexample $(X, c(X))$ where

(i) $h(X) \neq c(X)$.

(ii) X is distributed according to $\mathbb{P}(y) = \mathcal{D}(y)/\mathcal{D}(\{x : c(x) \neq h(x)\})$.

Observe that by marking a state $|x\ y\rangle$ as good if $h(x) = 1 - y$, we can see how to implement an equivalence query using Grover search, and thus one can hope to use this tool from classical learning theory to achieve an advantage. However, when one removes the simplifying assumption of a known distribution, further problems arise.

For a generic distribution, we do not know $\mathcal{D}(x)$ for any $x \in \mathcal{X}$ and therefore one cannot run exact Grover search. Instead, we consider a well-studied technique Boyer et al. (1998), in which one makes a random number of M queries to the Grover oracle, where M is uniformly distributed between 0 and a chosen threshold T_G. This search succeeds with non-negligible probability if the amplitude of the projection of the initial state onto the subspace spanned by the "good" states (the "good" subspace) is $\Omega(1/T_G)$. For an equivalence query h, this amplitude is $\sqrt{\mathcal{D}(\{x : c(x) \neq h(x)\})}$, which could be arbitrarily small (as $\mathcal{D}$ is arbitrary). Hence, it may take an arbitrarily large (expected) number of iterations of Grover's search (and hence oracle calls) to run a classical equivalence query learning algorithm.

To solve this issue, we show how to use equivalence queries that succeed with a constant probability, called imperfect equivalence queries, to PAC learn a concept. We can then run these imperfect equivalence queries using Grover search. We use a classical (ideal) equivalence query algorithm, replacing equivalence queries with repeated imperfect equivalence queries, but with a maximum imperfect equivalence query budget R. Suppose that the algorithm requires equivalence queries to hypotheses $h_1, \ldots h_k$. If all of the successfully run an equivalence query for every hypothesis, then the classical algorithm succeeds, and we use its output. Otherwise, we hit the imperfect equivalence query budget R and must terminate the classical algorithm early. By choosing R sufficiently large, we can be sure that if we hit the budget, most of the imperfect equivalence queries were spent on hypotheses h_i that are "close" to c (and hence have a low chance of the Grover search succeeding). Thus if we take the "average" of the hypotheses h_i weighted by the number of imperfect equivalence queries spent on each hypothesis, we also output a classifier close to c.

To conclude the section, we sketch a proof of our lower bound. We consider an arbitrary concept class $\mathcal{C}$ of VC dimension d. We note that there is a shattered set $Y \subseteq \mathcal{X}$ of size d, and take $\mathcal{D}$ to be a "perturbed" delta-function distribution on Y. We can thus think of concepts c in $\mathcal{C}$ as bit strings of length d, where the bit string describes c's action on Y. Since Y is shattered by $\mathcal{C}$, all possible bit strings will appear. Any candidate PAC algorithm must be able to recover most of the bit string with high probability. We reduce to a known problem by introducing a weak phase-kickback oracle for the bit string, which we use to

implement the PAC oracle. We can then use a standard lower bound van Apeldoorn et al. (2023) on recovering a bit string with high probability using a weak phase kickback oracle.

1.3. Discussion of access models

Since Valiant introduced PAC learning, a plethora of access models have been put forward, inspired by different real-world scenarios. We provide a brief discussion of the access model considered in this paper (access to Q_c and $Q_c^\dagger$), and its relationship to three other popular models. We compare to labelled examples $(X, c(X))$, quantum samples $|\psi_c\rangle$, and membership queries - where a learner submits an $x \in \mathcal{X}$ and receives $c(x)$ in return.

Labelled examples are the most commonly considered access model in the machine learning literature. Even if learning complexity is polynomial, there may be no (time) efficient learner Pitt and Valiant (1988). Some classes have time-efficient learners with labelled examples and membership queries, but not with only labelled examples Angluin and Kharitonov (1991). Curiously, we are not aware of a work that gives a learning complexity lower bound on labelled examples and membership queries, but we believe that the standard labelled examples lower bound setup (see Arunachalam and de Wolf (2017)) should give the same lower bound as only labelled examples. We note that membership queries can be much stronger than either labelled examples, quantum samples or our model; given labelled examples or quantum samples, it takes on average $1/\mathcal{D}(x)$ queries to find $c(x)$. Given our access model, one can perform amplitude amplification such that only $1/\sqrt{\mathcal{D}(x)}$ samples are required, but this may still be large. Indeed, if $\mathcal{D}(x) = 0$, then a membership query of x is impossible using labelled examples, membership queries or our model. Thus, membership queries are incomparable with our model; they provide some advantage, but this is in time complexity, and they do not, in general, provide an information theoretic advantage.

Quantum samples were introduced as the natural quantum generalisation of labelled examples. As noted, for specific concept classes and distributions, they are known to give learning (and time) complexity advantages over labelled examples. However, in general, quantum samples and labelled examples have the same asymptotic learning complexity. Furthermore, in practice it is unclear how to prepare $|\psi_c\rangle$, given that it depends on c and $\mathcal{D}$. Whether or not this is reasonable depends on how the labelled examples are classically generated - if they are completely "black-box", then it seems unlikely that quantum samples will be an appropriate resource without very strong quantum data loading subroutines, such as QRAM. If c and $\mathcal{D}$ are "white-box", i.e. there is some circuit producing them, then an appropriate quantisation procedure will lead to Q_c. Given a circuit description of Q_c, one can theoretically perform $Q_c^\dagger$. Thus, in most scenarios where quantum samples are sensible for learning classical data, our model is also reasonable.

A more promising case for quantum samples is when data is inherently quantum. Suppose a quantum process produces $\left|\widetilde{\psi_c}\right\rangle = \sum_x \sqrt{\mathcal{D}(x)}\,|x, c(x), g(x)\rangle$, where $g(x)$ is some extraneous function. In this case, learning c allows us to make physical predictions of the $c(x)$ register given the x register, without knowledge of $g(x)$. This is useful, e.g., in learning far-range behaviour/correlations. By the Stinespring dilation theorem, the quantum process has some unitary representation Q_c, which can be taken as our oracle. Note that our algorithm is insensitive to the addition of the $g(x)$ register (or extraneous phases).

1.4. Open problems

This work leaves several interesting avenues for further research. Firstly, one could attempt to tighten the upper bound (3) to remove polylogarithmic factors and prove a tight matching lower bound. The removal of a $\log(1/\epsilon)$ factor in the query complexity for classical PAC learning took 27 years Blumer et al. (1989); Hanneke (2016); we hope that the quantum case will be simpler. Moreover, in order to achieve $1/\sqrt{\epsilon}$ scaling with our method, one would require the optimal classical equivalence query learning complexity to have no ϵ dependence and thus, a different approach is likely to be required.

It is interesting to consider the power of quantum learning algorithms with access to the oracle Q_c, but not its inverse $Q_c^\dagger$. The inverse oracle seems necessary for Grover's search, and thus it is unclear if a quantum advantage is possible. The lack of such an advantage would have interesting implications for understanding what makes quantum computing more powerful than classical computation.

Finally, one could consider the implications of this work to generic advantages in more practical models of quantum machine learning, such as quantum neural networks.

1.5. Organisation

We first cover all required technical preliminaries in section 2. In section 3, we cover our Grover subroutine that leads to the quadratic advantage. Equivalence queries and how to use imperfect equivalence queries in a classical learning algorithm are both described in section 4. Using the results of these two sections, we derive the upper bound (3) in section 5; we prove an almost matching lower bound on our quantum model in section 6, using a reduction to a phase oracle problem. Finally, we consider the application of our algorithm to learning k-juntas in appendix D.

2. Preliminaries

We will only consider functions defined on finite sets. We first introduce the standard, classical model of PAC learning Valiant (1984). For a finite set $\mathcal{X}$, let $\{0,1\}^{\mathcal{X}} = \{f : \mathcal{X} \to \{0,1\}\}$, an element $f \in \{0,1\}^{\mathcal{X}}$ is called a classifier. We wish to *approximately* learn an unknown classifier c from a known subset of classifiers $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$, where $\mathcal{C}$ is called a concept class.

There is an unknown distribution $\mathcal{D}$ on $\mathcal{X}$, where $\mathcal{D}(x)$ denotes the probability of drawing x from $\mathcal{X}$. The distance between two classifiers is defined as the probability they disagree: $d(h_1, h_2) := \mathbb{P}_{X\sim\mathcal{D}}[h_1(X) \neq h_2(X)]$. For a fixed tolerance $\epsilon > 0$ we say a classifier $h \in \{0,1\}^{\mathcal{X}}$ is ϵ-*approximately correct* if $d(h, c) \leq \epsilon$.

A learning algorithm $\mathcal{A}$ has access to some oracle that gives information about c. Traditionally, one assumes that the oracle generates a labelled example $(X, c(X))$ at random, where X is distributed according to $\mathcal{D}$. We will consider an additional type of oracle in section 4. The sample complexity of $\mathcal{A}$ is the number of labelled examples it receives.

For a fixed error probability δ, we say that an algorithm $\mathcal{A}$ is an (ϵ, δ) learner if, with probability at least $1 - \delta$ (over the randomness of the algorithm), the algorithm outputs an ϵ-approximately correct hypothesis, *for every possible c and $\mathcal{D}$*.

For a fixed concept class $\mathcal{C}$ and $\epsilon, \delta > 0$, one wishes to find an (ϵ, δ)-learner with minimum sample complexity. The optimal sample complexity will depend on ϵ, δ and some measure of complexity of the class $\mathcal{C}$, which we now define. For a subset $Y \subseteq \mathcal{X}$, we define $\mathcal{C}|_Y := \{c|_Y : c \in \mathcal{C}\}$ as the restriction of the concept class to Y. We say that $\mathcal{C}$ shatters Y if $\mathcal{C}|_Y = \{0,1\}^Y$, i.e., if all possible labellings of Y appear in concepts in $\mathcal{C}$. The Valiant-Chapernikis (VC) dimension Vapnik and Chervonenkis (1971) of $\mathcal{C}$, denoted d, is the maximum size of a shattered set, that is $d = \max\{|Y| : Y \text{ is shattered by } \mathcal{C}\}$.

In Blumer et al. (1989); Hanneke (2016), it was shown that the optimal sample complexity using labelled examples, denoted $T_C(\epsilon, \delta, d)$ scales as

$$T_C = \Theta\left[\frac{1}{\epsilon}\left(d + \log\left(\frac{1}{\delta}\right)\right)\right]. \tag{5}$$

In the quantum PAC setting Bshouty and Jackson (1995), one assumes that the data is stored coherently, i.e., one considers the state

$$|\psi_c\rangle := \sum_{x \in \mathcal{X}} \sqrt{\mathcal{D}(x)}\, |x\ c(x)\rangle\,, \tag{6}$$

chosen so that measuring $|\psi_c\rangle$ in the computational basis gives a random labelled example. Instead of the classical sample complexity, one considers the minimum number of copies $T_S(\epsilon, \delta, d)$ of $|\psi_c\rangle$ required to PAC learn $\mathcal{C}$. Since one can always measure the state in place of a call to a classical oracle, T_S is, at worst, the optimal sample complexity of a classical algorithm. In fact, Arunachalam and de Wolf Arunachalam and de Wolf (2018) showed that there is no (asymptotic) quantum advantage from using states instead of oracle calls - the optimal T_S grows exactly as in equation (5).

We assume a stronger model, in which one has access to an oracle Q_c (which depends on the underlying concept), defined by its action on a fixed known input state $|\text{IN}\rangle$ (independent of the underlying concept):

$$Q_c\,|\text{IN}\rangle = |\psi_c\rangle = \sum_{x \in \mathcal{X}} \sqrt{\mathcal{D}(x)}\, |x\ c(x)\rangle\,. \tag{7}$$

This is similar in spirit to the recent work van Apeldoorn et al. (2023), which deals with state tomography with a state preparation unitary. We also assume that the algorithm has access to the inverse of the oracle, $Q_c^\dagger$. This is relevant if, for example, Q_c is given as a quantum circuit of one or two-qubit gates; in this case, $Q_c^\dagger$ may be constructed by reversing the order of the gates and replacing each with its inverse. We define the learning complexity of any algorithm as the total number of queries to Q_c or $Q_c^\dagger$. The minimum learning complexity of any (ϵ, δ)-learner is denoted $T_Q(\epsilon, \delta, \mathcal{C})$.

The lower bound of Arunachalam and de Wolf (2018) does not apply to a model with access to Q_c, as it assumes the quantum algorithm is described by a POVM that is *independent of the underlying concept* c. However, Q_c explicitly depends on c and thus, any algorithm (or POVM) that calls Q_c will violate the assumptions in Arunachalam and de Wolf (2018). Hence, one can hope for quantum advantage in this setting.

We recap all of the different learning models considered in Table 1.

Model	Quantum or Classical	Learning resource	Optimal (ϵ, δ) learner complexity	Bounds on optimal learner complexity
Labelled examples	Classical	Sample $(X, C(X))$ where $X \sim \mathcal{D}$	T_C	$\Theta\left[\frac{1}{\epsilon}\left(d + \log\left(\frac{1}{\delta}\right)\right)\right]$
Equivalence queries	Classical	See section 4	T_E	$O\left(\left[d + \log\left(\frac{1}{\delta}\right)\right] \log^9\left(\frac{1}{\epsilon}\right)\right)$
Imperfect equivalence queries	Classical	See section 4	T_{IE}	$O(T_E + \log\left(\frac{1}{\delta}\right))$
Quantum samples	Quantum	Copy of $\lvert\psi_c\rangle$	T_S	$\Theta(T_C)$
Quantum oracle calls	Quantum	Application of Q_c or $Q_c^\dagger$	T_O	$O(\frac{1}{\sqrt{\epsilon}} T_{IE})$, $\Omega(\frac{d}{\sqrt{\epsilon}})$

Table 1: Different learning models considered in our work. T_M corresponds to the minimum number of resources needed by any (ϵ, δ)-learner in model M. The models introduced by this work have grey rows.

We end the preliminaries section with a recap of Grover's algorithm. For a subspace $\mathcal{V}$ of a Hilbert space $\mathcal{H}$, let $\Pi_\mathcal{V}$ be the orthogonal projection map onto $\mathcal{V}$. Furthermore, let $I_\mathcal{V}$ be the reflection operator in $\mathcal{V}^\perp$, given by $I_\mathcal{V} = \mathbb{1} - 2\Pi_\mathcal{V}$. For a state $\lvert\psi\rangle$, let $I_{\lvert\psi\rangle}$ be the reflection operator when $\mathcal{V} = \text{span}\{\lvert\psi\rangle\}$.

Grover search takes as its input a "good" subspace $\mathcal{G} \subseteq \mathcal{H}$, and an input state $\lvert\psi\rangle$. One then implements the Grover operator $D = -I_{\lvert\psi\rangle} I_\mathcal{G}$. The state $\lvert\psi\rangle$ can be decomposed as $\lvert\psi\rangle = \sin(\theta)\lvert g\rangle + \cos(\theta)\lvert b\rangle$, where $\lvert g\rangle, \lvert b\rangle$ are orthonormal, $\theta \in [0, \pi/2]$, $\lvert g\rangle \in \mathcal{G}, \lvert b\rangle \in \mathcal{G}^\perp$. It is well-known Nielsen and Chuang (2010) that $D^n \lvert\psi\rangle = \sin((2n+1)\theta)\lvert g\rangle + \cos((2n+1)\theta)\lvert b\rangle$ and thus if one knows θ exactly, one can apply D^n such that $\sin((2n+1)\theta) \approx 1$.

3. Grover Subroutine

An essential subroutine for our quantum advantage is to use calls to Q_c and $Q_c^\dagger$ to run a Grover search Nielsen and Chuang (2010); Grover (1996). This leads to a quadratic improvement in learning complexity (up to polylogarithmic factors) over classical PAC learning. In this section, we describe our Grover subroutine.

Our Grover subroutine takes as an input a "good" subset $G \subseteq \{(x, b) : x \in \mathcal{X}, b \in \{0, 1\}\}$, where we wish to find an x such that $(x, c(x)) \in G$. We define a corresponding "good" subspace by $\mathcal{G} = \text{span}\{\lvert x\, b\rangle : (x, b) \in G\}$. In order to implement Grover's search,

we need to implement the Grover operator, as defined above. We show that implementing D requires a constant number of queries.

Lemma 3: One can implement the Grover operator D with one call to Q_c and one to $Q_c^\dagger$.

Proof Sketch: We use that $I_{|\psi_c\rangle} = Q_c(\mathbb{1} - 2\,|\text{IN}\rangle\langle\text{IN}|)Q_c^\dagger$, see appendix A for details. □

We decompose $|\psi_c\rangle = \sin(\theta)\,|g\rangle + \cos(\theta)\,|b\rangle$, where $|g\rangle, |b\rangle$ are orthonormal, $\theta \in [0, \pi/2]$, $|g\rangle \in \mathcal{G}, |b\rangle \in \mathcal{G}^\perp$. If we knew θ exactly, we could apply D^n such that $\sin((2n+1)\theta) \approx 1$. However, since θ depends on $\mathcal{D}$, which is unknown, this is impossible. Instead, we use the well-established Boyer et al. (1998) version of Grover's search for an unknown number of items. Our exact subroutine is given below; algorithm 1. The properties of algorithm 1 are summarised in the following theorem:

Theorem 4: Let $G \subseteq \{(x, b) : x \in \mathcal{X}, b \in \{0, 1\}\}$ be a good subset, $\epsilon > 0$ be a fixed tolerance. Suppose that we run algorithm 1 with these inputs, then

(i) In the worst case, the algorithm makes $O(1/\sqrt{\epsilon})$ oracle (or inverse oracle) calls

(ii) If $\mathbb{P}_{X\sim\mathcal{D}}\left[(X, c(X)) \in G\right] \geq \epsilon$ then the algorithm succeeds, i.e., returns $(x, c(x)) \in G$, with probability at least $p = 0.09$.

(iii) Conditional on succeeding, the output of the algorithm $(X, c(X))$ is distributed according to

$$\mathbb{P}\left[(X, c(X))|\text{ algorithm succeeds }\right] = \frac{\mathbb{P}_{X\sim\mathcal{D}}\left[X\right]}{\mathbb{P}_{X\sim\mathcal{D}}\left[(X, c(X)) \in G\right]}. \tag{8}$$

Proof Sketch: Part (i): This follows from the definition of the algorithm and Lemma 3.

Part (ii): Let $M = \lceil 2/\sqrt{\epsilon}\rceil$, let θ be as above and let $p_s(\theta)$ be the probability that the algorithm succeeds. We use Lemma 2 (section 6) from Boyer et al. (1998), which claims

$$p_s(\theta) = \frac{1}{2} - \frac{1}{4M}\frac{\sin(4M\theta)}{\sin(2\theta)}. \tag{9}$$

We relegate the technical details of boudning this function to appendix A.

Part (iii). This follows from the form of $D^n\,|\psi_c\rangle$. □

Algorithm 1:
Input: $G \subseteq \{(x, b) : x \in \mathcal{X}, b \in \{0, 1\}\}$ a good subspace, $\epsilon > 0$ a tolerance
Output: labelled example $(x, c(x))$. Succeeds if $(x, c(x)) \in G$

1. Produce $|\psi_c\rangle = Q_c\,|\text{IN}\rangle$

2. Pick N from $0, 1 \ldots, \lceil 2/\sqrt{\epsilon} \rceil - 1$ uniformly at random
3. Apply D, the Grover operator, N times to $|\psi_c\rangle$
4. Measure the resulting state in the computational basis

We discuss how to combine the Grover subroutine with the algorithm of section 4 to achieve a quantum learning complexity of equation (3) in section 5.

4. Learning with imperfect equivalence queries

Equivalence queries are an alternative learning model for PAC learning. It was recently shown Gluch and Urbanke (2021) that PAC learning with equivalence queries gives an exponential advantage over learning with labelled examples. In this section, we show how to use imperfect equivalence queries to PAC learn a concept class.

Definition 5: An (ideal) equivalence query consists of submitting a candidate hypothesis h for an underlying true concept c. If $h = c$ then we are told YES. Otherwise, we receive a labelled example $(x, c(x))$ where $c(x) \neq h(x)$ at random, according to the distribution $\mathbb{P}(y) = \mathcal{D}(y)/\mathcal{D}(\{x : c(x) \neq h(x)\})$. Such a labelled example where $h(x) \neq c(x)$ is called a counterexample.

Equivalence queries are a very strong learning model, which is perhaps unrealistic. Thus, we assume we can only implement them probabilistically:

Definition 6: An imperfect equivalence query consists of submitting a candidate hypothesis h for the underlying concept c. In return we receive some labelled example $(x, c(x))$ with the following promises

- The distribution of $(X, c(X))$ *conditional on being a counterexample* is the same as an ideal equivalence query.
- If $d(h, c) \geq \epsilon$ then with some constant probability p we receive a counterexample.

Note that we can tell whether our imperfect equivalence query failed or not - we can look at the result $(x, c(x))$ and check whether $h(x) = c(x)$. If they are equal, the equivalence query failed. Otherwise, it succeeded. Classically, we can implement an imperfect equivalence query using $1/\epsilon$ random labelled examples - we just sample $1/\epsilon$ times and see whether $c(x) \neq h(x)$ for any of our samples. On a quantum computer we can do this in $1/\sqrt{\epsilon}$ time using Grover's algorithm, as described in section 3 in Theorem 4.

We need one additional tool from classical learning theory to run our algorithm:

Definition 7: Suppose we have a set of classifiers $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and a distribution ρ on $\mathcal{H}$. Then the weighted majority vote Masegosa et al. (2020), $\mathrm{WMV}_{\mathcal{H},\rho} \in \{0,1\}^{\mathcal{X}}$ is defined such that it maximises

$$\mathbb{P}_{h\sim\rho}\left[\mathrm{WMV}_{\mathcal{H},\rho}(x) = h(x)\right], \tag{10}$$

for every x (ties can be broken arbitrarily).

Suppose we have a classical algorithm $\mathcal{A}$ that uses $T_E(\epsilon, \delta, d)$ (ideal) equivalence queries to PAC learn a concept class $\mathcal{C}$. We show how to use $O(T_E + \log(1/\delta))$ imperfect equivalence queries to PAC learn the same concept class.

The full detail of the algorithm is given below in algorithm 2. We give some rough intuition for why the algorithm works - the technical proof is given by 2 lemmas - see appendix B. If $\mathcal{A}$ terminates, then with high probability, it outputs an approximately correct hypothesis. If we pick R large enough, then with high probability T_E ideal queries to hypotheses h_i with $d(h_i, c) \geq \epsilon$ would all succeed in $< R/3$ imperfect equivalence queries. Thus, if the algorithm $\mathcal{A}$ does not terminate and we make R total imperfect equivalence queries, with high probability, we spent $> 2/3$ of our imperfect equivalence queries on hypotheses h_i with $d(h_i, c) < \epsilon$. Hence, if we take the weighted majority vote of all of the hypotheses we queried, weighted by the number of imperfect equivalence queries spent on each hypothesis, most of the vote will be decided by hypotheses that are close to the concept c. Thus, the weighted majority vote will also be close to c. We formally describe the performance of our algorithm in the following theorem:

Theorem 8: Let the maximum number of imperfect equivalence queries of algorithm 2 be

$$R(T_E(\epsilon, \delta, d), \delta) = 6T_E(\epsilon, \delta, d)/p + (3/2p^2)\log(1/\delta), \tag{11}$$

where p is the constant appearing in theorem 4. Then algorithm 2 produces a hypothesis h with $d(h, c) \leq 4\epsilon$ with probability at least $1 - 2\delta$.

Algorithm 2:
Input: $\delta > 0, \epsilon > 0$ (the usual PAC parameters) and $\mathcal{A}$, a classical equivalence query learning algorithm with worst case query complexity $T_E > 0$
Output: Hypothesis $h \in \{0,1\}^{\mathcal{X}}$

1. Set the maximum imperfect equivalence query budget as $R = 6T_E/p + (3/2p^2)\log(1/\delta)$. If R total imperfect equivalence queries have ever been made, go to step 3
2. Run $\mathcal{A}$, whenever it requires an equivalence query to a hypothesis h, repeatedly make imperfect equivalence queries until one succeeds. If $\mathcal{A}$ terminates, output the output of $\mathcal{A}$
3. Let $\mathcal{H} = \{h_1, \ldots, h_k\}$ be the set of hypothesis we ran imperfect equivalence queries on (so that $k \leq T_E$). Suppose we spent n_i imperfect equivalence queries on h_i (so that $\sum n_i = R$). Let $\rho(h_i) = n_i/N$ and output $h = \mathrm{WMV}_{\mathcal{H},\rho}$

5. Upper bound on quantum learning complexity

Here, we combine the results of sections 3 and 4 to give an upper bound on T_Q, the learning complexity of PAC learning with a state preparation oracle Q_c (and its inverse).

Suppose that it takes $E(\epsilon)$ queries to perform an imperfect equivalence query for a hypothesis h. If we have a classical equivalence learning algorithm $\mathcal{A}$ with a query complexity of $T_E(\epsilon, \delta, d)$, then we can use algorithm 2 of section 4 to get a quantum PAC learning algorithm with learning complexity $E(\epsilon/4)R(T_E(\epsilon/4, \delta/2, d), \delta/2)$. The current best known T_E Gluch and Urbanke (2021) has a worst-case query complexity of $T_E = O\left(\left[d + \log\left(\frac{1}{\delta}\right)\right] \log^9\left(\frac{1}{\epsilon}\right)\right)$.

If we use the Grover subroutine (section 3 algorithm 1) with $G = \{(x, 1-h(x)) : x \in \mathcal{X}\}$ to implement the imperfect equivalence queries, we find $E(\epsilon) = O(1/\sqrt{\epsilon})$. Substituting these T_E and E into the bound above, we get an upper bound of $T_O = O\left(\frac{1}{\sqrt{\epsilon}}\left[d + \log\left(\frac{1}{\delta}\right)\right] \log^9\left(\frac{1}{\epsilon}\right)\right)$, which is a square-root improvement (up to polylogarithmic factors) over the classical PAC learning sample complexity of equation (5).

6. Lower bound on quantum learning complexity

In this section, we prove a lower bound on quantum PAC learning with a state preparation oracle (and its inverse). We show that $\Omega(d/\sqrt{\epsilon})$ oracle calls are necessary.

Suppose we have a concept class $\mathcal{C}$ with VC dimension $d+1$. Then there is a set Z of size $d+1$ in $\mathcal{X}$ which is shattered by $\mathcal{C}$. We pick a marked element $x_0 \in Z$ and let $Y = Z \setminus \{x_0\}$. We define our distribution $\mathcal{D}$ as a perturbed delta-function, the standard distribution used to prove lower bounds in learning: $\mathcal{D}(x_0) = 1 - 4\epsilon$, $\mathcal{D}(y) = 4\epsilon/d$, for $y \in Y$, and $\mathcal{D}(x) = 0$, otherwise.

We also restrict our concept class to $\widetilde{\mathcal{C}} = \{c \in \mathcal{C} : c(x_0) = 0\}$. If our PAC algorithm works on $\mathcal{C}$, it will certainly work on $\widetilde{\mathcal{C}}$. Since our distribution is restricted to Z we need only identify the behaviour of our concept on Z. Thus, we can index our concepts by bit-strings $u \in \{0,1\}^d$ and index them with elements of Y. To be precise, we identify a concept $c \in \widetilde{\mathcal{C}}$ with a bit-string $u \in \{0,1\}^d$, where $u_y = c(y)$.

For a given bit-string $u \in \{0,1\}^d$, the state preparation oracle acts as

$$Q_u \left|\mathrm{IN}\right\rangle = \sqrt{1-4\epsilon}\left|x_0\ 0\right\rangle + \sqrt{\frac{4\epsilon}{d}} \sum_{x \in Y} \left|x\ u_x\right\rangle . \tag{12}$$

Our main approach is to reduce to the following fact from Lemma 51 in van Apeldoorn et al. (2023).

Lemma 9: Let $u \in \{0,1\}^d$ be a bit string, and let O_u be a weak phase-kickback oracle, that is

$$O_u \left|x\right\rangle = e^{2i\eta u_x} \left|x\right\rangle . \tag{13}$$

Then recovering more than 3/4 of the bits of u with high probability requires at least $\Omega(d/\eta)$ calls to O_u, its inverse or controlled versions of these.

Proof: See van Apeldoorn et al. (2023) □

We will use calls to controlled versions of O_u (denoted $c\text{-}O_u$) to implement the PAC state generation oracle Q_u. We fix $\eta \in [0, \pi/2]$ such that $\sin(\eta) = \sqrt{4\epsilon}$.

Lemma 10: One can implement Q_u using one call to $c\text{-}O_u$, one to $c\text{-}O_u^\dagger$ and two qubit-ancillae.

Proof Sketch: One can use several H gates to transform the phase oracle into an amplitude oracle. For a detailed proof, see appendix C. □

We thus deduce our bound

Theorem 11: $T_O = \Omega\left(\frac{d}{\sqrt{\epsilon}}\right)$

Proof Sketch: This follows by combining lemma 9 and lemma 10, see appendix C for details. □

Note that our lower bound matches our upper bound (equation (3)), up to polylogarithmic factors.

Acknowledgments

The authors thank J. van Apeldoorn, R. de Wolf, S. Arunachalam, J. Cudby, C. Long and J. Bayliss for helpful discussions related to this work.

Wilfred Salmon was supported by the EPRSC and Hitachi. Sergii Strelchuk acknowledges support from the Royal Society University Research Fellowship. Tom Gur is supported by the UKRI Future Leaders Fellowship MR/S031545/1 and an EPSRC New Horizons Grant EP/X018180/1. Sergii Strelchuk and Tom Gur are further supported by EPSRC Robust and Reliable Quantum Computing Grant EP/W032635/1.

References

Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2088):3089–3114, 2007.

Dana Angluin and Michael Kharitonov. When won't membership queries help? In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 444–454, 1991.

Richard P Anstee, Lajos Rónyai, and Attila Sali. Shattering news. *Graphs and Combinatorics*, 18:59–73, 2002. doi: 10.1007/s003730200003.

Srinivasan Arunachalam and Ronald de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, 6 2017. ISSN 0163-5700. doi: 10.1145/3106700.3106710. URL https://doi.org/10.1145/3106700.3106710.

Srinivasan Arunachalam and Ronald de Wolf. Optimal quantum sample complexity of learning algorithms. *Journal of Machine Learning Research*, 19(71):1–36, 2018. URL http://jmlr.org/papers/v19/18-195.html.

Srinivasan Arunachalam, Yihui Quek, and John Smolin. Private learning implies quantum stability. *arXiv preprint arXiv:2102.07171*, 2021.

Alp Atici and Rocco A. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6, 2007. doi: https://doi.org/10.1007/s11128-007-0061-6.

Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi: 10.1137/S0097539796300921.

Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 10 1989. ISSN 0004-5411. doi: 10.1145/76359.76371. URL https://doi.org/10.1145/76359.76371.

Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 46(4-5):493–505, 1998. doi: https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5⟨493::AID-PROP493⟩3.0.CO;2-P. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291521-3978%28199806%2946%3A4/5%3C493%3A%3AAID-PROP493%3E3.0.CO%3B2-P.

Nader H Bshouty and Jeffrey C Jackson. Learning dnf over the uniform distribution using a quantum example oracle. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 118–127, 1995. doi: 10.1137/S0097539795293123.

Haoyuan Cai, Qi Ye, and Dong-Ling Deng. Sample complexity of learning parametric quantum circuits. *Quantum Science and Technology*, 7(2):025014, 2022.

Hao-Chung Cheng, Min-Hsiu Hsieh, and Ping-Cheng Yeh. The learnability of unknown quantum measurements. *arXiv preprint arXiv:1501.00559*, 2015.

Kai-Min Chung and Han-Hsuan Lin. Sample efficient algorithms for learning quantum channels in pac model and the approximate state discrimination problem. *arXiv preprint arXiv:1810.10938*, 2018.

Grzegorz Gluch and Ruediger Urbanke. Exponential separation between two learning models and adversarial robustness. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 20785–20797. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/ae06fbdc519bddaa88aa1b24bace4500-Paper.pdf.

Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 212–219, New York, NY, USA, 1996. Association for Computing Machinery. ISBN 0897917855. doi: 10.1145/237814.237866. URL https://doi.org/10.1145/237814.237866.

Jeongwan Haah, Robin Kothari, Ryan O'Donnell, and Ewin Tang. Query-optimal estimation of unitary channels in diamond distance. *arXiv preprint arXiv:2302.14066*, 2023.

Steve Hanneke. The optimal sample complexity of pac learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016. URL http://jmlr.org/papers/v17/15-389.html.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1963.10500830.

Robin Kothari and Ryan O'Donnell. *Mean estimation when you have the source code; or, quantum Monte Carlo methods*, pages 1186–1215. Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2023. doi: 10.1137/1.9781611977554.ch44. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977554.ch44.

Andres Masegosa, Stephan Lorenzen, Christian Igel, and Yevgeny Seldin. Second order pac-bayesian bounds for the weighted majority vote. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 5263–5273. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/386854131f58a556343e056f03626e00-Paper.pdf.

Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511976667.

Arun Padakandla and Abram Magner. Pac learning of quantum measurement classes: Sample complexity bounds and universal consistency. In *International Conference on Artificial Intelligence and Statistics*, pages 11305–11319. PMLR, 2022.

Niklas Pirnay, Ryan Sweke, Jens Eisert, and Jean-Pierre Seifert. Superpolynomial quantum-classical separation for density modeling. *Phys. Rev. A*, 107:042416, 4 2023. doi: 10.1103/PhysRevA.107.042416. URL https://link.aps.org/doi/10.1103/PhysRevA.107.042416.

Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

Andrea Rocchetto. Stabiliser states are efficiently pac-learnable. *Quantum Info. Comput.*, 18(7–8):541–552, 6 2018. ISSN 1533-7146.

L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 11 1984. ISSN 0001-0782. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

Joran van Apeldoorn, Arjan Cornelissen, András Gilyén, and Giacomo Nannicini. *Quantum tomography using state-preparation unitaries*, pages 1265–1318. Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 2023. doi: 10.1137/1.9781611977554.ch47. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977554.ch47.

V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971. doi: 10.1137/1116025. URL https://doi.org/10.1137/1116025.

Chi Zhang. An improved lower bound on query complexity for quantum pac learning. *Information Processing Letters*, 111(1):40–45, 2010.

Appendix A. Proofs for Grover's subroutine

We give full proofs for Lemma 3 and Theorem 4 below:

Lemma 12: One can implement the Grover operator D with one call to Q_c and one to $Q_c^\dagger$.

Proof: Note that I_G is independent of c and, therefore, may be implemented by a (possibly exponentially sized circuit) without any queries. To implement $I_{|\psi_c\rangle}$, note that

$$I_{|\psi_c\rangle} = \mathbb{1} - 2\,|\psi_c\rangle\langle\psi_c|\,, \tag{14}$$
$$= Q_c(\mathbb{1} - 2\,|\mathrm{IN}\rangle\langle\mathrm{IN}|)Q_c^\dagger, \tag{15}$$
$$= Q_c I_{|\mathrm{IN}\rangle} Q_c^\dagger. \tag{16}$$

Note that $I_{|\mathrm{IN}\rangle}$ is independent of c and, therefore, may be implemented by a (possibly exponentially sized circuit) without any queries. □

Theorem 13: Let $G \subseteq \{(x,b) : x \in \mathcal{X}, b \in \{0,1\}\}$ be a good subset, $\epsilon > 0$ be a fixed tolerance. Suppose that we run algorithm 1 with these inputs, then

(i) In the worst case, the algorithm makes $O(1/\sqrt{\epsilon})$ oracle (or inverse oracle) calls

(ii) If $\mathbb{P}_{X\sim\mathcal{D}}\left[(X, c(X)) \in G\right] \geq \epsilon$ then the algorithm succeeds, i.e., returns $(x, c(x)) \in G$, with probability at least $p = 0.09$.

(iii) Conditional on succeeding, the output of the algorithm $(X, c(X))$ is distributed according to

$$\mathbb{P}\left[(X, c(X))|\text{ algorithm succeeds }\right] = \frac{\mathbb{P}_{X\sim\mathcal{D}}\left[X\right]}{\mathbb{P}_{X\sim\mathcal{D}}\left[(X, c(X)) \in G\right]}. \tag{8}$$

Proof:
Part (i): From the definition of the algorithm and Lemma 3, the worst case number of oracle calls is $1 + 2(\lceil 2/\sqrt{\epsilon}\rceil - 1) = O(1/\sqrt{\epsilon})$.

Part (ii): Let $M = \lceil 2/\sqrt{\epsilon}\rceil$, let θ be as above and let $p_s(\theta)$ be the probability that the algorithm succeeds. Note that $\mathbb{P}_{X\sim\mathcal{D}}\left[(X, c(X)) \in G\right] \geq \epsilon \Leftrightarrow \sin(\theta) \geq \sqrt{\epsilon}$. We use Lemma 2 (section 6) from Boyer et al. (1998), which claims

$$p_s(\theta) = \frac{1}{2} - \frac{1}{4M}\frac{\sin(4M\theta)}{\sin(2\theta)}. \tag{17}$$

For $\sin(\theta) \in [\sqrt{\epsilon}, 1/\sqrt{2}]$:

$$M \geq \frac{2}{\sin(\theta)}, \tag{18}$$

$$\geq \frac{1}{\sin(2\theta)}, \tag{19}$$

and thus

$$p_s(\theta) \geq \frac{1}{2} - \frac{1}{4} = \frac{1}{4} > 0.09. \tag{20}$$

Note that for $\theta \in [\pi/4, \pi/2]$,

$$\sin(2\theta) \geq \frac{\pi/2 - \theta}{\pi/4}, \tag{21}$$

Thus for $\theta \in [\pi/4, (1/2 - 1/4M)\pi]$, we have that

$$p_s(\theta) \geq \frac{1}{2} - \frac{1}{4M} \cdot \frac{4/\pi}{\pi/2 - (1/2 - 1/4M)\pi}, \tag{22}$$

$$= \frac{1}{2} - \frac{4}{\pi^2} > 0.09. \tag{23}$$

Finally, for $\theta \in [(1/2 - 1/4M)\pi, \pi/2]$, note that $\sin(2\theta) \geq 0$ and $\sin(4M\theta) \leq 0$ so that $p_s(\theta) \geq 1/2 > 0.09$.

Part (*iii*). This follows from the form of $D^n |\psi_c\rangle$; the relative magnitude of the amplitudes in $|g\rangle$ is unchanged by the Grover operator D. □

Appendix B. Proofs for learning with imperfect equivalence queries

We give full proofs of the performance of algorithm 2 below. Before proving two technical lemmas, we introduce some terminology:

Definition 14: A transcript of a run of algorithm 2 is given by the list of hypotheses $\mathcal{H} = \{h_i\}$ that the algorithm queried along with a corresponding collection of natural numbers $n_i > 0$, where n_i is the number of imperfect equivalence queries spent on h_i.

The time-spent distribution ρ is the probability distribution on $\mathcal{H}$ given by $\rho(h_i) = n_i / \sum_i n_i$.

Finally, $F = \{i : d(h_i, c) \geq \epsilon\}$ is called the "feasible" set, where our imperfect equivalence query succeeds with probability at least p. Correspondingly $I = \{i : d(h_i, c) < \epsilon\}$ is the "infeasible" set, where there is no promise on the probability of success.

Firstly, we show that with high probability that a bounded number of queries is spent on the feasible set

Lemma 15: With probability $\geq 1-\delta$ the total number of imperfect equivalence queries to feasible hypotheses is at most

$$2T_E/p + (1/2p^2)\log(1/\delta). \tag{24}$$

Proof: A imperfect equivalence query of a feasible hypothesis has (by definition) a chance $\geq p$ of succeeding, and the individual imperfect equivalence queries are independent. Additionally, there are at most T_E feasible hypotheses to query (since the classical algorithm makes at most T_E total equivalence queries). Thus, the probability that we succeed on all the feasible hypotheses using at most m imperfect queries feasible hypotheses is lower bounded by the probability of getting at least T_E successes from a binomial distribution $B(m,p)$. Thus, the chance of failure is lower bounded by the chance of fewer than T_E successes from $B(m,p)$.

Let $X \sim B(m,p)$. Applying Hoeffding's inequality Hoeffding (1963), for $m \geq T_E/p$ we see that

$$\mathbb{P}\left[X < t\right] \leq e^{-2m(p-T_E/m)^2}. \tag{25}$$

Thus it is sufficient for

$$2m\left(p - \frac{T_E}{m}\right)^2 \geq \log(1/\delta). \tag{26}$$

In turn, it is sufficient that

$$2mp^2 - 4pT_E \geq \log(1/\delta), \tag{27}$$

whence we deduce our bound.
□

Next we prove that if we make enough imperfect equivalence queries on infeasible hypotheses, the weighted majority vote of the transcript must be close to the underlying concept c

Lemma 16: Suppose we spend at least $2R/3$ imperfect equivalence queries on infeasible hypotheses. Then the weighted majority vote M of the transcript with the time-spent distribution has $d(M,c) < 4\epsilon$.

Proof: Fix the transcript $h_1, \ldots h_k$. Let ρ be the time-spent distribution and let ρ' be the time-spent distribution conditioned on the infeasible set. That is, for $i \in I$, $\rho'(h_i) = \rho(h_i)/\rho(I)$. Similarly let $\tilde{\rho}$ the the time-spent distribution conditioned on the feasible set. We first show that if the infeasible set overwhelmingly votes for a bit y, then the whole transcript must also vote for that y. To be precise, suppose that

$\mathbb{P}_{h\sim\rho'}[h(x)=y] > 3/4$, then

$$\mathbb{P}_{h\sim\rho}[h(x)=y] = \mathbb{P}_{h\sim\rho'}[h(x)=y]\,\mathbb{P}_{h\sim\rho}[h\in I] + \mathbb{P}_{h\sim\tilde{\rho}}[h(x)=y]\,\mathbb{P}_{h\sim\rho}[h\in F], \quad (28)$$

$$> \frac{3}{4}\cdot\frac{2}{3}, \quad (29)$$

$$= \frac{1}{2}. \quad (30)$$

Let $M = \mathrm{WMV}_{\mathcal{H},\rho}$. By the above, if $\mathbb{P}_{h\sim\rho'}[h(x)=c(x)] > \frac{3}{4}$, then $M(x)=c(x)$. We deduce (inspired by Masegosa et al. (2020)) that

$$\mathbb{P}_{X\sim\mathcal{D}}[M(X)\neq c(X)] \leq \mathbb{P}_{X\sim\mathcal{D}}\left[\mathbb{P}_{h\sim\rho'}[h(X)\neq c(X)] \geq \frac{1}{4}\right], \quad (31)$$

$$\text{Markov's inequality, } \leq 4\mathbb{E}_{X\sim\mathcal{D}}\mathbb{E}_{h\sim\rho'}[\mathbb{1}_{\{h(X)\neq c(X)\}}], \quad (32)$$

$$= 4\mathbb{E}_{h\sim\rho'}[d(h,c)], \quad (33)$$

$$\text{definition of infeasible set, } < 4\epsilon \quad (34)$$

□

We can now prove the performance of our algorithm

Theorem 17: Let the maximum number of imperfect equivalence queries of algorithm 2 be

$$R(T_E(\epsilon,\delta,d),\delta) = 6T_E(\epsilon,\delta,d)/p + (3/2p^2)\log(1/\delta), \quad (11)$$

where p is the constant appearing in theorem 4. Then algorithm 2 produces a hypothesis h with $d(h,c)\leq 4\epsilon$ with probability at least $1-2\delta$.

Proof: By Lemma 15, with probability $\geq 1-\delta$ we spend at most $R/3$ imperfect equivalence queries on feasible hypotheses - suppose this happens. If we succeed in an equivalence query for every hypothesis required by $\mathcal{A}$ then with probability at least $1-\delta$, $\mathcal{A}$ outputs a hypothesis h with $d(h,c)\leq\epsilon$. Otherwise, we spend at least $2R/3$ imperfect equivalence queries on infeasible hypotheses (as we assumed the feasible ones took at most $R/3$ imperfect equivalence queries) and then by Lemma 16 the weighted majority vote $\mathrm{WMV}_{\mathcal{H},\rho}$ has $d(\mathrm{WMV}_{\mathcal{H},\rho},c) < 4\epsilon$. Thus algorithm 2 outputs a 4ϵ-approximately correct hypothesis with probability at least $(1-\delta)^2 \geq 1-2\delta$. □

Appendix C. Proofs for lower bound on quantum learning complexity

We provide full proofs of lemma 10 and Theorem 11 below:

Lemma 18: One can implement Q_u using one call to $c\text{-}O_u$, one to $c\text{-}O_u^\dagger$ and two qubit-ancillae.

Proof: First, it is convenient to shift the phase to have a $\pm$ symmetry. Define a constant phase gate as

$$P_\alpha |x\rangle = e^{i\alpha} |x\rangle . \tag{35}$$

Then let

$$\widetilde{O}_u = P_\eta O_u^\dagger, \tag{36}$$

so that

$$\widetilde{O}_u |x\rangle = e^{i\eta \hat{u}_x} |x\rangle , \tag{37}$$

where

$$\hat{u}_x = (-1)^{u_x}. \tag{38}$$

We start by generating a uniform superposition of indices with the two-qubit ancillae in the $|+\rangle$ state:

$$\frac{1}{2\sqrt{d}} \sum_{x\in Y} |x\rangle \left[|00\rangle + |01\rangle + |10\rangle + |11\rangle\right]. \tag{39}$$

We next apply 4 controlled gates - either c-P_η, c-$P_{-\eta}$ c-$\widetilde{O}_u$ and c-$\widetilde{O}_u^\dagger$, such that each term in the superposition in equation (39) picks up a different phase:

$$\mapsto \frac{1}{2\sqrt{d}} \sum_{x\in Y} |x\rangle \left[e^{i\eta} |00\rangle + e^{-i\eta} |01\rangle + e^{i\eta\hat{u}_x} |10\rangle + e^{-i\eta\hat{u}_x} |11\rangle\right]. \tag{40}$$

Note that this requires two calls to singly controlled versions of the oracle - we can implement a double-controlled version by using a CCNOT (Toffoli) gate followed by a controlled oracle. Next, we apply a Hadamard gate to the second qubit register

$$\mapsto \frac{1}{\sqrt{2d}} \sum_{x\in Y} |x\rangle \left[|0\rangle \left(\cos(\eta) |0\rangle + i \sin(\eta) |1\rangle\right) + |1\rangle \left(\cos(\eta\hat{u}_x) |0\rangle + i \sin(\eta\hat{u}_x) |1\rangle\right)\right]. \tag{41}$$

We then apply $S^\dagger$ to the second qubit register (to remove the factors of i). We also use the even/odd ness of cos/sin to regroup the terms:

$$\mapsto \frac{1}{\sqrt{2d}} \sum_{x\in Y} |x\rangle \left[\cos(\eta)(|0\rangle + |1\rangle) |0\rangle + \sin(\eta)(|0\rangle + \hat{u}_x |1\rangle) |1\rangle\right]. \tag{42}$$

We then apply a Hadamard gate to the first qubit register:

$$\mapsto \cos(\eta) \left(\frac{1}{\sqrt{d}} \sum_{x\in Y} |x\rangle\right) |00\rangle + \sin(\eta) \left(\frac{1}{\sqrt{d}} \sum_{x\in Y} |x\ u_x\rangle\right) |1\rangle \tag{43}$$

Conditional on the final qubit being in the state $|0\rangle$, we apply a unitary to the first register that maps the uniform superposition over Y into the state $|x_0\rangle$:

$$\mapsto \cos(\eta)\,|x_0\ 0\ 0\rangle + \sin(\eta)\left(\frac{1}{\sqrt{d}}\sum_{x\in Y}|x\ u_x\rangle\right)|1\rangle \tag{44}$$

Finally, conditional on the first register not being in the state $|x_0\rangle$, we apply an X gate to the second qubit register, followed by an H gate on the second qubit register:

$$\mapsto \left[\cos(\eta)\,|x_0\ 0\rangle + \sin(\eta)\left(\frac{1}{\sqrt{d}}\sum_{x\in Y}|x\ u_x\rangle\right)\right]|+\rangle \tag{45}$$

But by the definition of η, we see that this is exactly equal to the action of the PAC oracle:

$$(Q_u\,|\mathrm{IN}\rangle)\,|+\rangle \tag{46}$$

□

Theorem 19: $T_O = \Omega\left(\frac{d}{\sqrt{\epsilon}}\right)$

Proof: We can replace every call to Q_u (or its inverse) in our PAC algorithm with the unitary process described in Lemma 10, which requires a constant number of calls to (a controlled) O_u (or its inverse). If the PAC algorithm outputs a correct hypothesis, then by construction of our distribution, it must agree on at least $3/4$ of the bits of u. Thus, the algorithm replaced with calls to O_u (and its inverse) satisfies the conditions of Lemma 9, and thus it must use at least $\Omega(d/\eta)$ calls to O_u. Hence, we reach a lower bound of

$$T_O = \Omega\left(\frac{d}{\arcsin\sqrt{4\epsilon}}\right) = \Omega\left(\frac{d}{\sqrt{\epsilon}}\right). \tag{47}$$

□

Appendix D. Application to learning k-juntas

A k-junta is a function $f : \{0,1\}^n \to 0,1$ that only depends on a subset of k bits. Letting $\mathcal{X} = \{0,1\}^n$, we can consider the concept class $\mathcal{C} = \{f \in \{0,1\}^{\mathcal{X}} : f \text{ is a } k \text{ junta}\}$. The exact VC dimension of $\mathcal{C}$ is unknown, but we can bound it using the inequalities

$$2^d \leq |\mathcal{C}| \leq |\mathcal{X}|^d + 1. \tag{48}$$

The first of these comes from noting that if $\mathcal{C}$ shatters a set of size ℓ, it must contain at least 2^ℓ elements; the second is called Sauer's lemma Anstee et al. (2002). We can bound

$$|\mathcal{C}| \leq \binom{n}{k} 2^{(2^k)}, \tag{49}$$

since there are $\binom{n}{k}$ ways to choose the k bits determining the junta, and then $2^{(2^k)}$ choices for the underlying function. We deduce that

$$d \leq \log\left[\binom{n}{k}\right] + 2^k \leq k\log(en/k) + 2^k. \tag{50}$$

Thus, our learning algorithm can PAC learn a k-junta with

$$O\left(\frac{1}{\sqrt{\epsilon}}\left[k\log\left(\frac{n}{k}\right) + 2^k + \log\left(\frac{1}{\delta}\right)\right]\log^9(1/\epsilon)\right), \tag{51}$$

oracle calls. This has a worse scaling in n than algorithms presented in Atici and Servedio (2007), but has a better scaling in ϵ and works for *any* underlying distribution, whereas previous work has focused on the uniform distribution.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il