Separating quantum and classical computing: rigorous proof and practical application

Marshall, S.C.

CHAPTER 6

Shadows of quantum machine learning

6.1 Introduction

Quantum machine learning is a rapidly growing field [93–95] driven by its potential to achieve quantum advantages in practical applications. A particularly interesting approach to make quantum machine learning applicable in the near term is to develop learning models based on parametrized quantum circuits [96–98]. Indeed, such quantum models have already been shown to achieve good learning performance in benchmarking tasks, both in numerical simulations [99–103] and on actual quantum hardware [104–107]. Moreover, based on widely-believed cryptography assumptions, these models also hold the promise to solve certain learning tasks that are intractable for classical algorithms [108, 109], including predicting ground state properties of highly-interacting quantum systems [110].

Despite these advances, quantum machine learning is facing a major obstacle for its use in practice. A typical workflow of a machine learning model involved, e.g., in driving autonomous vehicles, is divided into: (i) a *training phase*, where the model is trained, typically using training data or by reinforcement; followed by (ii) a *deployment phase*, where the trained model is evaluated on new input data. For quantum machine learning models, both of these phases require access to a quantum computer. But

given that in many practical machine learning applications, the trained model is meant for a widespread deployment, the current scarcity of quantum computing access dramatically reduces the applicability of quantum machine learning. One way of addressing this problem is by generating *shadow models* out of quantum machine learning models. That is, we propose inserting a *shadowing phase* between the training and deployment, where a quantum computer is used to collect information on the quantum model. Then a classical computer can use this information to evaluate the model on new data during the deployment phase.

The conceptual idea of generating shadows of quantum models was already proposed by Schreiber *et al.* [111], albeit under the terminology of *classical surrogates*. In that work, as well as in that of Landman *et al.* [112], the authors make use of the general expression of quantum models as trigonometric polynomials [113] to learn the Fourier representation of trained models and evaluate them classically on new data. However, these works also suggest that a classical model could potentially be trained directly on the training data and achieve the same performance as the shadow model, thus circumventing the need for a quantum model in the first place. This raises the concern that all quantum models that are compatible with a classical deployment would also lose all quantum advantage, hence severely limiting the prospects for a widespread use of quantum machine learning.
Therefore, two natural open questions are raised:

1. *Can shadow models achieve a quantum advantage over entirely classical (classically trained and classically evaluated) models?*

2. *Do there exist quantum models that do not admit efficiently evaluatable shadow models?*

In this chapter, we resolve both of these key open questions. We propose a general definition for shadow models, rooted in the fundamental idea that quantum machine learning models can be universally expressed as linear models [114]. This formulation of shadow models allows us to leverage various results and techniques from quantum information theory for the analysis of this model class. From a practical perspective, employing shadow tomography techniques [115–118] allows to easily construct diverse shadow models that will resonate with the practitioners of quantum machine learning. Furthermore, in our exploration of the computational capabilities of shadow models, we find them to capture a distinct computational class. Specifically, we demonstrate that, under widely-believed cryptography assumptions, there exist learning tasks where

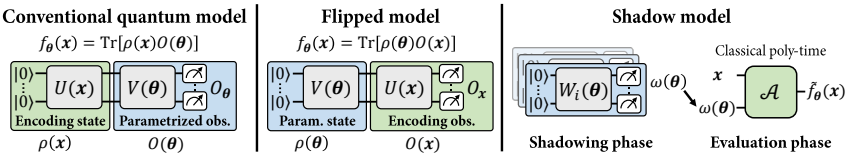

Figure 6.1: Quantum and shadow models. (left) Conventional quantum models can be expressed as inner products between a data-encoding quantum state $\rho(\boldsymbol{x})$ and a parametrized observable $O(\boldsymbol{\theta})$. The resulting linear model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$ naturally corresponds to a quantum computation, depicted here. (middle) We define flipped models $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ as quantum linear models where the role of the quantum state $\rho(\boldsymbol{\theta})$ and the observable $O(\boldsymbol{x})$ is flipped compared to conventional models. (right) Flipped models are associated to natural shadow models: one can use techniques from shadow tomography to construct a classical representation $\hat{\rho}(\boldsymbol{\theta})$ of the parametrized state $\rho(\boldsymbol{\theta})$ (during the shadowing phase), such that, for encoding observables $O(\boldsymbol{x})$ that are classically representable (e.g., linear combinations of Pauli observables), $\hat{\rho}(\boldsymbol{\theta})$ can be used by a classical algorithm to evaluate the model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ on new input data (during the evaluation phase). More generally, a shadow model is defined by (i) a shadowing phase where a (bit-string) advice $\omega(\boldsymbol{\theta})$ is generated by the evaluation of multiple quantum circuits $W_1(\boldsymbol{\theta}), \ldots, W_M(\boldsymbol{\theta})$, and (ii) an evaluation phase where this advice is used by a classical algorithm $\mathcal{A}$, along with new input data $\boldsymbol{x}$ to evaluate their labels $\widetilde{f_{\boldsymbol{\theta}}}(\boldsymbol{x})$. In section 6.3, we show that under this general definition, all shadow models are shadows of flipped models.

shadow models exhibit a provable quantum advantage over fully classical models. However, contrary to this advantage, we also establish that there exist quantum models that are strictly more powerful than the class of shadow models, based on common assumptions in complexity theory.

For ease of exposition, we will first adhere to a working definition of a shadow model as a model that is trained on a quantum computer, but can be evaluated classically on new input data with the help of information generated by a quantum computer (i.e., quantum-generated advice) that is independent of the new data. We will (informally) call a model "shadowfiable" if there exists a method of turning it into a shadow model. In Section 6.3, we will make our definitions more precise.

6.2 The flipped model

The construction of our shadow models starts from a simple yet key observation: all standard quantum machine learning models for supervised learning can be expressed as linear models [114]. To delve into this claim, we first draw upon early works that utilized parametrized quantum circuits in machine learning [99, 104]. These works proposed quantum models that are naturally expressed as linear functions of the form

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})] \tag{6.1}$$

where $\rho(\boldsymbol{x})$ are quantum states that encode classical data $\boldsymbol{x} \in \mathcal{X}$ and $O(\boldsymbol{\theta})$ are parametrized observables whose inner product with $\rho(\boldsymbol{x})$ defines $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ (see Fig. 6.1). In a regression task, one would use such a model to assign a real-valued label to an input $\boldsymbol{x}$, while in classification tasks, one would additionally apply, e.g., a sign function, to discretize its output into a class. From a circuit picture, such models can be evaluated on a quantum computer by: (i) preparing an initial state ρ_0, e.g., $|0\rangle\langle 0|^{\otimes n}$, (ii) evolving it under a data-dependent circuit $U(\boldsymbol{x})$, (iii) followed by a variational circuit $V(\boldsymbol{\theta})$, (iv) before finally measuring the expectation value of a Hermitian observable O. Together, steps (i) and (ii) define

$$\rho(\boldsymbol{x}) = U(\boldsymbol{x})\rho_0 U^\dagger(\boldsymbol{x}), \tag{6.2}$$

while steps (iii) and (iv) define

$$O(\boldsymbol{\theta}) = V^\dagger(\boldsymbol{\theta})OV(\boldsymbol{\theta}). \tag{6.3}$$

Since the early works, it is known that quantum linear models also capture quantum kernel models as a special case [119], simply by making $O(\boldsymbol{\theta})$ directly dependent on the training data of the learning task. Perhaps more surprisingly, quantum linear models can also encompass more general data re-uploading models, composed of several layers of data encoding and variational processing $U_1(\boldsymbol{x})V_1(\boldsymbol{\theta})U_2(\boldsymbol{x})\ldots$ Indeed, data re-uploading models can be mapped to linear models through circuit transformations (e.g., gate teleportation) that relocate all data-encoding gates to the first layer of the circuit [114].

6.2.1 Flipped model definition

The definition of a quantum linear model in Equation 6.1 can in general accommodate any pair of Hermitian operators in place of $\rho(\boldsymbol{x}), O(\boldsymbol{\theta})$. However, due to how these models are evaluated on a quantum computer, one commonly works under the constraint that $\rho(\boldsymbol{x})$ defines a quantum state (i.e., a positive semi-definite operator with unit trace). Indeed, from an operational perspective, $\rho(\boldsymbol{x})$ must be physically prepared on a quantum device before being measured with respect to the observable $O(\boldsymbol{\theta})$ (which only needs to be Hermitian in order to be a valid observable).

For reasons that will become clearer from the shadowing perspective, we define a so-called *flipped model*, where we flip the role of $\rho(\boldsymbol{x})$ and $O(\boldsymbol{\theta})$. That is, we consider

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})] \tag{6.4}$$

where $\rho(\boldsymbol{\theta})$ is a parametrized quantum state and $O(\boldsymbol{x})$ is an observable that encodes the data and can take more general forms than Equation 6.3 as we will see next. This model also corresponds to a straightforward quantum computation as $\rho(\boldsymbol{\theta})$ can be physically prepared before being measured with respect to $O(\boldsymbol{x})$.

A simple example of flipped model is for instance defined by:

$$\rho(\boldsymbol{\theta}) = V(\boldsymbol{\theta})\rho_0 V^\dagger(\boldsymbol{\theta}) \quad \& \quad O(\boldsymbol{x}) = \sum_{j=1}^{m} w_j(\boldsymbol{x})P_j \tag{6.5}$$

for an initial state ρ_0, a variational circuit $V(\boldsymbol{\theta})$, and a collection of Pauli observables $\{P_j\}_{j=1}^m$ weighted by data-dependent weights $w_j(\boldsymbol{x}) \in \mathbb{R}$. One can evaluate this model by repeatedly preparing $\rho(\boldsymbol{\theta})$ on a quantum computer, measuring it in a Pauli basis specified by a P_j, and weighting the outcome by $w_j(\boldsymbol{x})$. For other examples of flipped models, see Appendix 6.A.2.

As opposed to conventional quantum linear models, flipped models are well-suited to construct shadow models. Since the variational operators $\rho(\boldsymbol{\theta})$ are quantum states, one can straightforwardly use techniques from shadow tomography [115] to construct classical shadows $\hat{\rho}(\boldsymbol{\theta})$ of these states. What we call classical shadows $\hat{\rho}(\boldsymbol{\theta})$ here are collections of measurement outcomes obtained from copies of $\rho(\boldsymbol{\theta})$ that can be used to classically approximate expectation values of certain observables O (for a certain restricted family). If we take these observables to be our data-dependent $O(\boldsymbol{x})$, then we end up with a classical model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ that approximates our flipped model. Note here that one has total freedom on the classical shadow techniques they may use to define their shadow models, and a plethora of protocols have already been proposed in the literature [115–118]. But it is important to keep in mind that each of these protocols comes with its limitations, as it may restrict the class of states $\rho(\boldsymbol{\theta})$ or the class of observables $O(\boldsymbol{x})$ for which an *efficient and faithful* shadow model can be constructed. By *efficient* we refer here to the number of measurements performed on $\rho(\boldsymbol{\theta})$ and the time complexity of estimating the expectation values of observables $O(\boldsymbol{x})$ from these measurements. And by *faithful* we refer to the approximation error between the shadow model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ resulting from the shadow protocol and the original flipped model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. For instance, in the example of Equation 6.5, we know that if all Pauli operators $\{P_j\}_{j=1}^m$ are k-local, then $\widetilde{\mathcal{O}}\left(3^k B^2 \varepsilon^{-2}\right)$ measurements of $\rho(\boldsymbol{\theta})$, where $B = \max_{\boldsymbol{x}} \sum_{j=1}^m |w_i(\boldsymbol{x})|$, are sufficient to guarantee $\max_{\boldsymbol{x}} \left|\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right| \leq \varepsilon$ with high probability. But for non-local Pauli operators (i.e., large k), this protocol becomes highly inefficient if we want to guarantee a low error ε.

Importantly, shadowfied flipped models are not limited to constructions based on classical shadow protocols. Given that the states $\rho(\boldsymbol{\theta})$ are not given to us a black-box (as is generally assumed in shadow tomography), one can use prior knowledge on these states to construct efficient shadowing procedure. For instance, if $\rho(\boldsymbol{\theta})$ is known to be a superposition of a tractable number of computational basis states, or well-approximated by a matrix product state (MPS) with low bond dimension, then efficient tomography protocols may be used [120].

6.2.2 Properties of flipped models

Flipped models are a stepping stone toward the claims of quantum advantage and "shadowfiability" that are the focus of this chapter. Nonetheless, they constitute a newly introduced model, which is why it is useful to understand first how they relate to previous quantum models and what

learning guarantees they can have.

Since conventional linear models of the form of Equation 6.1 play a central role in quantum machine learning, we start by asking the question: when can these models be represented by (efficiently evaluatable) flipped models? That is, given a conventional model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$, can we construct a flipped model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})]$ such that $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \approx f_{\boldsymbol{\theta}}(\boldsymbol{x}), \forall \boldsymbol{x}, \boldsymbol{\theta}$, and $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ is as efficient to evaluate as $f_{\boldsymbol{\theta}}(\boldsymbol{x})$. Clearly, a conventional model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for which the parametrized operator $O(\boldsymbol{\theta})$ is also a quantum state (i.e., a positive semi-definite trace-1 operator) is by definition also a flipped model. Therefore, a natural strategy to flip a conventional model is to transform its observable $O(\boldsymbol{\theta})$ into a quantum state $\rho'(\boldsymbol{\theta})$. This transformation involves dealing with the negative eigenvalues of $O(\boldsymbol{\theta})$, which is straightforward[1], as well as *normalizing* these eigenvalues, which more importantly affects the efficiency of evaluating the resulting flipped model. Indeed, the normalization factor α that results from normalizing $O(\boldsymbol{\theta})$ corresponds to its trace norm $\|O\|_1 = \mathrm{Tr}\left[\sqrt{O^2}\right]$ and needs to be absorbed into the observable $O'(\boldsymbol{x}) = \alpha\rho(\boldsymbol{x})$ of the flipped model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ to guarantee $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$. This directly impacts the spectral norm $\|O'\|_\infty = \max_{|\psi\rangle} \langle O' \rangle_\psi = \alpha$ of the flipped model, and therefore the efficiency of its evaluation, as $\mathcal{O}(\|O'\|_\infty^2/\varepsilon^2)$ measurements of $\rho'(\boldsymbol{\theta})$ are needed in order to estimate $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ to additive error ε (see Appendix 6.B.1 for a derivation). Therefore, we end up showing that, for a conventional model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ acting on n qubits and with a bounded observable trace norm $\|O\|_1 \leq \alpha$, we can construct a flipped model acting on $m = n + 1$ qubits and with observable spectral norm $\|O'\|_\infty = \alpha$.

Interestingly, in the relevant regime where the number of qubits n, m used by the linear models involved in this flipping is logarithmic in $\|O\|_1$ (e.g., where O is a Pauli observable and hence $\|O\|_1 = 2^n$), we find that this requirement on the spectral norm $\|O'\|_\infty$ of the resulting flipped model is unavoidable in the worst case, up to a logarithmic factor in $\|O\|_1$. We refer to Appendix 6.B.3 for proof of these statements and a more in-depth discussion.

Another property of interest in machine learning is the generalization performance of a learning model. That is, we want to bound the gap between the performance of the model on its training set (so-called training error) and its performance on the rest of the data space (or expected error). Such bounds have for instance been derived in terms of the number of

[1] The sign of the eigenvalues of the observable $O(\boldsymbol{\theta})$ can be taken into account using an auxiliary qubit, without overheads in the efficiency of evaluation. See Appendix 6.B.3 for more details.

encoding gates in the quantum model [121], or the rank of its observable [122]. In the case of flipped model, we find instead a bound in terms of the number of qubits n and the spectral norm $\|O\|_\infty$ of the observable. Since these quantities are operationally meaningful, this gives us a natural way of controlling the generalization performance of our flipped models. Stated informally, we find that if a flipped model achieves a small error $|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \eta$ for all $\boldsymbol{x}$ in a training set of size M, then we only need M to scale as $\widetilde{\Omega}\left(\frac{n\|O\|_\infty^2}{\varepsilon\eta^2}\right)$ in order to guarantee a small expected error $|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq 2\eta$ with probability $1-\varepsilon$ over the entire data distribution.

Note that the dependence on n and $\|O\|_\infty$ is linear and quadratic, respectively, which means that we can afford a large number of qubits and a large spectral norm and still guarantee a good generalization performance. This is particularly relevant as the spectral norm is a controllable quantity, meaning we can easily fine-tune our models to perform well in training and generalize well. E.g., in the case of the model in Equation 6.5, this spectral norm is bounded by $\max_{\boldsymbol{x}} \sum_{j=1}^{m} |w_i(\boldsymbol{x})|$, which scales favorably with the number of qubits n if $m \in \mathcal{O}(\text{poly}(n))$ or if the vector $\boldsymbol{w}(\boldsymbol{x})$ is sparse.

6.2.3 Quantum advantage of a shadow model

We recall that we (informally) define shadow models as models that are trained on a quantum computer, but, after a shadowing procedure that collects information on the trained model, are evaluated classically on new input data. In this section, we consider the question of achieving a quantum advantage using such shadow models. It may seem at first sight that this question has a straightforward answer, which is "no": if the function learned by a model is classically computable, then there should be no room for a quantum advantage. However, as demonstrated in Refs. [109, 123], one can also achieve a quantum advantage based on so-called *trap-door functions*. These are functions that are believed to be hard to compute classically, unless given a key (or advice) that allows for an efficient classical computation. Notably, there exist trap-door functions where this key can be efficiently computed using a quantum computer, but not classically. This allows us to construct shadow models that make use of this quantum-generated key to compute an otherwise classically untractable function.

Similarly to related results showing a quantum advantage in machine learning with classical data [108, 124], we consider a learning task where the target function (i.e., the function generating the training data) is derived from cryptographic functions that are widely believed to be

hard to compute classically. More precisely, we introduce a variant of the discrete cube root learning task [109], which is hard to solve classically under a hardness assumption related to that of the RSA cryptosystem [125]. In this task, we consider target functions defined on $\mathbb{Z}_N = \{0, \ldots, N-1\}$ as

$$g_s(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \sqrt[3]{\boldsymbol{x}} \bmod N \in [s, s + \frac{N-1}{2}], \\ 0, & \text{otherwise} \end{cases} \tag{6.6}$$

where $N = pq$ is an n-bit integer, product of two primes $p,\ q$ of the form $3k+2,\ 3k'+2$, such that the discrete cube root is properly defined as the inverse of the function $\boldsymbol{y}^3 \bmod N$. These target functions are particularly appealing because of a number of interesting properties:

(i) It is believed that given only $\boldsymbol{x}$ and N as input, computing $g(\boldsymbol{x}) = \sqrt[3]{\boldsymbol{x}} \bmod N$ with high probability of success over random draws of $\boldsymbol{x}$ and N is classically intractable. This assumption is known as the discrete cube root (DCR) assumption.

(ii) On the other hand, computing $\boldsymbol{x}^a \bmod N$ is classically efficient for any $a \in \mathbb{Z}_N$. For $a = 3$, this implies that $g^{-1}(\boldsymbol{y}) = \boldsymbol{y}^3 \bmod N$ is a one-way function, under the DCR assumption.

(iii) The function $g(\boldsymbol{x}) = \sqrt[3]{\boldsymbol{x}} \bmod N$ has a "trap-door", in that there exists another way of computing it efficiently. For every N (as specified above), there exists a *key* $d \in \mathbb{Z}_N$ such that $g(\boldsymbol{x}) = \boldsymbol{x}^d \bmod N$. Finding d is efficient quantumly by using Shor's factoring algorithm [126], but hard classically under the DCR assumption.

Observations (i) and (ii) can be leveraged to show that learning the functions g_s from examples is also intractable. Indeed, Alexi *et al.* [127] showed that a classical algorithm that could faithfully capture a single bit $g_s(\boldsymbol{x})$ of the discrete cube root of $\boldsymbol{x}$, for even a $1/2 + 1/\mathrm{poly}(n)$ fraction of all $\boldsymbol{x} \in \mathbb{Z}_N$, could also be used to reconstruct $g(\boldsymbol{x}),\ \forall \boldsymbol{x} \in \mathbb{Z}_N$, with high probability of success. Since, from observation (ii), the training data for the learning algorithm can also be generated efficiently classically from N, a classical learner that learns $g_s(\boldsymbol{x})$ correctly for a $1/2 + 1/\mathrm{poly}(n)$ fraction of all $\boldsymbol{x} \in \mathbb{Z}_N$ would then contradict the DCR assumption.

Observation (iii) allows us to define the following flipped model:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta}) O(\boldsymbol{x})] \tag{6.7}$$
$$\rho(\boldsymbol{\theta}) = |d', s'\rangle\langle d', s'| \ \ \& \ O(\boldsymbol{x}) = \sum_{d', s'} \widehat{g}_{d', s'}(\boldsymbol{x}) \, |d', s'\rangle\langle d', s'| \, .$$

That is, $\rho(\boldsymbol{\theta})$ (for $\boldsymbol{\theta} = (N, s')$) specifies candidates for the key d' and the parameter s' of interest, while $O(\boldsymbol{x})$ uses that information to compute

$$\widehat{g}_{d',s'}(\boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x}^{d'} \bmod N \in [s', s' + \frac{N-1}{2}], \\ 0, & \text{otherwise.} \end{cases} \tag{6.8}$$

The state $\rho(\boldsymbol{\theta}) = |d, s'\rangle\langle d, s'|$ for the right key d can be prepared efficiently using Shor's algorithm applied on N (provided with the training data). As for $O(\boldsymbol{x})$, it simply processes classically a bit-string to compute $\widehat{g}_{d',s'}(\boldsymbol{x})$ efficiently, which corresponds to $g_s(\boldsymbol{x})$ when $(d', s') = (d, s)$. Finding an s' close to s is an easy task given training data and $d' = d$. Since $\rho(\boldsymbol{\theta})$ is a computational basis state, this flipped model admits a trivial shadow model where a single computational basis measurement of $\rho(\boldsymbol{\theta})$ allows to evaluate $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ classically for all $\boldsymbol{x}$. Therefore, we end up showing the following theorem:

Theorem 6.1 (Quantum advantage (informal))
There exists a learning task where a shadow model first trained using a quantum computer then evaluated classically on new input data, can achieve an arbitrarily good learning performance, while any fully classical model cannot do significantly better than random guessing, under the hardness of classically computing the discrete cube root.

In Appendix 6.C we formalize the statement of this result using the PAC framework and provide more details on the setting and the proofs.

6.3 General shadow models

As mentioned at the start of this chapter, shadow models are not limited to shadowfied flipped models, and the main alternative proposals are based on the Fourier representation of quantum models [111, 112]. It is clear that Fourier models are defined very differently from flipped models, but one may wonder whether they nonetheless include shadowfied flipped models as a special case, or the other way around.

In this section, we first start by showing that there exist quantum models that admit shadow models (i.e., are shadowfiable) but cannot be shadowfied efficiently using a Fourier approach. This then motivates our proposal for a general definition of shadow models, and we show that, under this definition, all shadow models can be expressed as shadowfied flipped models. Finally, we show the existence of quantum models that are not shadowfiable at all under likely complexity theory assumptions.

6.3.1 Shadow models beyond Fourier

An interesting approach to construct shadows of quantum models is based on their natural Fourier representation. It has been shown [113, 128] that quantum models can be expressed as generalized Fourier series of the form

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{\boldsymbol{\omega}\in\Omega} c_{\boldsymbol{\omega}}(\boldsymbol{\theta}) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}} \tag{6.9}$$

where the accessible frequencies Ω only depend on properties of the encoding gates used by the model (notably the number of encoding gates and their eigenvalues). Since these frequencies can easily be read out from the circuit, one can proceed to form a shadow model by estimating their associated coefficients $c_{\boldsymbol{\omega}}(\boldsymbol{\theta})$ using queries of the quantum model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ at different values $\boldsymbol{x}$ and, e.g., a Fourier transform [111]. Given a good approximation of these coefficients, one can then compute estimates of $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for arbitrary new inputs $\boldsymbol{x}$. We will refer to such a shadowing approach that considers the quantum model as a black-box, aside from the knowledge of its Fourier spectrum, as the Fourier shadowing approach.

Although we will be explicit about this in the next subsection, we will consider a shadowing procedure to be successful, if, with high probability, the resulting shadow model agrees with the original model on all inputs[2], i.e.,

$$\max_{\boldsymbol{x}\in\mathcal{X}} \left| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right| \leq \varepsilon, \tag{6.10}$$

for a specified $\varepsilon \geq 0$.

We show that the Fourier shadowing approach can suffer from an exponential sample complexity in the dimension of the input data $\boldsymbol{x}$, making it intractable for high-dimensional input spaces. To see this, consider the linear model:

$$\begin{gathered} f_{\boldsymbol{y}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x}) O(\boldsymbol{y})] \\ \rho(\boldsymbol{x}) = \bigotimes_{i=1}^{n} R_Y(x_i) \left|0\right\rangle\!\left\langle 0\right| R_Y^\dagger(x_i) \;\&\; O(\boldsymbol{y}) = \left|\boldsymbol{y}\right\rangle\!\left\langle \boldsymbol{y}\right| . \end{gathered} \tag{6.11}$$

for $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{y} \in \{0,1\}^{\otimes n}$. Let us first restrict our attention to the domain $\boldsymbol{x} \in \{0,\pi\}^n$. It is quite clear that on this domain, $f_{\boldsymbol{y}}(\boldsymbol{x}) = \delta_{\boldsymbol{x}/\pi,\boldsymbol{y}}$ plays the role of a database search oracle, where the database has 2^n

[2] We want the shadowing procedure to be successful independently of the data distribution under which the model should be trained, which justifies this definition. We discuss this point further in Appendix 6.D.4.

elements and a unique marked element $\boldsymbol{y}$. From lower bounds on database search, we know that $\Omega(2^n)$ calls to this oracle are needed to find $\boldsymbol{y}$ [129]. This implies that a Fourier shadowing approach would require $\Omega(2^n)$ calls to $f_{\boldsymbol{y}}(\boldsymbol{x}) = \delta_{\boldsymbol{x}/\pi,\boldsymbol{y}}$ in order to guarantee $\max_{\boldsymbol{x}\in\mathcal{X}} \left|\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right| \leq 1/4$. In Appendix 6.D.3, we explain how this result can be generalized to the full domain $\boldsymbol{x} \in \mathbb{R}^n$, and we relate this bound on the sample complexity to the Fourier decomposition of the model.

On the other hand, note that the flipped model associated to $f_{\boldsymbol{y}}(\boldsymbol{x})$ allows for a straightforward shadowing procedure. Indeed, by preparing $O(\boldsymbol{y})$ and measuring it in the computational basis, one straightforwardly obtains $\boldsymbol{y}$ and can therefore classically compute the expectation value of any tensor product observable $\rho(\boldsymbol{x})$ as specified by Equation 6.11. Therefore, we have shown that there exist shadowfiable models that are not efficiently Fourier-shadowfiable, i.e., for which a shadowing procedure based solely on the knowledge of their Fourier spectrum and on black-box queries has query complexity that is exponential in the input dimension.

6.3.2 All shadow models are shadows of flipped models

We give a general definition of shadow models that can encompass all methods that have been proposed to generate them. In contrast to the definition of classical surrogates proposed by Schreiber *et al.* [111], we give explicit definitions for the shadowing and evaluation phases of shadow models which makes explicit the need for a quantum computer in the shadowing phase. Indeed, as mentioned in the introduction, the term *classical surrogate* has been used to describe both a classically evaluatable model obtained from a quantum shadowing procedure and a fully classical model trained directly on the data. We want to avoid this confusion in the definition of shadow models. We view a general shadowing phase as the generation of *advice* that can be used to classically evaluate a quantum model. This advice is generated by the execution of quantum circuits that may or may not depend on the (trained) quantum circuit from the training phase. For instance, when we shadowfy a flipped model, we simply prepare the parametrized states $\rho(\boldsymbol{\theta})$ and use (randomized) measurements to generate an operationally meaningful classical description. In the case of Fourier shadowing, this advice is instead generated by evaluations of the quantum model $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ for different inputs $\boldsymbol{x} \in \mathbb{R}^d$ that are rich enough to learn the Fourier coefficients of this model. We propose the following definition:

Definition 6.3.1 (General shadow model)
Let $W_1(\boldsymbol{\theta}), ..., W_M(\boldsymbol{\theta})$ be a sequence of $\mathcal{O}(poly(m))$-time quantum circuits

applied on all-zero states $|0\rangle^{\otimes m}$*, and that can potentially be chosen adaptively. Call* $\omega(\boldsymbol{\theta}) = (\omega_1(\boldsymbol{\theta}), \ldots, \omega_M(\boldsymbol{\theta}))$ *the outcomes of measuring the output states of these circuits in the computational basis. A general shadow model is defined as:*

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta})) \tag{6.12}$$

where $\mathcal{A}$ *is a classical* $\mathcal{O}(poly(M, m, d))$*-time algorithm that processes the outcomes* $\omega(\boldsymbol{\theta})$ *along with an input* $\boldsymbol{x} \in \mathbb{R}^d$ *to return the (real-valued) label* $f_{\boldsymbol{\theta}}(\boldsymbol{x})$*.*

From this definition, a shadow model is a classically evaluatable model that uses quantum-generated advice. Crucially, this advice must be independent of the data points $\boldsymbol{x}$ we wish to evaluate the model on in the future. We distinguish the notion of a shadow model from that of a *shadowfiable* quantum model, that is a quantum model that admits a shadow model:

Definition 6.3.2 (Shadowfiable model)
A model $f_{\boldsymbol{\theta}}$ *acting on* n *qubits is said to be shadowfiable if, for* $\varepsilon, \delta > 0$*, there exists a shadow model* $\widetilde{f}_{\boldsymbol{\theta}}$ *such that, with probability* $1 - \delta$ *over the quantum generation of the advice* $\omega(\boldsymbol{\theta})$ *(i.e., the shadowing phase), the shadow model satisfies*[2]

$$\max_{\boldsymbol{x} \in \mathcal{X}} \left| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right| \leq \varepsilon, \tag{6.13}$$

and uses $m, M \in \mathcal{O}(poly(n, 1/\varepsilon, 1/\delta))$ *qubits and circuits to generate its advice* $\omega(\boldsymbol{\theta})$*.*

While we have seen that there exist shadowfiable models that cannot be shadowfied efficiently using a Fourier approach, we show that all shadowfiable models as defined above can be approximated by shadowfiable flipped models.

Lemma 1 (Flipped models are shadow-universal)
All shadowfiable models as defined in Defs. 6.3.1 and 6.3.2 can be approximated by flipped models $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ *with the guarantee that computational basis measurements of* $\rho(\boldsymbol{\theta})$ *and efficient classical post-processing can be used to evaluate* $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ *to good precision with high probability.*

This result is essentially based on the observation that the evaluation of a general shadow model as defined in Def. 6.3.1 can be done entirely coherently. Instead of classically running the algorithm $\mathcal{A}$ using the random advice $\omega(\boldsymbol{\theta})$, one can quantumly simulate this algorithm (using

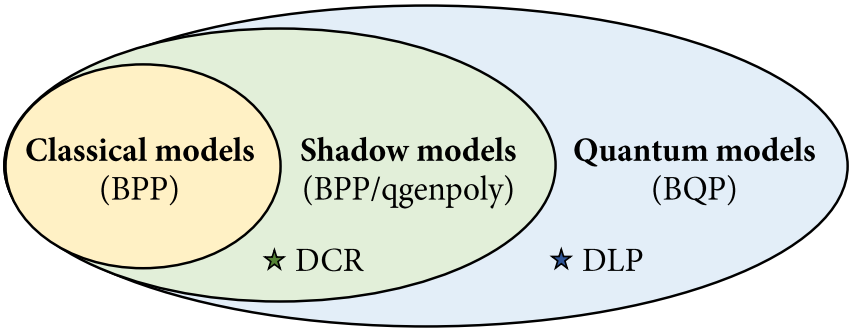

Figure 6.2: Separations between classical, shadow, and quantum models. Under the assumption that the discrete cube root (DCR) cannot be computed classically in polynomial time, we have a separation between shadow models (captured by the class BPP/qgenpoly) and classical models (in BPP). Under the assumption that there exist functions that can be computed in quantum polynomial time but not in classical polynomial time with the help of advice (i.e., BQP $\not\subset$ P/poly), we have a separation between quantum models (universal for BQP) and shadow models (BPP/qgenpoly). A candidate function for this separation is the discrete logarithm (DLP).

a reversible execution) and execute it on the coherent advice $\rho(\boldsymbol{\theta}) = |\omega(\boldsymbol{\theta})\rangle\langle\omega(\boldsymbol{\theta})|$ generated by $\{W_1(\boldsymbol{\theta}), ..., W_M(\boldsymbol{\theta})\}$ before the computational basis measurements. We refer to Appendix 6.D.1 for a more detailed statement and proof.

6.3.3 Not all quantum models are shadowfiable

From the discrete cube root learning task, we already understand that a learning separation can be established between classical and shadowfiable models. We would also like to understand whether a learning separation exists between shadowfiable models and general quantum models, or equivalently, whether all quantum models are shadowfiable. We show that this also is not the case, under widely believed assumptions (see Fig. 6.2).

Theorem 6.2 (Not all shadowfiable)
Under the assumption that BQP $\not\subset$ P/poly, there exist quantum models, i.e., models in BQP, that are not shadowfiable, i.e., that are not in BPP/qgenpoly.

We start by noting that shadow models can be characterized by a

complexity class we define as BPP/qgenpoly,[3] which contains all functions that can be computed efficiently classically with the help of polynomially-sized advice *generated efficiently by a quantum computer*. This class is trivially contained in the standard class BPP/poly, which doesn't have any constraint on how the advice is generated and can be derandomized to P/poly (i.e., BPP/poly=P/poly [130]). Note however that BPP/qgenpoly constitutes a physically relevant class, since it only contains problems that can be solved efficiently by classical and quantum computers, as opposed to P/poly, which contains undecidable problems, such as a version of the halting problem. We refer to Appendix 6.A.3 for formal definitions of these complexity classes, and an in-depth discussion.

On the other hand, it is easy to show that quantum models (more precisely quantum linear models) can also represent any function in BQP, i.e., all functions that are efficiently computable on a quantum computer. For this, one simply takes a simple encoding of an n-bit input $\boldsymbol{x}$:

$$\rho(\boldsymbol{x}) = \bigotimes_{i=1}^{n} X_i^{x_i} \left|0\right\rangle\!\left\langle 0\right| X_i^{x_i} \tag{6.14}$$

along with an observable

$$O_n = U_n^\dagger Z_1 U_n \tag{6.15}$$

specified by an arbitrary n-qubit circuit U_n in BQP and the Pauli-Z operator applied on it first qubit. The resulting model $f_n(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O_n]$ can then be used to decide any language in BQP.

Combining these two observations, we get that the proposition "all quantum models are shadowfiable" would imply that BQP $\subseteq$ BPP/qgenpoly $\subseteq$ P/poly, which violates the widely-believed conjecture [15] that BQP $\not\subseteq$ P/poly (see Appendix 6.D.2 for a formal proof). To give an example of candidates of non-shadowfiable quantum models, the discrete logarithm $\log_g x \bmod p$ (or even one bit of it) is provably in BQP but is not believed to be in P/poly. Therefore, a model that could be used to compute the discrete logarithm (e.g., the quantum model of Liu *et al.* [108]) is likely not shadowfiable.

[3] Stands for Bounded-error Probabilistic Polynomial-time with quantumly generated (polynomial-time) advice of polynomial size. In this chapter, we talk mostly about complexity classes for decision problems. Note however that for models, since these compute real-valued functions (represented to machine precision), we should instead consider the function-problem version of these complexity classes.

6.4 Discussion

In this work, we examined the class of quantumly trainable, classically evaluatable models we refer to as shadow models. Our analysis has shown that these models can be universally captured by a restricted family of quantum linear models, wherein data-encoding and variational operations are flipped compared to conventional quantum models. Furthermore, we demonstrated that shadow models belong to an intriguing complexity class, coined BPP/qgenpoly, exhibiting superiority over classical models (in BPP) but inferiority to fully quantum models (in BQP), based on prevalent complexity theory assumptions.

By presenting shadows models as flipped linear models, we illustrated how shadow tomography protocols could be applied straightforwardly to construct shadow models in practice. Yet, it is important to note a crucial distinction between a shadow tomography scenario and a shadow model: in the latter, one has control over the quantum state intended for shadowing. This distinction introduces new possibilities for devising 'state-aware' shadow tomography protocols aimed at constructing shadow models. This could potentially alleviate some of the limitations of current classical shadow protocols.

Considering our findings on learning separations, we identified a noteworthy characteristic of shadow models: their ability to quantumly compute useful advice for a classical evaluation algorithm, enabling them to tackle otherwise classically-intractable tasks. The example we presented, based on trap-door functions, readily allows for such constructions, but it remains somewhat contrived. Exploring similar constructions for physically-relevant problems, such as predicting ground state properties of complex quantum systems, would be an intriguing avenue for future research.

6.A Formal definitions

6.A.1 Linear models

Definition 6.A.1 (Conventional linear model)
Let $U(\boldsymbol{x})$ be an encoding quantum circuit that is parametrized by input data $\boldsymbol{x} \in \mathbb{R}^d$, ρ_0 a fixed input quantum state (diagonal in the computational basis), $V(\boldsymbol{\theta})$ a variational quantum circuit parametrized by a vector $\boldsymbol{\theta} \in \mathbb{R}^p$ and $O = \sum_{i=1}^m w_i O_i$ an observable specified by a (trainable) linear combination of Hermitian matrices $\{O_i\}_{i=1}^m$. A conventional linear model is defined by the parametrized function:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x}) O(\boldsymbol{\theta})] \tag{6.16}$$

for $\rho(\boldsymbol{x}) = U(\boldsymbol{x})\rho_0 U^\dagger(\boldsymbol{x})$ and $O(\boldsymbol{\theta}) = V^\dagger(\boldsymbol{\theta}) O V(\boldsymbol{\theta})$ (when the weights $\{w_i\}_{i=1}^m$ are also trainable, we include them in the parameters $\boldsymbol{\theta}$ of the model).

Definition 6.A.2 (Flipped model)
Let $V(\boldsymbol{\theta})$ be a variational quantum circuit parametrized by a vector $\boldsymbol{\theta} \in \mathbb{R}^p$, ρ_0 a fixed input quantum state (diagonal in the computational basis), $U(\boldsymbol{x})$ an encoding quantum circuit that is parametrized by input data $\boldsymbol{x} \in \mathbb{R}^d$ and $O_{\boldsymbol{x}} = \sum_{i=1}^m w(\boldsymbol{x})_i O_i$ an observable specified by a linear combination of Hermitian matrices $\{O_i\}_{i=1}^m$, weighted by a data-dependent function $w : \mathbb{R}^d \to \mathbb{R}^m$. A flipped model is defined by the parametrized function:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta}) O(\boldsymbol{x})] \tag{6.17}$$

for $\rho(\boldsymbol{\theta}) = V(\boldsymbol{\theta})\rho_0 V^\dagger(\boldsymbol{\theta})$ and $O(\boldsymbol{x}) = U^\dagger(\boldsymbol{x}) O_{\boldsymbol{x}} U(\boldsymbol{x})$.

6.A.2 Shadow models

Definition 6.A.3 (Shadow model)
Let $\{W_1(\boldsymbol{\theta}), ..., W_M(\boldsymbol{\theta})\}$ be a sequence of m-qubit unitary circuits that are dependent on a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, and can potentially be chosen adaptively. We define the quantum-generated *advice $\omega(\boldsymbol{\theta}) = (\omega_1(\boldsymbol{\theta}), \ldots, \omega_M(\boldsymbol{\theta}))$ as the measurement outcomes $\omega_i(\boldsymbol{\theta})$ obtained by measuring the states $W_i(\boldsymbol{\theta}) |0\rangle^{\otimes m}$ in the computational basis (and a description of their associated circuits $W_i(\boldsymbol{\theta})$). A shadow model is defined as the parametrized function:*

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta})) \tag{6.18}$$

for $\mathcal{A}$ a classical $\mathcal{O}(poly(m, M, d))$-time algorithm that takes as input the advice $\omega(\boldsymbol{\theta})$, a data vector $\boldsymbol{x} \in \mathbb{R}^d$ and outputs a real-valued label $f_{\boldsymbol{\theta}}(\boldsymbol{x})$

Examples of shadow models:

- Take a flipped model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ for $\rho(\boldsymbol{\theta})$ a quantum state generated by a circuit $V(\boldsymbol{\theta})$ applied to $|0\rangle^{\otimes n}$ and $O(\boldsymbol{x}) = \sum_{i=1}^{m} w(\boldsymbol{x})_i P_i$ where $\{P_i\}_{i=1}^{m}$ are all k-local Pauli strings acting on n qubits.
  A simple shadow model associated to this flipped model consists in estimating all the expectation values $\langle P_i \rangle \approx \mathrm{Tr}[\rho(\boldsymbol{\theta})P_i]$ via repeated measurements of $\rho(\boldsymbol{\theta})$ in the eigenbasis of each of the $m = \binom{n}{k}3^k$ Pauli strings P_i, and taking their weighted combination $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{m} w(\boldsymbol{x})_i \langle P_i \rangle$. In this case, the unitary circuits $\{W_1(\boldsymbol{\theta}), ..., W_M(\boldsymbol{\theta})\}$ are simply obtained by $V(\boldsymbol{\theta})$ followed by a basis change unitary (corresponding to the Pauli basis of P_i). As for the classical algorithm $\mathcal{A}$, this is simply a collection of mean estimators that compute estimates $\langle P_i \rangle$ out of measurement outcomes, followed by the computation of a weighted sum. The number of measurements needed for this shadow model to guarantee $\left|\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right| \leq \varepsilon, \ \forall \boldsymbol{x} \in \mathbb{R}^d$ is $M \in \widetilde{\mathcal{O}}\left(\frac{m \max_{\boldsymbol{x}} \|w(\boldsymbol{x})\|_1^2}{\varepsilon^2}\right)$. Indeed, estimating each $\langle P_i \rangle$ to additive error $\frac{\varepsilon}{\max_{\boldsymbol{x}} \|w(\boldsymbol{x})\|_1}$ allows us to guarantee the desired total additive error, and each of this estimates can be obtained using $\widetilde{\mathcal{O}}\left(\frac{\max_{\boldsymbol{x}} \|w(\boldsymbol{x})\|_1^2}{\varepsilon^2}\right)$ samples.
  A more interesting shadow model relies on Pauli classical shadows [131] where random Pauli measurements are used to construct $\omega(\boldsymbol{\theta})$. Median-of-mean estimators then use these measurement outcomes to compute empirical estimates $\hat{\rho}(\boldsymbol{\theta})$ of $\rho(\boldsymbol{\theta})$ and approximate all expectation values $\mathrm{Tr}[\rho(\boldsymbol{\theta})P_i]$. The advantage of this shadow model is that it requires $M \in \widetilde{\mathcal{O}}\left(\frac{3^k \max_{\boldsymbol{x}} \|w(\boldsymbol{x})\|_1^2}{\varepsilon^2}\right)$ measurements for the same guarantees as the shadow model above, which constitutes savings of a factor $\widetilde{\mathcal{O}}\left(\binom{n}{k}\right)$.

- Another interesting flipped model that can be turned into a shadow model: $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ for $\rho(\boldsymbol{\theta})$ arbitrarily defined and $O(\boldsymbol{x}) = |\psi(\boldsymbol{x})\rangle\langle\psi(\boldsymbol{x})|$, for some pure states $|\psi(\boldsymbol{x})\rangle$. Given that, $\mathrm{Tr}\left[O(\boldsymbol{x})^2\right] = 1, \ \forall \boldsymbol{x} \in \mathbb{R}^d$, then Clifford classical shadows [131] allow to construct a representation $\omega(\boldsymbol{\theta})$ of $\rho(\boldsymbol{\theta})$ that guarantees $\left|\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(\boldsymbol{x})\right| \leq \varepsilon, \ \forall \boldsymbol{x} \in \mathbb{R}^d$ using only $M \in \widetilde{\mathcal{O}}\left(\frac{1}{\varepsilon^2}\right)$ measurements. However, for

this estimation to be computationally efficient, the states $|\psi(\boldsymbol{x})\rangle$ need to be stabilizer states, or be generated by few (i.e., $\mathcal{O}(\log(n))$) non-Clifford gates [132].

- Consider the model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sum_{i=1}^{m} w(\boldsymbol{x})_i \operatorname{Tr}[\rho(\boldsymbol{\theta})P_i]^2$ for $m = 4^n$, i.e., $\{P_i\}_{i=1}^m$ are all n-qubit Pauli strings. By preparing two-copy states $\rho(\boldsymbol{\theta}) \otimes \rho(\boldsymbol{\theta})$ and performing simultaneous Bell measurements between pairs of qubits of these two copies, $M = \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$ such measurements give a $\omega(\boldsymbol{\theta})$ rich enough to compute any $\operatorname{Tr}[\rho(\boldsymbol{\theta})P_i]^2$ to precision ε [133]. Therefore, evaluating $f_{\boldsymbol{\theta}}(\boldsymbol{x})$ requires only $M \in \widetilde{\mathcal{O}}\left(\frac{\max_{\boldsymbol{x}} \|w(\boldsymbol{x})\|_1^2}{\varepsilon^2}\right)$ measurements using this shadow model, compared to $2^{\Omega(n)}$ for a shadow model that would construct $\omega(\boldsymbol{\theta})$ using single-copy measurements only (assuming that for all $i \in \{1, \ldots, m\}$, there exists an $\boldsymbol{x} \in \mathbb{R}^d$ such that $w(\boldsymbol{x})_i \neq 0$). For a $w(\boldsymbol{x})$ that is k-sparse for all $\boldsymbol{x}$, with $k \ll m$, this constitutes an exponential separation.

6.A.3 Complexity classes

Definition 6.A.4 (BQP)
A language L is in BQP if and only if there exists a polynomial-time uniform family of quantum circuits $\{U_n : n \in \mathbb{N}\}$, such that

1. *For all $n \in \mathbb{N}$, U_n takes as input an n-qubit computational basis state, and outputs one bit obtained by measuring the first qubit in the computational basis.*
2. *For all $x \in L$, the probability that the output of $U_{|x|}$ applied on the input x is 1 is greater or equal to $2/3$.*
3. *For all $x \notin L$, the probability that the output of $U_{|x|}$ applied on the input x is 0 is greater or equal to $2/3$.*

Definition 6.A.5 (P/poly)
A language L is in P/poly if and only if there exists a polynomial-time classical algorithm $\mathcal{A}$ and a sequence of polynomial-size advice strings $\{\alpha_n \in \{0,1\}^{poly(n)}\}_{n\in\mathbb{N}}$ such that for all $n \in \mathbb{N}$ and all $x \in \{0,1\}^n$:

$$\mathcal{A}(x, \alpha_n) = 1 \iff x \in L. \tag{6.19}$$

Definition 6.A.6 (BPP/qgenpoly)
A language L is in BPP/qgenpoly if and only if there exists a polynomial-time uniform family of quantum circuits $\{U_n : n \in \mathbb{N}\}$ and a polynomial-time probabilistic classical algorithm $\mathcal{A}$, such that

1. *For all $n \in \mathbb{N}$, U_n takes as input the computational basis state $|0\rangle^{\otimes m}$ for $m \in \mathcal{O}(poly(n))$, and outputs m bits obtained by measuring all qubits in the computational basis. This constitutes the* quantum-generated *advice ω_n.*

2. *For all $x \in \{0,1\}^*$, $\mathcal{A}$ takes as input x and $\omega_{|x|}$.*

3. *For all $x \in L$, the probability that $\mathcal{A}(x, \omega_{|x|})$ outputs 1 is greater or equal to $2/3$, taken over the randomness of $\omega_{|x|}$ and the internal randomness of $\mathcal{A}$.*

4. *For all $x \notin L$, the probability that $\mathcal{A}(x, \omega_{|x|})$ outputs 0 is greater or equal to $2/3$, taken over the randomness of $\omega_{|x|}$ and the internal randomness of $\mathcal{A}$.*

The following inclusions are easy to show:

$$\mathsf{BPP} \overset{(i)}{\subseteq} \mathsf{BPP/qgenpoly} \overset{(ii)}{\subseteq} \mathsf{BQP}. \tag{6.20}$$

(i) follows from making the quantum-generated advice ω_n empty in the definition of BPP/qgenpoly. (ii) follows from the ability to efficiently simulate classical computations on a quantum computer. As illustrated in Figure 6.4, the algorithm $\mathcal{A}$ in the definition of BPP/qgenpoly can be simulated unitarily, and absorbed in the uniform family of quantum circuits $\{U_n : n \in \mathbb{N}\}$, resulting in polynomial-time quantum circuits that fit the definition of BQP.

Our learning separations results, i.e., Theorem 6.1 (Lemmas 6.C.3 and 6.C.4 in the Appendix) and Theorem 6.2 (Lemma 6.D.2 in the Appendix), can be seen as evidence that the inclusions (i) and (ii) are strict, based on complexity-theory assumptions. Other notable inclusions that are useful to prove our results are:

$$\mathsf{BPP/qgenpoly} \subsetneq \mathsf{BPP/poly} = \mathsf{P/poly}. \tag{6.21}$$

The equality BPP/poly = P/poly follows from the derandomization results of Adleman [130], which show that the random errors made by an algorithm in BPP/poly can be canceled by an appropriate choice of the random bits used by the randomized algorithm, which are then appended to the poly-sized advice to obtain an algorithm in P/poly. The inclusion BPP/qgenpoly $\subseteq$ BPP/poly simply comes from the fact that BPP/poly is not restricted in *how* the poly-sized advice is generated, and the remainder of its definition is identical to that of BPP/qgenpoly. The restriction we

make on *how* the advice is generated in BPP/qgenpoly makes it a physically relevant complexity class, as opposed to (BP)P/poly. All problems in BPP/qgenpoly can be solved efficiently (i.e., in polynomial time) using (classical and) quantum computers, while P/poly notably contains undecidable problems. Consider for instance the unary version of the halting problem: UHALT $= \{1^n, \text{where } n \text{ encodes } (M, x) \text{ such that the Turing machine } M \text{ halts on } x\}$ is an undecidable language (as any algorithm that would decide it would also be able to decide the traditional halting problem), but by considering the advice $\alpha_n = 1$ if $1^n \in$ UHALT and 0 otherwise (uniquely defined for each input size n), one trivially obtains an algorithm in P/poly that solves it. The fact that this advice cannot be generated by a uniform family of (poly-time) circuits is irrelevant for the class P/poly, which is at the source of this result. However it is relevant for the class BPP/qgenpoly, and, in fact, having UHALT $\in$ BPP/qgenpoly would mean that one could solve an undecidable problem using uniform (poly-time) circuits. The impossibility of this hypothetical result gives BPP/qgenpoly $\subsetneq$ P/poly.

6.B Properties of flipped models

6.B.1 Sample complexity of evaluating quantum models

Consider a linear quantum model (either conventional or flipped) of the form

$$f_{\boldsymbol{y}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})], \tag{6.22}$$

for a quantum state $\rho(\boldsymbol{x})$ parametrized by a vector $\boldsymbol{x} \in \mathbb{R}^d$ and $O(\boldsymbol{y})$ a Hermitian observable parametrized by a vector $\boldsymbol{y} \in \mathbb{R}^p$. Assume that we can prepare single copies of $\rho(\boldsymbol{x})$ and that we can measure them in the eigenbasis of $O(\boldsymbol{y})$. We ask: given error parameters $\varepsilon, \delta > 0$, how many such measurements of $\rho(\boldsymbol{x})$ do we need in order to compute an estimate $\widehat{f}_{\boldsymbol{y}}(\boldsymbol{x})$ of $f_{\boldsymbol{y}}(\boldsymbol{x})$ such that $\left|\widehat{f}_{\boldsymbol{y}}(\boldsymbol{x}) - f_{\boldsymbol{y}}(\boldsymbol{x})\right| \leq \varepsilon$ with success probability at least $1 - \delta$.

It is easy to see that this problem corresponds to a simple Monte Carlo mean estimation. Indeed, we can write a decomposition of $O(\boldsymbol{y})$ in its eigenbasis as:

$$O(\boldsymbol{y}) = \sum_i \lambda_i(\boldsymbol{y}) \, |\phi_i\rangle\langle\phi_i| \tag{6.23}$$

where $\lambda_i(\boldsymbol{y})$ is a real eigenvalue (since $O(\boldsymbol{y})$ is Hermitian) associated to the eigenstate $|\phi_i\rangle\langle\phi_i|$ (these eigenstates can in general also depend on $\boldsymbol{y}$,

but we do not write this dependence explicitly for ease of notation). We can also write a decomposition of $\rho(\boldsymbol{x})$ in this same basis as :

$$\rho(\boldsymbol{x}) = \sum_{i,j} \rho_{i,j}(\boldsymbol{x}) \left|\phi_i\right\rangle\!\left\langle\phi_j\right| \tag{6.24}$$

such that $\mathrm{Tr}[\rho(\boldsymbol{x})] = \sum_i \rho_{i,i}(\boldsymbol{x}) = 1$ by the unit-trace property of $\rho(\boldsymbol{x})$, and $\left\langle\phi_i\right| \rho(\boldsymbol{x}) \left|\phi_i\right\rangle = \rho_{i,i}(\boldsymbol{x}) \geq 0$ from its positive semi-definiteness. From these two properties, we deduce that $\{\rho_{i,i}(\boldsymbol{x})\}_i$ defines a probability distribution over the eigenstates $\{\left|\phi_i\right\rangle\!\left\langle\phi_i\right|\}_i$. Therefore, we can see that:

$$\mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})] = \mathrm{Tr}\left[\sum_{i,j,k} \rho_{i,j}(\boldsymbol{x})\lambda_k(\boldsymbol{y}) \left|\phi_i\right\rangle\!\left\langle\phi_j\middle|\phi_k\right\rangle\!\left\langle\phi_k\right|\right] \tag{6.25}$$

$$= \mathrm{Tr}\left[\sum_{i,j} \rho_{i,j}(\boldsymbol{x})\lambda_j(\boldsymbol{y}) \left|\phi_i\right\rangle\!\left\langle\phi_j\right|\right] \tag{6.26}$$

$$= \sum_i \rho_{i,i}(\boldsymbol{x})\lambda_i(\boldsymbol{y}) \tag{6.27}$$

simply corresponds to the expectation value of the random variable $i \mapsto \lambda_i(\boldsymbol{y})$ under the probability distribution $\{\rho_{i,i}(\boldsymbol{x})\}_i$, i.e., the probability distribution obtained by measuring $\rho(\boldsymbol{x})$ in the eigenbasis of $O(\boldsymbol{y})$.

Therefore, we can use known results from (classical) Monte Carlo estimation to bound the sample complexity of evaluating this mean value, and therefore the quantum model. With the assumption that $\rho(\boldsymbol{x})$ is given as a black-box and that it can generate arbitrary quantum states (and therefore arbitrary distributions $\{\rho_{i,i}(\boldsymbol{x})\}_i$), the only property of the random variable $i \mapsto \lambda_i(\boldsymbol{y})$ we can use to bound the sample complexity is its bounded domain. Indeed, without additional assumptions of the distribution, we have a tight sample complexity bound of

$$\Theta\left(\frac{B^2 \log\left(\delta^{-1}\right)}{\varepsilon^2}\right) \tag{6.28}$$

samples in order to estimate the mean of a random variable taking values in $[-B, B]$, to precision ε and with probability of success $1 - \delta$ [134, 135]. In the case of a quantum model, the random $i \mapsto \lambda_i(\boldsymbol{y})$ takes values in $[-\|O(\boldsymbol{y})\|_\infty, \|O(\boldsymbol{y})\|_\infty]$, where $\|O(\boldsymbol{y})\|_\infty$ is the spectral norm of the observable $O(\boldsymbol{y})$. Therefore, the sample complexity of estimating a

quantum model $f_{\boldsymbol{y}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})]$ is in

$$\Theta\left(\frac{\|O(\boldsymbol{y})\|_\infty^2 \log(\delta^{-1})}{\varepsilon^2}\right), \tag{6.29}$$

in the absence of any constraint (or information) on the quantum states $\rho(\boldsymbol{x})$.

6.B.2 Generalization performance

In this section, we study the generalization performance of flipped models. Our result can be stated informally in the following lemma:

Lemma 6.B.1 (Generalization bounds (informal))
Consider a flipped model $f_{\boldsymbol{\theta}}$ that acts on n qubits and has a bounded observable norm $\|O\|_\infty$. If this model achieves a small training error $|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \eta$ for all $\boldsymbol{x}$ in a dataset of size M, then it also has a small expected error $|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - f(\boldsymbol{x})| \leq 2\eta$ with probability $1-\varepsilon$ over the data distribution, provided that the size of the dataset scales as $M \geq \widetilde{\Omega}\left(\frac{n\|O\|_\infty^2}{\varepsilon\eta^2}\right)$.

To prove this result, we take an approach very similar to that of Aaronson [136], where we lower bound the number of qubits n and spectral norm $\|O\|_\infty$ needed by a flipped model to encode arbitrary k-bit strings, in a way that can be recovered efficiently via repeated measurements. These bounds naturally allow us to upper bound the fat-shattering dimension of flipped models, a complexity measure that is widely used in generalization bounds [137].

Encoding bit-strings in flipped models

Theorem 6.B.1
Let k and n be positive integers with $k > n$. For all k-bit strings $\boldsymbol{y} = y_1 \ldots y_k$, let $\rho(\boldsymbol{y})$ be an n-qubit mixed state that "encodes" $\boldsymbol{y}$, meaning each bitstring $\boldsymbol{y}$ is associated to an arbitrary n-qubit quantum state $\rho(\boldsymbol{y})$. Suppose there exist Hermitian observables $O_1, \ldots, O_k$ with spectral norms $\|O_i\|_\infty$ such that we call $\|O\|_\infty = \max_i \|O_i\|_\infty$, as well as real numbers $\alpha_1, \ldots, \alpha_k$, such that, for all $\boldsymbol{y} \in \{0,1\}^k$ and $i \in \{1, \ldots, k\}$,

(i) if $y_i = 0$, then $\mathrm{Tr}[\rho(\boldsymbol{y})O_i] \leq \alpha_i - \gamma$, and

(ii) if $y_i = 1$, then $\mathrm{Tr}[\rho(\boldsymbol{y})O_i] \geq \alpha_i + \gamma$.

Then $n\|O\|_\infty^2/\gamma^2 \in \Omega(k)$.

Proof. We take a similar approach to the proof of Aaronson [136], in which we show that a combination of an encoding $\rho(\boldsymbol{y})$ and observables $O_1, \ldots, O_k$ that satisfies guarantees (i) and (ii) would need $n\|O\|_\infty^2/\gamma^2$ to scale linearly in the length k of the bit-strings it encodes in order not to contradict with Holevo's bound.

Suppose by contradiction that such an encoding scheme exists with $n\|O\|_\infty^2/\gamma^2 \in o(k)$. We first adapt to the setting of Aaronson by constructing two-outcome POVMs $\{E_i, I - E_i\}$ out of the observables O_i. That is, we take the general Hermitian matrices O_i with eigenvalues in $[\lambda_{\min}, \lambda_{\max}] \subset [-\|O\|_\infty, \|O\|_\infty]$, and transform them into Hermitian matrices E_i with eigenvalues in $[0, 1]$, such that the POVM $\{E_i, I - E_i\}$ accepts ρ (i.e., outputs 1) with probability $\mathrm{Tr}(\rho E_i)$, and rejects ρ (i.e., outputs 0) with probability $1 - \mathrm{Tr}(\rho E_i)$. Specifically, we define

$$E_i = \frac{O_i + |\lambda_{\min}| I}{|\lambda_{\min}| + |\lambda_{\max}|}. \tag{6.30}$$

Conditions (i) and (ii) then translate to:

$$\begin{cases} (i') \text{ if } y_i = 0, \text{ then } \mathrm{Tr}[\rho(\boldsymbol{y}) E_i] \leq \frac{\alpha_i + |\lambda_{\min}|}{|\lambda_{\min}| + |\lambda_{\max}|} - \frac{\gamma}{|\lambda_{\min}| + |\lambda_{\max}|} \\ (ii') \text{ if } y_i = 1, \text{ then } \mathrm{Tr}[\rho(\boldsymbol{y}) E_i] \geq \frac{\alpha_i + |\lambda_{\min}|}{|\lambda_{\min}| + |\lambda_{\max}|} + \frac{\gamma}{|\lambda_{\min}| + |\lambda_{\max}|} \end{cases} \tag{6.31}$$

From here, by noting that $|\lambda_{\min}| + |\lambda_{\max}| \leq 2\|O\|_\infty$, we can directly apply Theorem 2.6 of Aaronson [136] and get our result, but we detail the reasoning further for clarity.

We first need to amplify the probability that we correctly identify whether $y_i = 0$ or 1 from measuring copies of $\rho(\boldsymbol{y})$, since the probabilities obtained from E_i can be arbitrarily small. Consider an amplified scheme, where each bit-string $\boldsymbol{y} \in \{0, 1\}^k$ is encoded by the tensor product $\rho(\boldsymbol{y})^{\otimes \ell}$, for some $\ell > 1$ to be defined later. For all $i \in \{1, \ldots, k\}$, let $\{E_i^*, I - E_i^*\}$ be the amplified POVM that applies $\{E_i, I - E_i\}$ to each of the ℓ copies of $\rho(\boldsymbol{y})$ and accepts if and only if at least $\widetilde{\alpha}_i \ell = \frac{\alpha_i + |\lambda_{\min}|}{|\lambda_{\min}| + |\lambda_{\max}|} \ell$ of these POVMs do. For all $j \in \{1, \ldots, \ell\}$, call $X_i^{(j)}$ the random variable that takes the value 1 if $\{E_i, I - E_i\}$ accepts the j-th copy of $\rho(\boldsymbol{y})$ (i.e., with probability $p_i = \mathrm{Tr}(\rho(\boldsymbol{y}) E_i)$), and value 0 otherwise.

Consider the case where $y_i = 0$. We have $\mathrm{Tr}[O_i \rho(y)] \leq \alpha_i - \gamma$, which implies that $\widetilde{\alpha}_i \geq p_i + \frac{\gamma}{|\lambda_{\min}| + |\lambda_{\max}|}$. Therefore, the probability that at

least $\widetilde{\alpha}_i \ell$ of the POVMs accept is then:

$$P(\overline{X}_i \geq \widetilde{\alpha}_i) \leq P\left(\overline{X}_i \geq p_i + \frac{\gamma}{|\lambda_{\min}| + |\lambda_{\max}|}\right) \tag{6.32}$$

for $\overline{X}_i = \frac{1}{\ell}\sum_{j=1}^{\ell} X_i^{(j)}$. From the Chernoff bound, we hence get:

$$P(\overline{X}_i \geq \widetilde{\alpha}_i) \leq e^{-2\left(\frac{\gamma}{|\lambda_{\min}|+|\lambda_{\max}|}\right)^2 l}. \tag{6.33}$$

To guarantee an acceptance probability $\mathrm{Tr}(\rho(\boldsymbol{y})^{\otimes \ell} E_i^*) \leq 1/3$, it is then sufficient to take $\ell = \left\lceil \frac{2\log(3)\|O\|_\infty^2}{\gamma^2} \right\rceil$. A similar analysis holds for the case $y_i = 1$.

From here, the result we use that derives from Holevo's bound is Theorem 5.1 of Ambainis *et al.* [138]. It states that in order for the POVMs $\{E_i^*, I - E_i^*\}$ to correctly identify whether $y_i = 0$ or 1 with probability of failure less than $1/3$, we need a number of qubits $n\ell \geq (1 - H(1/3))k$, where H is the binary entropy function. This implies that $n\|O\|_\infty^2/\gamma^2 \geq \frac{2(1-H(1/3))}{\log(3)} k \in \Omega(k)$. □

Generalization bounds of flipped models

The conditions (i) and (ii) of Theorem 6.B.1 are very similar to that of a fat-shattering dimension of a concept class.

Definition 6.B.1
Let $\mathcal{X}$ be a data space, let $\mathcal{C}$ be a class of functions from $\mathcal{X}$ to $\mathbb{R}$, and let $\gamma > 0$. The fat-shattering dimension of the concept class $\mathcal{C}$ at width γ, denoted $\mathrm{fat}_{\mathcal{C}}(\gamma)$, is defined as the size k of the largest set of points $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(k)}\}$ for which there exist real numbers $\alpha_1, \ldots, \alpha_k$ such that for all $\boldsymbol{y} \in \{0,1\}^k$, there exists a $f \in \mathcal{C}$ that satisfies, for all $i \in \{1, \ldots, k\}$,

(i) if $y_i = 0$, then $f(\boldsymbol{x}) \leq \alpha_i - \gamma$, and

(ii) if $y_i = 1$, then $f(\boldsymbol{x}) \geq \alpha_i + \gamma$.

By comparing this definition with the statement of Theorem 6.B.1, we can show:

Corollary 6.B.2
Consider a flipped model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ defined on a data space $\mathcal{X}$, using n-qubit quantum states and observables with spectral norm $\|O\|_\infty =$

$\sup_{\boldsymbol{x}\in\mathcal{X}} \|O(\boldsymbol{x})\|_\infty$. *Call* $\mathcal{C}_{n,O} = \{f_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta}}$ *the concept (or hypothesis) class associated to this model. Then, for all* $\gamma > 0$, *we have* $\mathrm{fat}_{\mathcal{C}_{n,O}}(\gamma) \in \mathcal{O}(n\|O\|_\infty^2/\gamma^2)$.

Proof. We note that since, for all $\boldsymbol{x} \in \mathcal{X}$, $O(\boldsymbol{x})$ lives in a manifold of all observables with spectral norm $\|O\|_\infty$, then the fat-shattering dimension of $\mathcal{C}_{n,O}$ is upper bounded by that of the concept class $\widetilde{\mathcal{C}}_{n,O} = \{O' \mapsto \mathrm{Tr}[\rho(\boldsymbol{\theta})O']\}_{\boldsymbol{\theta}}$ defined on the input space of observables O' that satisfy $\|O'\|_\infty \leq \|O\|_\infty$. Theorem 6.B.1 immediately yields an upper bound for $\mathrm{fat}_{\widetilde{\mathcal{C}}_{n,O}}(\gamma)$, when we identify $\rho(\boldsymbol{\theta})$ in this corollary to $\rho(\boldsymbol{y})$ in the Theorem, and observe that the number of observables $O_1, \ldots, O_k$ that can be γ-shattered (i.e., conditions (i) and (ii) for all labelings) must satisfy $k \in \mathcal{O}(n\|O\|_\infty^2/\gamma^2)$. □

To obtain generalization bounds on the performance of flipped models, we combine this bound on their fat-shattering dimension with standard results from learning theory, e.g.:

Theorem 6.B.3 (Anthony and Barlett [137])
Let $\mathcal{X}$ *be a data space, let* $\mathcal{C}$ *be a class of functions from* $\mathcal{X}$ *to* $\mathbb{R}$, *and let* $\mathcal{D}$ *be a probability measure over* $\mathcal{X}$. *Fix an element* $f \in \mathbb{R}^{\mathcal{X}}$, *as well as error parameters* $\varepsilon, \eta, \gamma, \delta > 0$ *with* $\gamma > \eta$. *Suppose that we draw* m *samples* $X = (\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)})$ *from* $\mathcal{X}$ *according to* $\mathcal{D}$, *and choose any hypothesis* $h \in \mathcal{C}$ *such that* $|h(\boldsymbol{x}) - f(\boldsymbol{x})| \leq \eta$ *for all* $\boldsymbol{x} \in X$. *Then, there exists a positive constant* K *such that, provided*

$$m \geq \frac{K}{\varepsilon}\left(\mathrm{fat}_{\mathcal{C}}\left(\frac{\gamma-\eta}{8}\right)\log^2\left(\frac{\mathrm{fat}_{\mathcal{C}}\left(\frac{\gamma-\eta}{8}\right)}{(\gamma-\eta)\varepsilon}\right) + \log\left(\frac{1}{\delta}\right)\right),$$

with probability $1-\delta$,

$$\Pr_{\boldsymbol{x}\in\mathcal{D}}[|h(\boldsymbol{x}) - f(\boldsymbol{x})| > \gamma] \leq \varepsilon.$$

From Corollary 6.B.2, we have that

$$m \in \widetilde{\Omega}\left(\frac{1}{\varepsilon}\left(\frac{n\|O\|_\infty^2}{(\gamma-\eta)^2} + \log\left(\frac{1}{\delta}\right)\right)\right)$$

samples suffice to get these generalization guarantees. This proves Lemma 6.B.1.

6.B.3 Flipping bounds

In this section, we study mappings between conventional and flipped models (most importantly from conventional to flipped, but our flipping bounds can be used either way). We find that the important quantity that governs the trade-off in resources between these models is the observable trace norm $\|O\|_1 = \mathrm{Tr}\left[\sqrt{O^2}\right]$ of the model to be mapped. Since all observables $O(\boldsymbol{y})$ (where $\boldsymbol{y}$ is either a data vector or a parameters vector, depending on the model) have to be turned into unit-trace density matrices, their eigenvalues need (i) either to be normalized when preserving the number of qubits of the model, or (ii) be encoded in more qubits than in the original model (e.g., by using a binary encoding of all the eigenvalues of $O(\boldsymbol{y})$). Each of these options has its disadvantages: (i) normalizing eigenvalues introduces an overhead in the spectral norm $\|O'\|_\infty$ of the observable of the resulting model, which results in an overhead in the number of measurements needed to evaluate this model to the same precision as the original one. As for (ii), we show that the number of qubits of the new model would need to scale quadratically with $\|O\|_1$ in general, which is commonly an exponential quantity in the number of qubits of the original model (e.g., when O is a Pauli). We show the following lemma:

Lemma 6.B.2 (Flipping bounds (informal))
Any conventional linear model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$ acting on n qubits and with a bounded observable trace norm $\|O\|_1 \leq d$ admits an equivalent flipped model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})]$ acting on $m = n+1$ qubits and with observable spectral norm $\|O'\|_\infty = d$. The bound on the spectral norm is essentially tight in the regime where $n, m \in \mathcal{O}(\log(d))$, i.e., in this case we have $\|O'\|_\infty \geq \widetilde{\Omega}(d)$.

Upper bounds

Theorem 6.B.4
Given a specification of a conventional quantum model $f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$ acting on n qubits, with a known (upper bound on the) trace norm $\|O\|_1$ of its observable, one can construct an equivalent flipped model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})]$ acting on $n+1$ qubits such that $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) = f_{\boldsymbol{\theta}}(\boldsymbol{x})$, $\forall \boldsymbol{x}, \boldsymbol{\theta}$ and $\|O'\|_\infty = \|O\|_1$.

Proof. From Def. 6.A.1, we assume that, in the definition of the conventional model, $\rho(\boldsymbol{x})$ is obtained by applying a unitary $U(\boldsymbol{x})$ on a known quantum state ρ_0 that is diagonal in the computational basis (i.e., is a mixture of computational basis states). As for $O(\boldsymbol{\theta})$, we assume that it is

specified by a unitary $V(\boldsymbol{\theta})$ and a weighted sum of d Hermitian operators O_i, such that $O(\boldsymbol{\theta}) = \sum_{i=1}^{d} w_i V^\dagger(\boldsymbol{\theta}) O_i V(\boldsymbol{\theta})$ and for each $i \in \{1, \dots, d\}$ we know how to decompose O_i as $O_i = \sum_{j=0}^{2^n-1} \lambda_{i,j} W_i^\dagger |j\rangle\langle j| W_i$, for some known $\lambda_{i,j}$'s and W_i's. Note that, as opposed to the parameters that specify $V(\boldsymbol{\theta})$, the weights w_i influence the trace norm $\|O(\boldsymbol{\theta})\|_1$. Therefore we need to pay attention to the fact that $\|O(\boldsymbol{\theta})\|_1$ is only upper bounded by $\|O\|_1 = \sup_{\boldsymbol{\theta}} \|O(\boldsymbol{\theta})\|_1$, and that these two quantities are not always equal.

Out of the observables $O(\boldsymbol{\theta})$, we need to prepare quantum states $\rho'(\boldsymbol{\theta})$ such that $\mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})] = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$. The only difficulty is that quantum states are positive semi-definite and have unit trace while Hermitian observables generally do not fulfill any of these two conditions. To get around these constraints, we simply decompose the observables $O(\boldsymbol{\theta})$ into positive and negative components, that we both normalize. More precisely, call $O_+(\boldsymbol{\theta})$ ($O_-(\boldsymbol{\theta})$) the positive (negative) part of $O(\boldsymbol{\theta}) = O_+(\boldsymbol{\theta}) - O_-(\boldsymbol{\theta})$. We define:

$$\begin{cases} \rho'_+(\boldsymbol{\theta}) = O_+(\boldsymbol{\theta})/\|O_+(\boldsymbol{\theta})\|_1 \\ \rho'_-(\boldsymbol{\theta}) = O_-(\boldsymbol{\theta})/\|O_-(\boldsymbol{\theta})\|_1 \end{cases} \quad \text{and} \quad \begin{cases} p_+ = \|O_+(\boldsymbol{\theta})\|_1/\|O(\boldsymbol{\theta})\|_1 \\ p_- = \|O_-(\boldsymbol{\theta})\|_1/\|O(\boldsymbol{\theta})\|_1 \end{cases} \tag{6.34}$$

such that

$$\rho'(\boldsymbol{\theta}) = p_+ |0\rangle\langle 0| \otimes \rho'_+(\boldsymbol{\theta}) + p_- |1\rangle\langle 1| \otimes \rho'_-(\boldsymbol{\theta}) \tag{6.35}$$

is a valid quantum state (positive semi-definite and unit trace). We can then take

$$O'(\boldsymbol{x}) = \|O(\boldsymbol{\theta})\|_1 (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes \rho(\boldsymbol{x}) \tag{6.36}$$

which, as one can easily verify, leads to $\mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})] = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$. However, this still does not give us a proper flipped model as the renormalization factor $\|O(\boldsymbol{\theta})\|_1$ of $O'(\boldsymbol{x})$ can depend on the parameters $\boldsymbol{\theta}$ (and more precisely the weights w_i, see remark above). We would like to use here the upper bound $\|O\|_1$. To do so, we can simply (re-)define:

$$\begin{cases} p_+ = \|O_+(\boldsymbol{\theta})\|_1/\|O\|_1 \\ p_- = \|O_-(\boldsymbol{\theta})\|_1/\|O\|_1 \end{cases} \quad \text{and} \quad p_0 = \frac{\|O\|_1 - \|O_+(\boldsymbol{\theta})\|_1 - \|O_-(\boldsymbol{\theta})\|_1}{\|O\|_1} \tag{6.37}$$

and also

$$\begin{cases} \rho'(\boldsymbol{\theta}) = p_+ |0\rangle\langle 0| \otimes \rho'_+(\boldsymbol{\theta}) + p_- |1\rangle\langle 1| \otimes \rho'_-(\boldsymbol{\theta}) + p_0 I/2^{n+1} \\ O'(\boldsymbol{x}) = \|O\|_1 (|0\rangle\langle 0| - |1\rangle\langle 1|) \otimes \rho(\boldsymbol{x}) \end{cases} \tag{6.38}$$

such that we still have $\mathrm{Tr}[\rho'(\boldsymbol{\theta})O'(\boldsymbol{x})] = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$ but where we have now defined a proper flipped model. □

Going a bit further, we also propose an algorithm to evaluate the flipped model we constructed in our proof (see Algorithm 1). Given that we only assume to know the eigenvalue decomposition of the single O_i's, and not that of the full observable $O(\boldsymbol{\theta})$, we do not decompose $O(\boldsymbol{\theta})$ directly into its positive and negative components, but rather the O_i's. For this, we define:

$$\begin{cases} O_{i,+} = \sum_{j,\lambda_{i,j}\geq 0} \lambda_{i,j} W_i^\dagger |j\rangle\langle j| W_i \\ O_{i,-} = \sum_{j,\lambda_{i,j}\leq 0} |\lambda_{i,j}| W_i^\dagger |j\rangle\langle j| W_i \end{cases} \tag{6.39}$$

We then recover $\rho'(\boldsymbol{\theta})$ via importance sampling of the indices i, j and $\pm$ and the implementation of the pure state $V(\boldsymbol{\theta})W_i|j\rangle$ (see Algorithm 1)[4]. Naturally, one could alternatively design a full unitary implementation of $\rho'(\boldsymbol{\theta})$ using auxiliary qubits to prepare coherent encodings of the probability distributions appearing in Algorithm 1, along with controlled operations between these auxiliary qubits and the working register, but this would require more qubits and a more complicated quantum implementation.

Lower bounds

Theorem 6.B.5
For d an arbitrary positive integer, there exists a conventional model $\mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})]$, acting on $n \in \mathcal{O}(\log(d))$ qubits, that satisfies $\|O(\boldsymbol{y})\|_1 = d$ for all $\boldsymbol{y} \in \mathcal{Y}$, such that for any flipped model $\mathrm{Tr}[\rho'(\boldsymbol{y})O'(\boldsymbol{x})]$ acting on m qubits with $\|O'\|_\infty = \sup_{\boldsymbol{x}\in\mathcal{X}} \|O'(\boldsymbol{x})\|_\infty$, and any $\varepsilon \geq 0$, if the model satisfies

$$|\mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})] - \mathrm{Tr}[\rho'(\boldsymbol{y})O'(\boldsymbol{x})]| \leq \varepsilon \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \times \mathcal{Y} \tag{6.40}$$

then, it must also satisfy

$$m\|O'\|_\infty^2 \in \Omega(d^2(1/2-\varepsilon)^2).$$

Proof. The core of the proof is to show that a conventional model $\mathrm{Tr}[\rho(i)O(\boldsymbol{y})]$ with trace norm $\|O(\boldsymbol{y})\|_1 = d$ and acting on $n = \lceil \log_2(N+1) \rceil$ qubits, for $N = \lfloor d^2/4 \rfloor$, can represent the function $i \mapsto y_i$ for $1 \leq i \leq N$, for all

[4] In Algorithm 1, $\sigma(\boldsymbol{\theta})$ corresponds to either $\rho'_+(\boldsymbol{\theta})$ or $\rho'_-(\boldsymbol{\theta})$, depending on the sampled $b \in \{+,-\}$.

$\boldsymbol{y} \in \{0,1\}^N$. For this, we take, for all $1 \leq i \leq N$:

$$\rho(i) = \frac{1}{2}(|0\rangle + |i\rangle)(\langle 0| + \langle i|) \quad \text{and} \quad O(\boldsymbol{y}) = \sum_{i'=1}^{N+1} O_{i'}(\boldsymbol{y}) \tag{6.41}$$

for

$$O_i(\boldsymbol{y}) = \begin{cases} y_i(|0\rangle\langle i| + |i\rangle\langle 0|) & \text{if } 1 \leq i \leq N \\ (d - \sqrt{|\boldsymbol{y}|})\,|N+1\rangle\langle N+1| & \text{if } i = N+1 \end{cases} \tag{6.42}$$

where $|\boldsymbol{y}| = \sum_{i=1}^N y_i$ is the Hamming weight of $\boldsymbol{y}$. By construction, we have that the upper-left $N \times N$ block of $O(\boldsymbol{y})$ satisfies $\left\|O_{(N\times N)}(\boldsymbol{y})\right\|_1 = \sqrt{|\boldsymbol{y}|} \leq d$ (it corresponds to the adjacency matrix of a star graph of degree $D = |\boldsymbol{y}| \leq N$, which has trace norm $2\sqrt{D}$ [139]), such that $\|O(\boldsymbol{y})\|_1 = d$ for all $\boldsymbol{y} \in \{-1,1\}^N$. Also, it is easy to check that $\mathrm{Tr}[\rho(i)O(\boldsymbol{y})] = y_i$ for all $1 \leq i \leq N$.
We take this conventional model to be our target model. Satisfying the condition of Eq. (6.40) is then equivalent to satisfying:

$$|\mathrm{Tr}[\rho'(\boldsymbol{y})O'(i)] - y_i| \leq \varepsilon \quad \forall\, i, \boldsymbol{y} \in \{1,\ldots,N\} \times \{0,1\}^N.$$

Now note that this condition is stronger than that of Theorem 6.B.1 for $\gamma = 1/2 - \varepsilon$ and $\alpha_{i,j} = 1/2\ \forall i,j$. Therefore, in order not to contradict with this theorem, we must have $m\|O'\|_\infty^2 \in \Omega(N(1/2-\varepsilon)^2) = \Omega(d^2(1/2-\varepsilon)^2)$. □

Lemma 6.B.2 is obtained from Theorem 6.B.4 as stated above and Theorem 6.B.5 for the case $m \in \mathcal{O}(\log d)$, such that the statement $m\|O'\|_\infty^2 \in \Omega(d^2(1/2-\varepsilon)^2)$ becomes $\|O'\|_\infty \in \Omega\left(\frac{d}{\sqrt{\log(d)}}(1/2-\varepsilon)\right)$.

Circumventing lower bounds

Note that our flipping bounds are essentially tight only in the regime where the number of qubits used by the original and resulting model n, m are both in $\mathcal{O}(\mathrm{polylog}(\|O\|_1))$. This is a relevant regime, as it includes notably the case of Pauli observables (be they local or non-local) or linear combinations thereof. However, outside this regime our bounds can be circumvented.

Also note that an easy way of circumventing our lower bounds, even in the regime where $n, m \in \mathcal{O}(\mathrm{polylog}(\|O\|_1))$, is by imposing the constraint

that $O(\boldsymbol{y})$ is parametrized by $|\boldsymbol{y}| = \mathcal{O}(\text{poly}(n))$ parameters[5]. Indeed, in this case, one can simply use a similar construction to that in Ref. [114] (Fig. 3) where one would encode $\boldsymbol{y}$ (e.g., in binary form) in auxiliary qubits as $|\widetilde{\boldsymbol{y}}\rangle$ and use controlled operations that are independent of $\boldsymbol{y}$ to simulate the action of gates parametrized by $\boldsymbol{y}$. One can then note that by taking $\rho'(\boldsymbol{\theta}) = \rho_0 \otimes |\widetilde{\boldsymbol{y}}\rangle\langle\widetilde{\boldsymbol{y}}|$ and the rest of the resulting circuit to define $O'(\boldsymbol{x})$, one ends up with a flipped model that acts on $\mathcal{O}(\text{poly}(n))$ qubits and satisfies $\|O'\|_\infty = \|O\|_\infty$. For O defined by a Pauli observable for instance, we have $\|O\|_\infty = 1$ and $\|O\|_1 = 2^n$. Such a construction therefore does not suffer from an exploding spectral norm $\|O'\|_\infty$. However, it also does not lead to shadowfiable models in general as the parametrized states $\rho'(\boldsymbol{\theta})$ play a trivial role and the observables $O'(\boldsymbol{x})$ hide all of the quantum computation.

6.C Quantum advantage using shadow models

6.C.1 Discrete cube root learning task

In this section we rigorously define the discrete cube root learning task introduced in the main text and detail the proof of its classical hardness. More precisely, we start by introducing the discrete cube root problem and state formally its classical hardness assumption (the discrete cube root assumption). Then we construct a learning task that is classically hard based on this assumption.

The discrete cube root problem

A definition of the discrete cube root problem can be found in Ref. [125]. For convenience, we restate it in this appendix.[6] Consider two large prime numbers p and q of the form $3k+2$, $3k'+2$, for distinct k, k', and which can be represented by approximately the same number of bits. Let $N = pq$ be the product of these primes, which we assume to be an n-bit integer, and let $\mathbb{Z}_N = \{0, \ldots, N-1\}$.

[5] In our proof of Theorem 6.B.5, we use $|\boldsymbol{y}| \in \Omega(\|O\|_1^2)$, which is in $\Omega(\exp(n))$ for $n, m \in \mathcal{O}(\log(\|O\|_1))$, and subexponential in n for $n, m \in \mathcal{O}(\text{polylog}(\|O\|_1))$

[6] Note that, as opposed to the exposition of Ref. [125], we consider the domain $\mathbb{Z}_N = \{0, \ldots, N-1\}$ instead of $\{i \mid 0 < i < N,\ \gcd(i, N) = 1\}$ for the functions we define next. This allows us to apply more easily the result of Ref. [127] and construct a learning task with a stronger form of classical hardness.

We consider the "discrete cube" function $f_N(y) : \mathbb{Z}_N \to \mathbb{Z}_N$ defined as $f_N(y) = y^3 \bmod N$, as well as its inverse, which we denote as $g_N(x) = f_N^{-1}(x) = \sqrt[3]{x} \bmod N$ (see Fig. 6.3). As we explain in the following, f_N is particularly interesting because it is believed to be a *one-way function*: $f_N(y)$ can be computed efficiently classically using modular exponentiation, while $g_N(x)$ is believed hard to compute classically, with only knowledge of N and x (and not the factors p, q of N). But first, let us show that the inverse function g_N is properly defined.

Lemma 6.C.1
For p, q two distinct prime numbers of the form $3k+2$, $3k'+2$ and $N = pq$, the function $f_N(y) = y^3 \mod N$ is a bijection on $\mathbb{Z}_N$.

Proof. To show that f_N is a bijection on $\mathbb{Z}_N$, it is sufficient to show that it is injective, since it maps $\mathbb{Z}_N$ to itself.
Consider $y, z \in \mathbb{Z}_N$ such that $f_N(y) = f_N(z)$, i.e., $y^3 \bmod N = z^3 \bmod N$. We then have $y^3 - z^3 \equiv 0 \bmod N$, which means that there exists an $l \in \mathbb{N}$ such that $y^3 - z^3 = lN = lpq$. Therefore, we know that $y^3 - z^3$ is divisible by both p and q, which implies $y^3 \equiv z^3 \bmod p$ and $y^3 \equiv z^3 \bmod q$. From here, we apply a similar reasoning for p and q, that we detail only for p.
Given that we assumed $p = 3k + 2$, we know that $\gcd(p - 1, 3) = 1$. Therefore, Euclid's algorithm assures that there exist $d_p, d'_p \geq 1$ such that $3d_p = (p - 1)d'_p + 1$. Then notice that:

$$y^{3d_p} \equiv y^{(p-1)d'_p+1} \equiv y \bmod p \tag{6.43}$$

where the last congruence follows from applying Fermat's little theorem ($y^p \equiv y \bmod p$ for all $y \in \mathbb{N}$) to show by induction on $m \geq 0$ that $y^{(p-1)m+1} \equiv y \bmod p$. Similarly, $z^{3d_p} \equiv z \bmod p$, and therefore by raising to the power d_p both terms in $y^3 \equiv z^3 \bmod p$, we get $y \equiv z \bmod p$.
From the same reasoning, we get $y \equiv z \bmod q$, which implies that $x - y$ is divisible by both p and q. Since they are distinct primes, then $x - y$ is also divisible by $pq = N$, which means that $x \equiv y \bmod N$. This shows that f_N is injective, and therefore bijective. □

Now that we have shown that the discrete cube root function is properly defined on $\mathbb{Z}_N$, let us define the discrete cube root problem:

Definition 6.C.1 (Discrete cube root problem)
Let p and q be two distinct primes of the form $3k + 2, 3k' + 2$ represented by approximately the same number of bits, and such that $N = pq$ is an n-bit integer. Given as input both N and $x \in \mathbb{Z}_N$, output $y \in \mathbb{Z}_N$ such that $y^3 = x \bmod N$.

The assumption that the "discrete cube" function f_N is a one-way function is formalized by the so-called discrete cube root assumption, which is a special case of the RSA assumption for the exponent $e = 3$.

Definition 6.C.2 (Discrete cube root assumption [125])
For any polynomial $P(.)$, there does not exist a classical algorithm $\mathcal{A}$, that runs in time $P(n)$ and that, on input N and x (where N is the n-bit product of two random primes of the form $3k+2$ and x is chosen randomly from $\mathbb{Z}_N$), outputs $y \in \mathbb{Z}_N$ such that with probability $1/P(n)$ satisfies $y^3 = x \mod N$. The probability of success is taken over the random draws of the two primes p, q and the input $x \in \mathbb{Z}_N$ and any internal randomisation of $\mathcal{A}$.

While other one-way functions like the discrete exponential (along with its inverse, the discrete logarithm) also have similar classical hardness assumptions, the discrete cube (root) function is additionally known to be a *trap-door function*. That is, there exists a key $d \in \mathbb{Z}_N$ such that $g_N(x)$ can be alternatively computed via modular exponentiation as $g_N(x) = x^d \mod N$. This key d can be efficiently computed from the prime factors p and q of N. We show this using a similar reasoning to that around Equation 6.43. Call $\phi(N) = (p-1)(q-1)$. From $p = 3k+2$ and $q = 3k'+2$, we have $\gcd(\phi(N), 3) = 1$. Therefore, Euclid's algorithm assures that there exists $d, d' \geq 1$ such that $3d = \phi(N)d' + 1 = (p-1)(q-1)d' + 1$. We want to show that, for all $y \in \mathbb{Z}_N$,

$$y^{3d} \equiv y \mod N. \tag{6.44}$$

To do this, we show that $y^{3d} - y$ is divisible by both p and q. In the case of p, we have:

$$y^{3d} \equiv y^{(p-1)(q-1)d'+1} \equiv y \mod p. \tag{6.45}$$

The last equality follows again from Fermat's little theorem (see Equation 6.43). Similarly, $y^{3d} \equiv y \mod q$, which implies that $y^{3d} - y$ is divisible by p and q, and that $y^{3d} \equiv y \mod N$. Therefore d is a valid key for computing $g_N(y)$ for all $y \in \mathbb{Z}_N$. When knowing the factors p and q of N, one can compute $\phi(N)$ and use Euclid's algorithm to find d such that $3d = 1 \mod \phi(N)$. However, factoring a large N is believed to be computationally intractable classically, which justifies the discrete cube root assumption (Def. 6.C.2).

The learning task

Based on the discrete cube root assumption, we can construct a learning task that is not efficiently PAC learnable classically. In order to define the concept class of this learning task, we first consider the class of functions:

$$\mathcal{F}_n = \{g_N : \mathbb{Z}_N \to \mathbb{Z}_N \mid g_N(x) = \sqrt[3]{x} \bmod N, \\ \text{for } N \text{ an } n\text{-bit integer that satisfies the DCR conditions}\}$$

defined for any integer n. We define the concept class:

$$\mathcal{C}_n = \{g_{N,s} : \mathbb{Z}_N \to \{0,1\} \mid g_{N,s}(x) = \begin{cases} 1, & \text{if } g_N(x) \in [s, s + \frac{N-1}{2}], \\ 0, & \text{otherwise.} \end{cases} \tag{6.46}$$

$$\text{, for } g_N \in \mathcal{F}_n,\ s \in \mathbb{Z}_N\} \tag{6.47}$$

for any integer n. In our proofs of classical hardness and quantum learnability, we also need that the integer N corresponding to the target function should be specified with the training data. One way of doing so is to append it to all inputs $x \in \mathbb{Z}_N$ as (x, N) and redefine the concept class accordingly. To ease notation, we assume this transformation to be done implicitly in the following.

Because the discrete cube function is a one-way function and is bijective on $\mathbb{Z}_N$, it is easy to classically generate training data for any concept $g_{N,s} \in \mathcal{C}_n$, under the uniform distribution over $\mathbb{Z}_N$. Indeed, one can simply uniformly sample $y \in \mathbb{Z}_N$, compute its corresponding $x = y^3 \bmod N$ via modular exponentiation and keep from y only the label indicating whether $y \in [s, s + \frac{N-1}{2}]$. The bijectivity of f_N ensures that x is generated uniformly over $\mathbb{Z}_N$.

In the PAC setting, a learning algorithm has to find, for every concept $g_{N,s} \in \mathcal{C}_n$, for every data distribution $\mathcal{D}$, and for all $\varepsilon, \delta \in (0, 1/2)$, a hypothesis function h that, with probability $1-\delta$, satisfies $\Pr_{x\sim\mathcal{D}}[g_{N,s}(x) \neq h(x)] \leq \varepsilon$ in time and number of samples both polynomial in n, $1/\varepsilon$ and $1/\delta$.

Note that in Ref. [125], the authors consider instead the concept class

$$\mathcal{C}'_n = \{g_{N,i} : \mathbb{Z}_N \to \{0,1\} \mid g_{N,i}(x) = \mathrm{bin}(i, g_N(x)),\ \text{for } g_N \in \mathcal{F}_n,\ i \in [n]\} \tag{6.48}$$

for any integer n, where $\mathrm{bin}(i, y)$ denotes the i-th bit of y in binary form. This concept class is also not efficiently PAC learnable, although this is only shown in a much weaker sense. The authors show that no classical

algorithm can achieve an error $\varepsilon = 1/n^2$ with failure probability $\delta = 1/n^2$ in $\mathcal{O}(\text{poly}(n))$ time, while, for the concept class $\mathcal{C}_n$ we consider, we can show that $\varepsilon = 1/2 - 1/\text{poly}(n)$ and $\delta = 1/3$ would already break the DCR assumption.

Classical hardness

To show that classical learners cannot achieve significantly better than random guesses on the class C_n, we make use of a result from Alexi *et al.*[127]. This result makes use of the notion of a $\varepsilon(n)$-oracle:

Definition 6.C.3 ($\varepsilon(n)$-oracle)
Let $O_{N,s}$ be a probabilistic oracle, such that $O_{N,s}(x)$ computes $g_{N,s}(x)$ correctly with probability $1/2 + \varepsilon(n)$ over the random choice of x and the internal randomness of the oracle. We say that $O_{N,s}$ is an $\varepsilon(n)$-oracle.

Alexi *et al.* show that a $1/\text{poly}(n)$-oracle is sufficient to break the DCR assumption.

Lemma 6.C.2 (Corollary (a) to Theorem 1 in [127])
For any $g_{N,s} \in \mathcal{C}_n$, given a $1/poly(n)$-oracle $O_{N,s}$ to $g_{N,s}$, there exists a $\mathcal{O}(poly(n))$-time algorithm that uses $O_{N,s}$ to compute $g_N(x)$, $\forall x \in \mathbb{Z}_N$ (with success probability, e.g., $9/10$).

We make use of this result to show the following lemma:

Lemma 6.C.3
Under the discrete cube root assumption, no $\mathcal{O}(poly(n))$-time classical learning algorithm can achieve an expected error

$$\Pr_{x \sim \mathcal{U}(\mathbb{Z}_N)}[h(x) \neq g_{N,s}(x)] \leq 1/2 - 1/poly(n) \tag{6.49}$$

with probability $2/3$ over the random generation of its training data and its internal randomness, and this for every concept $g_{N,s} \in \mathcal{C}_n$. $\mathcal{U}(\mathbb{Z}_N)$ is the uniform distribution over $\mathbb{Z}_N$.

Proof. Suppose by contradiction that such a learning algorithm would exist for a certain concept $g_{N,s} \in \mathcal{C}_n$. Then, given N, one can use this learning algorithm to generate with probability $2/3$ a $1/\text{poly}(n)$-oracle $O_{N,s} = h$. This is possible since the generation of training data for the concept $g_{N,s}$ is classically efficient given N. Now, by applying Lemma 6.C.2, one obtains a $\mathcal{O}(\text{poly}(n))$-time algorithm that uses this $O_{N,s}$ to compute $g_N(x)$, $\forall x \in \mathbb{Z}_N$, with success probability $9/10$. The overall success probability of this procedure, taken over the random choice of x, N and the learning algorithm is 0.6, which contradicts the DCR assumption. □

6.C.2 A simple shadow model

In this section we show how to construct a simple shadow model which can solve the same learning task for which we just showed classical hardness. This shadow model is obtained from the following flipped model:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$$
$$\rho(\boldsymbol{\theta}) = |d', s'\rangle\langle d', s'| \;\; \& \;\; O(\boldsymbol{x}) = \sum_{d', s'} g_{N,s'}(\boldsymbol{x}) \, |d', s'\rangle\langle d', s'| \, . \tag{6.50}$$

That is, $\rho(\boldsymbol{\theta})$ consists of an n-qubit register which contains the candidate key $d' \in \mathbb{Z}_N$ and a second n-qubit register containing the candidate separating $s' \in \mathbb{Z}_N$ that is used to label whether $g_N(\boldsymbol{x}) = \boldsymbol{x}^d \bmod N \in [s', s' + \frac{N-1}{2}]$.

Lemma 6.C.4
The concept class $\mathcal{C}_n$ is efficiently PAC learnable under the uniform distribution $\mathcal{U}(\mathbb{Z}_N)$ using a shadow model.

Proof. We first describe how $\rho(\boldsymbol{\theta})$ can be computed efficiently quantumly. Using the specification of N provided by the training data[7], one can use Shor's algorithm to compute the factors p, q of N with arbitrarily high probability of success. This in turn allows to compute $\phi(N) = (p-1)(q-1)$ and the key $d' = d$ that satisfies $3d = 1 \bmod \phi(N)$ using Euclid's algorithm. As for the candidate separating s', it can be encoded using Pauli-X gates. The observable $O(\boldsymbol{x})$ can also be evaluated efficiently from computational basis measurements. For an outcome (d', s'), one simply computes $g_N(\boldsymbol{x}) = \boldsymbol{x}^d \bmod N$ via modular exponentiation and checks whether the output is in $[s', s' + \frac{N-1}{2}]$.
This model is naturally specified as a shadow model. One preparation of $\rho(\boldsymbol{\theta})$ followed by a computational basis measurement results in the classical advice $\omega(\boldsymbol{\theta}) = (d', s')$. The classical evaluation $\mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta}))$ of a new data point $\boldsymbol{x}$ is done as explained in the last paragraph.
The only remaining learning aspect is to identify an s' close to s from a training set $\{(x, y_i = g_{N,s}(x))\}_{x \sim \mathcal{U}(\mathbb{Z}_N)}$. We show that a training set X of size $|X| \geq \frac{\log(\delta)}{\log(1-2\varepsilon)}$ is guaranteed to contain an x^* such that, for

[7] Strictly speaking, the PAC framework does not allow one to provide N explicitly in the training data. One way around this issue is to redefine the concepts to be of the form $\tilde{g}_{N,s} : \{0,1\} \times \mathbb{Z}_N \to \{0,1\} \mid \tilde{g}_{N,s}(0, x) = g_{N,s}(x)$ and $\tilde{g}_{N,s}(1, x) =$ "j-th bit of N, for j encoded in the first $\log(N)$ bits of x". This way, we can efficiently recover N from the uniform distribution $\mathcal{U}(\{0,1\} \times \mathbb{Z}_N)$, while only marginally impacting the training data.

$s' = g_N(x^*)$, $|s - s'| \leq \varepsilon N$ with probability $1 - \delta$. We take this x^* to be $x^* = \text{argmin}_{x \in X} \mathcal{L}(x^d \bmod N)$, for $\mathcal{L}(y) = \sum_{x \in X} |g_{N,y}(x) - g_{N,s}(x)|$ the training loss on the training set X. We show this by proving:

$$\Pr\left(|s' - s| \geq \varepsilon N\right) \leq \delta. \tag{6.51}$$

This probability is precisely the probability that no $g_N(x) \in \{g_N(x)\}_{x \in X}$ is within ε distance of s, i.e.,

$$\Pr\left(\bigcap_{x \in X} g_N(x) \notin [s - \varepsilon N, s + \varepsilon N]\right). \tag{6.52}$$

As the elements of the training set are all identically distributed, we have that this probability is equal to

$$\Pr\left(g_N(x) \notin [s - \varepsilon N, s + \varepsilon N]\right)^{|X|}. \tag{6.53}$$

Since all the datapoints are uniformly sampled from $\mathbb{Z}_N$, the probability that a datapoint is in any region of size $2\varepsilon N$ is just 2ε. With the assumption that $|X| \geq \log_{1-2\varepsilon}(\delta)$ (and assuming $\varepsilon < 1/2$), we get:

$$\Pr\left(|s' - s| \geq \varepsilon N\right) \leq (1 - 2\varepsilon)^{\log_{1-2\varepsilon}(\delta/2)} = \delta. \tag{6.54}$$

From here, we simply notice that $|s' - s| \leq \varepsilon N$ guarantees an expected error

$$\Pr_{x \sim \mathcal{U}(\mathbb{Z}_N)}[g_{N,s'}(x) \neq g_{N,s}(x)] \leq 2\varepsilon. \tag{6.55}$$

□

6.D Relations between shadow models

6.D.1 Flipped models are universal

In this section we show that any shadowfiable model as defined in Defs. 6.3.1 and 6.3.2 can be approximated by an efficiently shadowfiable flipped model. This result corresponds to Lemma 1 in the main text, and more formally in the following lemma.

Lemma 6.D.1
Let $f_{\boldsymbol{\theta}}$ be a shadowfiable model acting on n qubits as defined in Def. 6.3.2 and let $\varepsilon, \delta > 0$. There exists a flipped model $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \text{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$ acting

on $\mathcal{O}(poly(n))$ qubits and evaluatable in $\mathcal{O}(poly(n, 1/\varepsilon))$ time such that

$$\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\boldsymbol{\theta}}(\boldsymbol{x}) - g_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq \varepsilon. \tag{6.56}$$

Moreover, this flipped model is also shadowfiable for the error parameter ε and the success probability $1-\delta$. More precisely, a computational basis measurement of $\rho(\boldsymbol{\theta})$ yields an advice $\omega(\boldsymbol{\theta})$ such that with probability $1-\delta$ over the randomness of this measurement, we have

$$\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{g}_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq \varepsilon, \tag{6.57}$$

where $\widetilde{g}_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta}))$ is a classical $\mathcal{O}(poly(n, 1/\varepsilon, 1/\delta, d))$-time algorithm that processes the advice $\omega(\boldsymbol{\theta})$ along with an input $\boldsymbol{x} \in \mathbb{R}^d$.

Proof. By Def. 6.3.2, the shadowfiable model $f_{\boldsymbol{\theta}}$ admits for every error of approximation $\varepsilon' > 0$ and probability of failure $\delta' > 0$ a shadow model $\widetilde{f}_{\boldsymbol{\theta}}$ that uses $m \cdot M \in \mathcal{O}(\text{poly}(n, 1/\varepsilon', 1/\delta'))$ qubits and guarantees

$$\max_{\boldsymbol{x}\in\mathcal{X}} \left| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right| \leq \varepsilon' \tag{6.58}$$

with probability $1-\delta'$ over the generation of its advice. Out of this shadow model, we use the construction described in Fig. 6.4.b) to define the flipped model $g_{\boldsymbol{\theta}}(\boldsymbol{x})$. Since this flipped model corresponds to the evaluation of the shadow model $\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x})$ averaged over the randomness of $\omega(\boldsymbol{\theta})$, we have, for all $\boldsymbol{x} \in \mathbb{R}^d$:

$$|f_{\boldsymbol{\theta}}(\boldsymbol{x}) - g_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq \left| (1-\delta')\varepsilon' + \delta' \left\| \widetilde{f}_{\boldsymbol{\theta}} \right\| \right| \tag{6.59}$$

$$\leq \varepsilon' + \delta' \left(\left\| \widetilde{f}_{\boldsymbol{\theta}} \right\| + \varepsilon' \right) \tag{6.60}$$

where we use that with probability $1-\delta'$ we have $\left| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right| \leq \varepsilon'$ and otherwise assume the worse case error $\left| f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right| \leq 2 \left\| \widetilde{f}_{\boldsymbol{\theta}} \right\|$, for $\left\| \widetilde{f}_{\boldsymbol{\theta}} \right\| = \max_{\boldsymbol{x}\in\mathcal{X}} \left| \widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) \right|$ (which can also be capped to $\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\boldsymbol{\theta}}(\boldsymbol{x})|$ without loss of generality). Therefore, by setting $\varepsilon' = \frac{\varepsilon}{2}$ and $\delta' = \min\left\{ \frac{\varepsilon}{2\left(\left\| \widetilde{f}_{\boldsymbol{\theta}} \right\| + \varepsilon'\right)}, \delta \right\}$ (important for the second part of the proof), we get

$$\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\boldsymbol{\theta}}(\boldsymbol{x}) - g_{\boldsymbol{\theta}}(\boldsymbol{x})| \leq \varepsilon. \tag{6.61}$$

This proves the first part of the lemma. From here, it is straightforward to notice that a measurement of $\rho(\boldsymbol{\theta}) = |\omega(\boldsymbol{\theta})\rangle\langle\omega(\boldsymbol{\theta})| \otimes |0\rangle\langle 0|^{\otimes a}$ in the computational basis yields an advice $\omega(\boldsymbol{\theta})$ such that the algorithm $\mathcal{A}$ associated to the shadow model $\widetilde{f}_{\boldsymbol{\theta}}$ satisfies

$$\max_{\boldsymbol{x}\in\mathcal{X}} |f_{\boldsymbol{\theta}}(\boldsymbol{x}) - \mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta}))| \leq \varepsilon' < \varepsilon \tag{6.62}$$

with probability at least $1 - \delta' \geq 1 - \delta$ over the randomness of measuring the advice. □

6.D.2 BQP and P/poly

In this section we give a rigorous proof that there exist quantum models that are not shadowfiable, under the assumption that BQP $\not\subset$ P/poly. The approach we take is similar to that of Huang *et al.* [131] (Appendix A), although we consider different complexity classes and therefore show different results. Let us start by noting that the shadow models defined in Def. 6.3.1 compute functions in a subclass of P/poly, namely the subclass in which the advice $\omega(\boldsymbol{\theta})$ is efficiently generated from the measurements of a polynomial number of quantum circuits. We call this complexity class BPP/qgenpoly and it is obvious that BPP/qgenpoly $\subseteq$ P/poly as the latter is equal to BPP/poly [130] and contains all classically efficiently computable functions with advice of polynomial length, without any constraints on how the advice is generated.

On the other hand, we know that quantum models can compute all functions in BQP. To see this, we first refer the reader to the definition of BQP in Def. 6.A.4. It is easy to show that for any language L in BQP, there exists a quantum model which can decide this language. Consider the following quantum model $f_n = \mathrm{Tr}[\rho(x)O_n]$, depicted in Fig. 6.5:

$$\rho(x) = \bigotimes_{i=1}^{n} X_i^{x_i} |0\rangle\langle 0| X_i^{x_i} \quad \& \quad O_n = U_n^\dagger Z_1 U_n \tag{6.63}$$

Where X_i is the Pauli-X gate acting on the i-th qubit, here parametrized by $x_i \in \{0, 1\}$, the i-th bit of x, Z_1 is the Pauli observable on the first qubit, and U_n is the quantum circuit used to decide the language L in Def. 6.A.4. Then:

1. For all $x \in L$, $f_n(x) = \Pr[\text{the output of } U_{|x|} \text{ applied on the input } x \text{ is } 1]$ - $\Pr[\text{the output of } U_{|x|} \text{ applied to the input } x \text{ is } 0] \geq 2/3 - 1/3 = 1/3$.

2. For all $x \notin L$, $f_n(x) = \Pr[\text{the output of } U_{|x|} \text{ applied on the input } x \text{ is } 1] - \Pr[\text{the output of } U_{|x|} \text{ applied to the input } x \text{ is } 0] \leq 1/3 - 2/3 = -1/3$.

Therefore, as $f_n(x) > 0$ if $x \in L$ and $f_n(x) < 0$ if $x \notin L$ such quantum model could efficiently decide the language. We show that if all such quantum models would be shadowfiable then $\mathsf{BQP} \subseteq \mathsf{P/poly}$.

Lemma 6.D.2
If all quantum models $f_{\boldsymbol{\theta}}$ are shadowfiable with the guarantee that, $\forall x \in \mathcal{X}$,

$$\left| f_{\boldsymbol{\theta}}(x) - \widetilde{f}_{\boldsymbol{\theta}}(x) \right| < 0.15, \tag{6.64}$$

with probability at least $2/3$ over the shadowing phase and the randomness of evaluating the shadow model $\widetilde{f}_{\boldsymbol{\theta}}$, then $\mathsf{BQP} \subseteq \mathsf{P/poly}$.

Proof. Consider a language L in BQP. We showed that there exists a quantum model $f_{\boldsymbol{\theta}} = f_n$ which can determine, for every $x \in \{0,1\}^n$, whether x is in L. Now, by our assumption, there exists a shadow model $\widetilde{f}_n$ such that, for every $x \in \{0,1\}^n$, $|f_n(x) - \widetilde{f}_n(x)| < 0.15$ with probability greater than $2/3$ over the randomness of sampling the advice $\omega(\boldsymbol{\theta})$ and the (potential) internal randomness of its classical algorithm $\mathcal{A}$. To get rid of these two sources of randomness, we make use of the same proof strategy as for Adleman's theorem [130]: Consider now a new algorithm $\mathcal{A}'$ which runs $\mathcal{A}$ $18n$ times, each time using a new sampled advice string $\omega(\boldsymbol{\theta})$ and a new random bit-string for its internal randomness. Then we take a majority vote from the $18n$ runs. By Chernoff bound, the probability that for any $x \in \{0,1\}^n$ the algorithm $\mathcal{A}'$ fails to determine if x belongs to L is at most $1/e^n$. Then, by union bound, the probability that $\mathcal{A}'$ decides all the $x \in \{0,1\}^n$ correctly is at least $1 - 2^n/e^n > 0$. This implies that there exists a particular choice of the $18n$ strings $\omega(\boldsymbol{\theta})$ and $18n$ random bit-strings used in each run of the algorithm $\mathcal{A}$ such that $\mathcal{A}'$ is correct for all x. The algorithm $\mathcal{A}'$, along with this particular choice of $18n$ strings as advice (which is of size polynomial in n) is our $\mathsf{P/poly}$ algorithm. Note that it guarantees $\forall \boldsymbol{x} \in \{0,1\}^n$, $\widetilde{f}_n(x) > 0$ if $x \in L$ and $\widetilde{f}_n(x) < 0$ if $x \notin L$. Therefore we can use the sign of $\widetilde{f}_n(x)$ to determine whether $x \in L$. This implies $\mathsf{BQP} \subseteq \mathsf{P/poly}$. □

6.D.3 Shadow models beyond Fourier

Generalization to the full domain $\boldsymbol{x} \in \mathbb{R}^n$

In Section 6.3.1, we showed how the model:

$$\begin{aligned} f_{\boldsymbol{y}}(\boldsymbol{x}) &= \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})] \\ \rho(\boldsymbol{x}) = \bigotimes_{i=1}^{n} R_Y(x_i)\,|0\rangle\langle 0|\,R_Y^\dagger(x_i) \;\&\; O(\boldsymbol{y}) &= |\boldsymbol{y}\rangle\langle\boldsymbol{y}|\,. \end{aligned} \tag{6.65}$$

for $\boldsymbol{y} \in \{0,1\}^{\otimes n}$ is not efficiently shadowfiable when we restrict the domain of $\boldsymbol{x}$ to be $\{0,\pi\}^n$ as, on this domain, $f_{\boldsymbol{y}}(\boldsymbol{x}) = \delta_{\boldsymbol{x}/\pi,\boldsymbol{y}}$ plays the role of a database search oracle. We can extend this intractability result to the full domain $\boldsymbol{x} \in \mathbb{R}^n$ by that noting that more general encoding states (even more general than those in Equation 6.65) take the form

$$\rho(\boldsymbol{x}) = |\psi(\boldsymbol{x})\rangle\langle\psi(\boldsymbol{x})|\,, \quad \text{for} \quad |\psi(\boldsymbol{x})\rangle = \sum_{\boldsymbol{j}=1}^{2^n} \alpha_{\boldsymbol{j}}(\boldsymbol{x})\,|\boldsymbol{j}\rangle \tag{6.66}$$

where $\alpha_{\boldsymbol{j}}(\boldsymbol{x}) \in \mathbb{C}$ is an arbitrary amplitude associated to a computational basis state $|\boldsymbol{j}\rangle$.

Now note that when we evaluate the quantum model $f_{\boldsymbol{y}}(\boldsymbol{x})$ on a quantum computer, we do not have direct access to the expectation values $\mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})]$ it corresponds to. We rather sample an eigenvalue of $O(\boldsymbol{y})$ according to the Born rule applied to $\rho(\boldsymbol{x})$. Therefore, a single evaluation of $f_{\boldsymbol{y}}(\boldsymbol{x})$ for the encoding $\rho(\boldsymbol{x})$ defined in Equation 6.66 returns 1 with probability $|\alpha_{\boldsymbol{y}}(\boldsymbol{x})|^2$ and 0 otherwise.

Let us call $U_{\boldsymbol{y}}$ the Grover operator associated to the database search oracle that marks $\boldsymbol{y}$, i.e.,

$$U_{\boldsymbol{y}} : |\boldsymbol{j}\rangle\,|0\rangle \mapsto |\boldsymbol{j}\rangle\,|\delta_{\boldsymbol{j},\boldsymbol{y}}\rangle\,. \tag{6.67}$$

and apply it on the state $|\psi(\boldsymbol{x})\rangle\,|0\rangle$:

$$U_{\boldsymbol{y}}\,|\psi(\boldsymbol{x})\rangle\,|0\rangle = \sum_{\boldsymbol{j}=1}^{2^n} \alpha_{\boldsymbol{j}}(\boldsymbol{x})\,|\boldsymbol{j}\rangle\,|\delta_{\boldsymbol{j},\boldsymbol{y}}\rangle\,. \tag{6.68}$$

One can then notice that measuring the second register also yields 1 with probability $|\alpha_{\boldsymbol{j}}(\boldsymbol{x})|^2$ and 0 otherwise. Therefore, a single evaluation of $f_{\boldsymbol{y}}(\boldsymbol{x})$ is as powerful as querying the Grover operator oracle in superposition,

which still suffers from the same query complexity lower bound as querying it classically.

Fourier decomposition

In this subsection, we derive the Fourier-series decomposition of the model in Eq. (6.65), i.e., the frequency spectrum Ω and the Fourier coefficients $c_{\boldsymbol{\omega}}$ in the expression:

$$f_{\boldsymbol{y}}(\boldsymbol{x}) = \sum_{\boldsymbol{\omega}\in\Omega} c_{\boldsymbol{\omega}}(\boldsymbol{y}) e^{-i\boldsymbol{\omega}\cdot\boldsymbol{x}}. \tag{6.69}$$

We start by noting that the model has a product structure, in which each (x_i, y_i) pair contributes similarly. Notably, the overlap between the i-th qubit in the state $R_Y(x_i)\,|0\rangle = \cos\left(\frac{x_i}{2}\right)|0\rangle - \sin\left(\frac{x_i}{2}\right)|1\rangle$ and the state $|y_i\rangle$ is:

$$\left|\left(\cos\left(\frac{x_i}{2}\right)\langle 0| - \sin\left(\frac{x_i}{2}\right)\langle 1|\right)|y_i\rangle\right|^2 = \cos^2\left(\frac{x_i+\pi y_i}{2}\right). \tag{6.70}$$

From the Euler decomposition of $\cos(x) = \frac{e^{ix}+e^{-ix}}{2}$, we get:

$$\cos^2\left(\frac{x_i+\pi y_i}{2}\right) = \frac{2+e^{i(x_i+\pi y_i)}+e^{-i(x_i+\pi y_i)}}{4}, \tag{6.71}$$

such that

$$f_{\boldsymbol{y}}(\boldsymbol{x}) = \prod_{i=1}^{n} \frac{2+e^{i(x_i+\pi y_i)}+e^{-i(x_i+\pi y_i)}}{4} \tag{6.72}$$

$$= \sum_{\boldsymbol{\omega}\in\{-1,0,1\}^n} \frac{1}{2^{n+|\boldsymbol{\omega}|}} e^{i(\boldsymbol{x}+\pi\boldsymbol{y})\cdot\boldsymbol{\omega}} \tag{6.73}$$

$$= \sum_{\boldsymbol{\omega}\in\{-1,0,1\}^n} \frac{e^{i\pi\boldsymbol{y}\cdot\boldsymbol{\omega}}}{2^{n+|\boldsymbol{\omega}|}} e^{i\boldsymbol{x}\cdot\boldsymbol{\omega}} \tag{6.74}$$

where $|\boldsymbol{\omega}| = \sum_{i=1}^{n} |\omega_i|$. We can therefore identify $\Omega = \{-1,0,1\}^n$ and $c_{\boldsymbol{\omega}}(\boldsymbol{y}) = \frac{e^{i\pi\boldsymbol{y}\cdot\boldsymbol{\omega}}}{2^{n+|\boldsymbol{\omega}|}}$ and note that the frequency spectrum of this model is exponentially large in n, with all its coefficients being non-zero, exponentially small in n, and differing only by a phase depending on $\boldsymbol{y}$. This justifies the exponential sample complexity needed to identify $\boldsymbol{y}$.

Obfuscation of $\boldsymbol{y}$

At this point, the interested reader might also point out that the obfuscation of the marked basis state $|\boldsymbol{y}\rangle\langle\boldsymbol{y}|$ only occurs somehow artificially by considering the observable $O(\boldsymbol{y})$ as a black-box. That is, we consider as a black-box not only the input-independent gates in the circuit but also the mapping from computational basis state to real values when measuring the output state of the circuit. More naturally, when measuring a basis state $|\boldsymbol{j}\rangle$, one would have a computable function that returns its corresponding eigenvalue, which in this case could reveal $\boldsymbol{y}$. An easy fix for this is to include the encoding of $\boldsymbol{y}$ in the circuit, by redefining $O(\boldsymbol{y})$ as

$$O(\boldsymbol{y}) = \bigotimes_{i=1}^{n} X^{y_i} |0\rangle\langle 0| X^{y_i}, \tag{6.75}$$

which delegates the obfuscation of $\boldsymbol{y}$ to gates in the circuit.

One can also go a step further in this obfuscation by considering instead the following observables

$$O(\boldsymbol{y}) = V_{\mathrm{DLP}} |\boldsymbol{y}\rangle\langle\boldsymbol{y}| V_{\mathrm{DLP}}^{\dagger} \tag{6.76}$$

where V_{DLP} is the unitary[8] that maps a basis state to its discrete logarithm:

$$V_{\mathrm{DLP}} : |\boldsymbol{y}\rangle \mapsto \left|(\log_g(\boldsymbol{y}) \bmod p) + 1\right\rangle = |\boldsymbol{y}'\rangle\,. \tag{6.77}$$

Now, even the knowledge of $\boldsymbol{y}$ and a description of the quantum circuit do not help identify $\boldsymbol{y}'$ classically, under the classical hardness assumption of DLP. Moreover, we still retain the hardness of Fourier-shadowing the resulting model $f_{\boldsymbol{y}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{y})]$ from the same database-search arguments. And finally, the flipped model associated to $f_{\boldsymbol{y}}$ still benefits from the same efficient shadowing procedure, as $O(\boldsymbol{y})$ can be prepared on a quantum computer and measured in the computational basis to reveal $\boldsymbol{y}'$.

6.D.4 Shadowfiablility

In our definition of shadowfiable models in the main text (see Def. 6.3.2), we take the convention that the shadow model should agree with the original quantum model for all possible inputs $\boldsymbol{x} \in \mathcal{X}$. This choice makes sense for two reasons:

[8] Note that this is indeed a unitary transformation, but that potentially needs auxiliary qubits to be implemented unitarily on a quantum computer.

1. We would like the shadowing procedure to work on all potential data distributions, as to be applicable in all learning tasks a given quantum model could be used in.

2. In the context of machine learning, one typically considers PAC conditions, meaning that the final model should achieve a small error $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}|h(\boldsymbol{x}) - g(\boldsymbol{x})|$ only with respect to some data distribution $\mathcal{D}$. Note that if the quantum model to be shadowfied achieves these PAC conditions, our demands on worst-case approximation will guarantee that the shadow model achieves them as well.

Nonetheless, one may still be interested in a notion of shadowfiability that considers an average-case error

$$\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}\left|\widetilde{f}_{\boldsymbol{\theta}}(\boldsymbol{x}) - f_{\boldsymbol{\theta}}(x)\right| \tag{6.78}$$

with respect to a specified data distribution $\mathcal{D}$. It is not entirely clear which models can still be shadowfied in this way. But our results on the universality of flipped models (Lemma 1 in the main text and Lemma 6.D.1 in the Appendix), as well as on the existence of quantum models that are not shadowfiable (Theorem 6.2 in the main text and Lemma 6.D.2) would also hold. More precisely, for each of these results, respectively:

1. The same proof structure of Lemma 6.D.1 can be used, as the constructed flipped model only adds a small controllable error to each $\boldsymbol{x} \in \mathcal{X}$.

2. One can consider here instead of quantum models that compute arbitrary functions in BQP, a restricted model that computes (single bits of) the discrete logarithm $\log_g(\boldsymbol{x})$ mod p (analogous to our DCR concept class defined in Equation 6.46). The result of Liu *et al.* [108] (Theorem 6 in the Supplementary Information) shows the classical hardness of achieving an expected error $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}|h(\boldsymbol{x}) - g(\boldsymbol{x})| \leq 1/2 - 1/\mathrm{poly}(n)$ for such target functions (and a hypothesis h producing labels $h(\boldsymbol{x}) \in \{0, 1\}$), under the assumption that DLP $\notin$ BPP. One can then use this result to show that there exist quantum models that are not average-case shadowfiable under the assumption that DLP $\notin$ P/poly.

Algorithm 6.1: Flipped evaluation of a conventional model

Input: an n-qubit unitary $U(\boldsymbol{x})$ and a quantum state ρ_0 (diagonal in the computational basis) such that $\rho(\boldsymbol{x}) = U(\boldsymbol{x})\rho_0 U^\dagger(\boldsymbol{x})$, n-qubit unitaries $V(\boldsymbol{\theta})$ and $\{W_i\}_{1\leq i\leq d}$, real values $\{w_i\}_{i=1}^d$, $\{\lambda_{i,j}\}_{0\leq j\leq 2^n-1}^{1\leq i\leq d}$, such that $O(\boldsymbol{\theta}) = \sum_i w_i V^\dagger(\boldsymbol{\theta}) O_i V(\boldsymbol{\theta})$ with $O_i = \sum_j \lambda_{i,j} W_i^\dagger |j\rangle\langle j| W_i$.

Output: A flipped evaluation of the conventional model $\mathrm{Tr}[\rho(\boldsymbol{x})O(\boldsymbol{\theta})]$

1 Initialize $o = 0$, $N = \mathcal{O}(\|O\|_1^2/\varepsilon^2)$;
2 **for** N *iterations* **do**
3 Sample $i \in \{1, \ldots, d+1\}$ w.p. $\left(\frac{w_1\|O_1\|_1}{\|O\|_1}, \ldots, \frac{w_d\|O_d\|_1}{\|O\|_1}, \frac{\|O\|_1 - \sum_{i=1}^d w_i\|O_i\|_1}{\|O\|_1}\right)$;
4 **if** $i = d+1$ **then**
5 break iteration (or alternatively prepare $\sigma(\boldsymbol{\theta}) = I/2^{n+1}$ and jump to line 9);
6 Sample $b \in \{+, -\}$ w.p. $\frac{\|O_{i,b}\|_1}{\|O_i\|_1}$;
7 Sample $j \in \{0, \ldots, 2^n - 1\}$ w.p. $\frac{\max(0, b\lambda_{i,j})}{\|O_{i,b}\|_1}$;
8 Prepare $\sigma(\boldsymbol{\theta}) = |\widetilde{b}\rangle\langle\widetilde{b}| \otimes V^\dagger(\boldsymbol{\theta}) W_i^\dagger |j\rangle\langle j| W_i V(\boldsymbol{\theta})$, for $\widetilde{b} = 2b - 1$;
9 Measure $(I \otimes U^\dagger(\boldsymbol{x}))\sigma(\boldsymbol{\theta})(I \otimes U(\boldsymbol{x}))$ in the computational basis and call the outcome $|\widetilde{b}\rangle \otimes |j\rangle$;
10 $o \leftarrow o + b\|O\|_1 \rho_{0,j}$, where $\rho_{0,j}$ is the j-th diagonal element of ρ_0;
11 **return** o/N

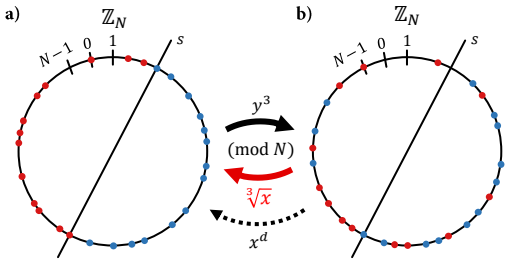

Figure 6.3: A visualization of the functions involved in the quantum advantage learning task. The core functions of this task map $\mathbb{Z}_N = \{0, \ldots, N-1\}$ to itself, for N a large semiprime. a) In feature space, data is linearly separable by a hyperplane parametrized by a certain $s \in \mathbb{Z}_N$. One can efficiently transform data y in feature space into its corresponding data x in input space via the "discrete cube" function $x = y^3 \bmod N$. b) To a fully classical learner, data in input space looks randomly labeled, as inverting it back to feature space via the discrete cube root function $y = \sqrt[3]{x} \bmod N$ is believed to be classically intractable. However, a shadow model can make use of the trap-door property of the discrete cube root function to efficiently compute a key $d \in \mathbb{Z}_N$ using a quantum computer and classically map data to feature space through the transformation $y = x^d \bmod N$.

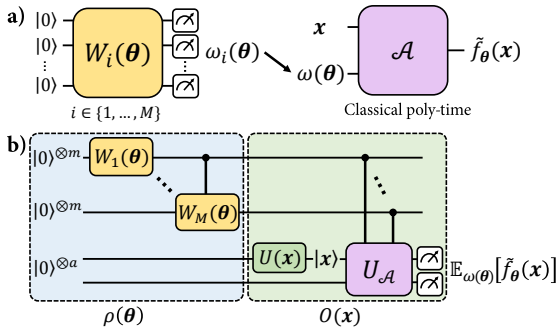

Figure 6.4: All shadow models can be expressed as shadowfiable flipped models. a) A shadow model consists of M unitary circuits $W_i(\boldsymbol{\theta})$ that can be chosen adaptively, and that generate advice $\omega_i(\theta)$ from computational basis measurements of the states $W_i(\boldsymbol{\theta})\,|0\rangle^m$. This advice, along with a (binary description of) an input $\boldsymbol{x} \in \mathbb{R}^d$ are processed by a classical algorithm $\mathcal{A}$ to compute an approximation $\widetilde{f_{\boldsymbol{\theta}}}$ of the shadowfiable model $f_{\boldsymbol{\theta}}$. b) A coherent implementation of this shadow model, where the unitaries $W_i(\boldsymbol{\theta})$ are applied on different m-qubit registers, and coherently controlled by previous registers (for adaptivity). These M registers constitute the coherent encoding of the advice $|\omega(\boldsymbol{\theta})\rangle$. The algorithm $\mathcal{A}$ can then be simulated by a reversible quantum computation $U_{\mathcal{A}}$ (see Sec. 3.2.5. in [140]) that processes a binary encoding $|\boldsymbol{x}\rangle$ of $\boldsymbol{x}$ and the coherent advice $|\omega(\boldsymbol{\theta})\rangle$ (either directly or indirectly via controlled operations that imprint $|\omega(\boldsymbol{\theta})\rangle$ on an auxiliary register). This coherent implementation of the shadow model can be viewed as a shadowfiable flipped model $g_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{Tr}[\rho(\boldsymbol{\theta})O(\boldsymbol{x})]$, such that one evaluation of this model samples an advice $\omega(\boldsymbol{\theta})$ and evaluates $\mathcal{A}(\boldsymbol{x}, \omega(\boldsymbol{\theta}))$ for that advice and a given $\boldsymbol{x}$.

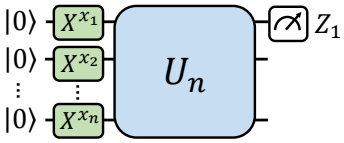

Figure 6.5: A universal quantum model for BQP. For an n-dimensional input $\boldsymbol{x} \in \{0,1\}^n$, this model acts on n qubits, encodes $\boldsymbol{x}$ in its binary form $|\boldsymbol{x}\rangle$ and applies a poly(n)-time unitary U_n before a Pauli-Z measurement of the first qubit. For appropriately chosen unitaries $\{U_n : n \in \mathbb{N}\}$, this model can decide any language in BQP. For more general computational basis measurements, the resulting model can represent arbitrary functions in FBQP, the functional version of BQP.