Computational speedups and learning separations in quantum machine learning

Gyurik, C.

Chapter 6

Exponential separations between classical and quantum learners

In this chapter we address the challenge of finding learning settings where quantum learners can achieve a provable exponential speedup over classical learners in the efficient probably approximately correct (PAC) framework.

First, in Section 6.1, we discuss known learning separations that rely on efficient data generation [126, 173], and we provide a fine-grained analysis of where the classical hardness of learning stems from. While discussing these learning separations, we find that the ones available in literature largely rely on the classical hardness of *evaluating* the function generating the data on unseen points, as opposed to the hardness of *identifying* it. We elaborate how the identification problem can be what is needed in practice, and we address this gap by proving two new learning separations where the classical hardness primarily lies in identifying the function generating the data (see Theorems 24 and 25).

Afterwards, in Section 6.2, we show how leveraging stronger complexity-theoretic assumptions can lead to learning separations where the data is generated by a genuine quantum process. Our main contribution is Theorem 26, which outlines a method of establishing learning separations from BQP-complete functions. We also provide two lemmas, Lemmas 27 and 28, which introduce natural assumptions under which the criteria in Theorem 26 are satisfied. Finally, in Section 6.2.1, we show how Theorem 26 can be used to build learning separations from problems in quantum many-body physics.

To connect our work to some of the related results in the field [107, 109, 146, 97], we discuss selected topics related to learning separations with classical data. In Section 6.3.1 we discuss the milestone work of Huang et al. [107] and how their classical machine learning methods based on the classical shadow framework relate to learning separations with quantum-generated data (i.e., those from Theorem 26). In particular, we highlight their limitations by constructing a family of Hamiltonians whose ground

state properties cannot be predicted based on cryptographic assumptions (see Theorem 30). In Section 6.3.2 we discuss a specific example (i.e., evaluating parameterized quantum circuits) that exemplifies how access to data radically enhances what is efficiently evaluated classically. In Section 6.3.3 we discuss how two physically-motivated problems (i.e., Hamiltonian learning, and identifying order parameters and phases of matter) naturally fit in a learning setting where the learner is constrained to output a hypothesis from a fixed hypothesis class.

6.1 Learning separations with efficient data generation

A commonality between the learning separations of [126, 173] is that the proof of classical non-learnability relies on the fact that the examples can be efficiently generated classically (i.e., the example oracle can be efficiently simulated classically)[1]. This is crucial, since it ensures that access to the example oracle does not enhance what a classical learner can evaluate relative to a conventional (non-learning) classical algorithm. This then allows one to directly deduce classical non-learnability from a complexity-theoretic hardness assumption related to the concepts, since the existence of an efficient classical learner would imply the existence of an efficient classical algorithm. A similar observation was made by the authors of [156] (which came out after our initial observation [94]), where they also study the problem of distribution-independent learning separations.

In this section we study the learning separations of [126, 173], and we characterize them with respect to the type of learning separation they achieve (as discussed in Section 2.5.1), and the kind of hardness assumptions they leverage to obtain classical non-learnability (as discussed in Section 2.5.2). Firstly, in Section 6.1.1, we discuss the CC/QQ separation of the discrete logarithm concept class of [126], whose concepts are believed to be classically intractable, no matter how they are specified. Secondly, in Section 6.1.2, we discuss the CC/QC separation of the cube root concept class of [116, 173], whose concepts are specified in a way that makes them classically intractable, though when specified in a different way they become classically efficient (i.e., the concepts are "obfuscated" versions of classically efficient functions).

While discussing the learning separations of [126, 173], we notice that their proofs largely rely on the classical difficulty of *evaluating* the hypotheses on unseen examples, rather than the difficulty of *identifying* a hypothesis that is close to the concept generating the examples. To complement these works, we present two new examples of learning separations where the classical hardness lies in *identifying* the concept that is generating the examples. Specifically, in Section 6.1.3, we provide an example of a CC/QC separation (contingent on a plausible though relatively unexplored hardness assumption) where the concepts are classically efficiently evaluatable, making it impossible for the classical hardness to come from evaluating them on unseen examples. Afterwards, in Section 6.1.4, we provide an example of a separation in the setting where the learner is constrained to output hypotheses from a fixed hypothesis

[1] The notion of efficiently generatable examples is closely related to the notion of *random verifiability* [25].

class, in which case the learner is only required to identify the concept generating the examples, therefore also eliminating the possibility that the classical hardness comes from evaluating them on unseen examples[2].

6.1.1 A learning separation based on a worst-case to average-case reduction

In this section we discuss the discrete logarithm concept class studied in [126]. In this work, the authors prove that the *discrete logarithm concept class* defined below exhibits a CC/QQ separation.

Definition 17 (Discrete logarithm concept class [126])**.** *Fix an n-bit prime number p and a generator a of $\mathbb{Z}_p^*$ (i.e., the multiplicative group of integers modulo p). We define the* discrete logarithm concept class *as $\mathcal{C}_n^{\mathrm{DL}} = \{c_i\}_{i\in\mathbb{Z}_p^*}$, where*

$$c_i(x) = \begin{cases} 1, & \text{if } \log_a x \in [i, i + \frac{p-3}{2}], \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

Remark(s). *Here $\log_a x$ denotes the discrete logarithm of x with respect to the generator a. That is, the discrete logarithm $\log_a x$ is the smallest positive integer ℓ such that $a^\ell \equiv x \mod p$.*

To see why the examples are efficiently generatable for the discrete logarithm class, first note that the examples are of the form

$$(x, c_i(x)) = (a^y, f_i(y)), \tag{6.2}$$

where $y \in \{1, \ldots, p-1\}$ is the unique integer such that $x \equiv a^y \mod p$, and we let

$$f_i(y) = \begin{cases} 1, & \text{if } y \in [i, i + \frac{p-3}{2}], \\ 0, & \text{otherwise.} \end{cases} \tag{6.3}$$

Secondly, note that $y \mapsto a^y \mod p$ is a bijection from $\{1, \ldots, p-1\}$ to $\mathbb{Z}_p^*$, which implies that sampling $x \in \mathbb{Z}_p^*$ uniformly at random is equivalent to sampling $y \in \{0, \ldots, p-1\}$ uniformly at random and computing $x = a^y \mod p$. By combining this observation with Eq. (6.2), one finds that one can efficiently generate examples of the discrete logarithm concept c_i under the uniform distribution over $\mathbb{Z}_p^*$ by sampling $y \in \{1, \ldots, p-1\}$ uniformly at random, and computing $(a^y, f_i(y))$.

The hardness assumption that one can leverage to obtain classical non-learnability is that the discrete logarithm is classically intractable (i.e., not in BPP)[3]. Namely,

[2]We remark that the concept class of Section 6.1.3 also exhibits a separation in the setting the learner is constrained to output a hypothesis from a fixed hypothesis class. However, we choose to present it as a CC/QC separation to highlight that such separations are still possible if the concepts are classically efficiently evaluatable. Moreover, we still include the separation in the setting the learner is constrained to output a hypothesis from a fixed hypothesis class of Section 6.1.4, because it is not contingent on a relatively unexplored hardness assumption.

[3]For some sequence of primes $\{p_n\}_{n\in\mathbb{N}}$, where $|p_n| = n$ and given n one can efficiently construct p_n.

in [36] it is shown that computing the most-significant bit of the discrete logarithm on any $\frac{1}{2} + \frac{1}{\text{poly}(n)}$ fraction of inputs is at least as hard as computing the discrete logarithm on all inputs. Using the terminology of Section 2.5.2, if one assumes that the discrete logarithm is classically intractable (i.e., not in BPP), then the concepts $c_0 \in \mathcal{C}_n^{\text{DL}}$ lie outside of $\mathsf{HeurBPP}$. In conclusion, to obtain the classical non-learnability it is sufficient to assume that the discrete logarithm is classically intractable.

We remark that one could obtain a similar learning separation for the singleton concept class $\mathcal{C}'_n = \{c\}$ for any choice of $c \in \mathcal{C}_n^{\text{DL}}$. This singleton concept class is quantum learnable without requiring use of the example oracle (i.e. without requiring any data), and one could thus argue that it is not a genuine learning problem anymore (i.e., similar to Observation 1).

Having discussed classical non-learnability, one still needs to ensure quantum learnability. To this end, the authors of [126] show that a general-purpose quantum learning algorithm (i.e., a quantum kernel method) can efficiently learn the discrete logarithm concept class under the uniform distribution. We summarize the result of [126] discussed in this section in the following theorem.

Theorem 22 ([126]). *$L_{\text{DLP}} = \left(\{\mathcal{C}_n^{\text{DLP}}\}_{n\in\mathbb{N}}, \{\mathcal{D}_n^U\}_{n\in\mathbb{N}}\right)$ exhibits a $\mathsf{CC/QQ}$ separation, where $\mathcal{D}_n^U$ denotes the uniform distribution over $\mathbb{Z}_p^*$.*

The hypothesis class the authors of [126] use is quantumly evaluatable, and to the best of our knowledge it is unknown whether the discrete logarithm concepts are quantumly learnable using a classically evaluatable hypothesis class (which would imply a $\mathsf{CC/QC}$ separation).

6.1.2 A learning separation based on obfuscation

The cube root concept class was first studied in [116], and the fact that this concept class exhibits a $\mathsf{CC/QC}$ separation was first observed in [173], albeit using different terminology. We note that there exist many similar concept classes based on public-key cryptosystems such as the RSA cryptosystem that exhibit $\mathsf{CC/QC}$ separations (see [115]). Recall that for $\mathsf{CC/QC}$ separations the hypothesis class has to be classically evaluatable, so the role of the quantum computer is only to identify which hypothesis is close to the concept that is generating the examples.

Definition 18 (Cube root concept class [116]). *Fix an n-bit integer $N = pq$[4], where p and q are two $\lfloor n/2 \rfloor$-bit primes such that $\gcd\big(3, (p-1)(q-1)\big) = 1$. We define the* cube root concept class *as $\mathcal{C}_n^{\text{DCR}} = \{c_i\}_{i\in[n]}$, with*

$$c_i(x) = \text{bin}(f_N^{-1}(x), i),$$

where $\text{bin}(y, i)$ denotes the ith bit of the binary representation of y, and the function f_N^{-1} is the inverse of $f_N(x) = x^3 \mod N$ defined on $\mathbb{Z}_N^*$ (i.e., the multiplicative group of integers modulo N).

Remark(s). *By requiring that p and q satisfy $\gcd\big(3, (p-1)(q-1)\big) = 1$, we ensure that f_N^{-1} exists.*

[4]Throughout this chapter, the integer N is known to the learner beforehand but p and q are not.

To see why the examples are efficiently generatable for the cube root concept class, first note that the examples are of the form

$$(x, c_i(x)) = \left(y^3, \mathrm{bin}(y, i)\right), \tag{6.4}$$

where $y \in \mathbb{Z}_N^*$ is the unique element such that $x \equiv y^3 \mod N$. Secondly, note that $f_N(x) = x^3 \mod N$ is a bijection from $\mathbb{Z}_N^*$ to itself, which implies that sampling $x \in \mathbb{Z}_N^*$ uniformly at random is equivalent to sampling $y \in \mathbb{Z}_N^*$ uniformly at random and computing $x = y^3 \mod N$. By combining this observation with Eq. (6.4), one finds that one can efficiently generate examples of the cube root concept c_i under the uniform distribution over $\mathbb{Z}_N^*$ by first sampling $y \in \mathbb{Z}_N^*$ uniformly at random, and then computing $y^3 \mod N$ together with the ith bit of the binary representation of y.

The hardness assumption that one can leverage to obtain classical non-learnability is what we will call the Discrete Cube Root Assumption (DCRA), which states that computing f_N^{-1} is classically intractable (i.e., not in BPP)[5]. Namely, in [17, 87] it is shown that computing the least-significant bit of f_N^{-1} on any $\frac{1}{2} + \frac{1}{\mathrm{poly}(n)}$ fraction of inputs is at least as hard as computing entire binary representation of $f_N^{-1}(x)$ on all inputs. Using the terminology of Section 2.5.2, if one assumes that computing f_N^{-1} is classically intractable (i.e., not in BPP), then the concepts $c_n \in \mathcal{C}_n^{\mathrm{DCR}}$ lie outside of $\mathsf{HeurBPP}$. In conclusion, analogous to the discrete logarithm concept class, to obtain the classical non-learnability it is sufficient to assume that computing f_N^{-1} is classically intractable.

We remark that also in this case one could obtain a similar learning separation for the singleton concept class $\mathcal{C}'_n = \{c_n\}$ for the concept $c_n \in \mathcal{C}_n^{\mathrm{DCR}}$. This singleton concept class is quantum learnable without requiring the example oracle (i.e. without requiring any data), and one could thus argue that it is not a genuine learning problem anymore (i.e., similar to Observation 1).

However, there is a significant distinction between the learning separations for the discrete logarithm concept class and the cube root concept class that is worth highlighting: the latter is quantumly learnable using a *classically* evaluatable hypothesis class. To see why this is the case, it is important to note that f_N^{-1} is of the form

$$f_N^{-1}(y) = y^{d^*} \mod N, \tag{6.5}$$

for some d^* that only depends on N[6]. The function f_N^{-1} is a type of "trap-door function" in that if one is also given d^*, then computing f_N^{-1} suddenly becomes classically tractable. In other words, there exist polynomially-sized Boolean circuits which evaluate this function, whereas for the discrete logarithm we do not know whether such circuits exist. In this example we thus see the relevance of how the concepts are specified. The specifications "f_N^{-1} where $f(x) = x^3$" and "$f_N^{-1} = x^{d^*}$" refer to the same functions, yet computing them is in one case classically tractable, and in the other case it is classically intractable (under the DCRA). The ideas of concealing (easy) functions in difficult descriptions is reminiscent of the term "obfuscation" in

[5]For some sequence of moduli $\{N_i\}_{i\in\mathbb{N}}$, where $|N_i| = i$ and given i one can efficiently construct N_i.

[6]In cryptographic terms, d^* is the private decryption key corresponding to the public encryption key $e = 3$ and public modulus N in the RSA cryptosystem.

computer science, and we will use this term in this context as well. Specifically, we say that the specification "f_N^{-1} where $f(x) = x^3$" is an obfuscation of the specification "$f_N^{-1} = x^{d^*}$". Using the terminology of Section 2.5.2, this establishes that the problem of evaluating the concepts actually lies inside $\mathsf{P/poly}$ (where the advice string – i.e., d^* – is used to "de-obfuscate" the function).

With regards to quantum learnability, in [173] the authors note that using Shor's algorithm a quantum learning algorithm can efficiently compute d^* following the standard attack on the RSA cryptosystem. The cube root concept class is thus quantumly learnable using the classically evaluatable hypothesis class

$$\left\{ f_{d,i}(x) = \mathrm{bin}(x^d \bmod N, i) \;\middle|\; d \in [N],\ i \in [n] \right\}, \tag{6.6}$$

Another feature of the cube root concept class which warrants a comment is that even though computing d^* does not require access to the example oracle (recall that N is known beforehand), we still have to learn the bit of x^{d^*} that is generating the examples, which does require access to the example oracle (i.e., it requires data). We summarize the results regarding the separation of the cube root concept class in the following theorem.

Theorem 23 ([173, 116])**.** *$L_{\mathrm{DCR}} = \left(\{\mathcal{C}_n^{\mathrm{DCR}}\}_{n\in\mathbb{N}}, \{\mathcal{D}_n^U\}_{n\in\mathbb{N}}\right)$ exhibits a $\mathsf{CC/QC}$ separation, where $\mathcal{D}_n^U$ denotes the uniform distribution over $\mathbb{Z}_N^*$.*

6.1.3 A learning separation with efficiently evaluatable concepts

In this section we establish a learning separation (contingent on a plausible though relatively unexplored hardness assumption) where the concepts do not just admit polynomial-sized Boolean circuits, but are also given in a representation which is efficiently evaluatable on a classical computer. For this concept class, the hardness of learning then cannot stem from the hardness of *evaluating* the concepts, and it thus lies in *identifying* which specific concept is generating the examples. To the best of our knowledge, no such separation was given in the literature before. The concept class that satisfies all of the above is the modular exponentiation concept class defined as follows.

Definition 19 (Modular exponentiation concept class)**.** *Let $N = pq$ be an n-bit 2^c-integer as defined in Definition 20 with* $\gcd\left(3, (p-1)(q-1)\right) = 1$*. We define the* modular exponentiation concept class *as*

$$\mathcal{C}_n^{\mathrm{modexp}} = \left\{ c_d \;\middle|\; d = 1, \ldots, (p-1)(q-1) \text{ and } \gcd(d, (p-1)(q-1) = 1 \right\},$$

where

$$c_d : \mathbb{Z}_N^* \to \mathbb{Z}_N^*, \quad c_d(x) = x^d \mod N, \tag{6.7}$$

and $\mathbb{Z}_N^*$ denotes the multiplicative group of integers modulo N.

Remark(s). *The concepts are not binary-valued, and it is an open question whether and how the separation can be translated to also hold for binary-valued concepts.*

Definition 20 (2^c-integer)**.** *An n-bit integer $N = pq$ is a 2^c-integer if p and q are two $\lfloor n/2 \rfloor$-bit primes such that:*

(i) There exists a constant c (i.e., independent of n) such that $2^c \nmid (p-1)(q-1)$.

(ii) There exists a constant c' (i.e., independent of n) such that $\gcd(p-1, q-1) = 2^{c'}$.

We show that the above concept class is not classically learnable under the assumption that computing f_N^{-1} is classically intractable (i.e., not in BPP) when we restrict N to be a 2^c integer. We will refer to this assumption as the 2^c-discrete cube root assumption (2^c-DCRA). First, note that the modular exponentiation concept class contains the cube root function f_N^{-1} discussed in the previous section (though this time it is not "obfuscated"). Moreover, using the construction from the previous section we can efficiently generate examples $(y, f_N^{-1}(y))$, for $y \in \mathbb{Z}_N^*$ uniformly at random. If we put these examples into an efficient classical learning algorithm for the modular exponentiation concept class, it would with high probability identify a classically efficiently evaluatable hypothesis that agrees with f_N^{-1} on a $1 - \frac{1}{\mathrm{poly}(n)}$ fraction of inputs. Analogous to the previous section, by the worst-case to average-case reduction of [17, 87] this directly violates the 2^c-DCRA.

We note that by imposing that N is a 2^c-integer might cause the 2^c-DCRA to no longer hold, since there could be an efficient classical algorithm for these specific 2^c-integer moduli. However, since 2^c-integers are generally not considered to be unsecure or "weak" moduli for the RSA cryptosystem, and since recently factored RSA numbers[7] are all essentially 2^c-integers, it is plausible that the 2^c-DCRA still holds (see Appendix D.2.1 for more details).

To show that the modular exponentiation concept class is quantumly learnable, we use a combination of the quantum algorithm for order-finding and the quantum algorithm for the discrete logarithm [176]. The key observation is that an example $(x, x^d \bmod N)$ specifies a congruence relation $d \equiv a \bmod r$, where r denotes the multiplicative order of $x \in \mathbb{Z}_N^*$, and a denotes the discrete logarithm of x^d in the subgroup generated by x (i.e., the smallest positive integer ℓ such that $x^\ell \equiv x^d \bmod N$). Next, using the fact that N is a 2^c-number, we show that a polynomial number of these congruences suffices to recover d with high probability. We summarize the learning separation of the modular exponentiation concepts in Theorem 24, and defer the proof to Appendix D.2.

Theorem 24. *If the 2^c-DCRA holds, then the learning problem*

$$L_{\mathrm{modexp}} = \left(\{\mathcal{C}_n^{\mathrm{modexp}}\}_{n\in\mathbb{N}}, \{\mathcal{D}_n^U\}_{n\in\mathbb{N}}\right)$$

exhibits a $\mathsf{CC/QC}$ separation, where $\mathcal{D}_n^U$ denotes the uniform distribution over $\mathbb{Z}_N^*$.

In conclusion, the modular exponentiation concept class exhibits a $\mathsf{CC/QC}$ separation (assuming the 2^c-DCRA hold), where the concepts are classically efficiently

[7] `https://en.wikipedia.org/wiki/RSA_numbers`

evaluatable. Since the concepts are classically efficiently evaluatable, one could argue that the classical hardness lies in *identifying* rather than *evaluating* a hypothesis that is close to the concept generating the examples. We remark that for the modular exponentation concept class, it is not possible to restrict the concept class and obtain a similar learning separation where a quantum learner does not require any data (i.e., similar to Observation 1). In fact, since the concepts are efficiently evaluatable, any polynomially-sized subset of concepts is classically learnable since one can simply do a brute-force search to find the concept that best matches the data.

In the next section, we present an example of a separation in the setting where the learner is constrained to only output hypotheses from a fixed hypothesis class. Since the learner is not required to evaluate the concepts on unseen examples, it can be argued that in this case the classical hardness also lies in identifying rather than evaluating the concept generating the examples.

6.1.4 A learning separation with a fixed hypothesis class

In this section we establish a separation in the setting where the learner is constrained to only output hypotheses from a fixed hypothesis class. Recall that in this setting the learner is not required to be able to evaluate the concepts, so the hardness of learning must stem from the hardness of identifying the hypothesis that is close to the concept generating the data. The main differences compared to the modular exponentiation concept class are that the concepts discussed in this section are binary-valued and that it is unknown whether they exhibit a separation in the setting where the learner is free to output arbitrary hypotheses. The concept class we discuss in this section is defined below, and it is a modification of the cube root concept class from Definition 18.

Definition 21 (Cube root identification concept class)**.** *Fix an n-bit integer $N = pq$, where p and q are two $\lfloor n/2 \rfloor$-bit primes such that* $\gcd\big(3, (p-1)(q-1)\big) = 1$. *We define the* cube root identification concept class *as $\mathcal{C}_n^{\mathrm{DCRI}} = \{c_m\}_{m \in \mathbb{Z}_N^*}$, with*

$$c_m(x) = \mathrm{bin}(m^3 \bmod N,\ \mathrm{int}(x_1 : \cdots : x_{\lfloor \log n \rfloor})),$$

where $\mathrm{bin}(y, k)$ *denotes the kth bit of the binary representation of y, and* $\mathrm{int}(x_1 : \cdots : x_{\lfloor \log n \rfloor})$ *is the integer encoded by the first $\lfloor \log n \rfloor$-bits of $x \in \{0,1\}^n$.*

We show that the cube root identification concept class is not classically learnable with a fixed hypothesis class under the *Discrete Cube Root Assumption* (DCRA) discussed in Section 6.1.2. To show that the existence of an efficient classical learner violates the DCRA, we let $e \in \mathbb{Z}_N^*$ and show we how an efficient classical learner can efficiently compute $m = f_N^{-1}(e)$. First, we generate examples $(x, \mathrm{bin}(e, k))$, where $k = \mathrm{int}(x_1 : \cdots : x_{\lfloor \log n \rfloor})$. When plugging these examples into an efficient classical learner it will with high probability identify an m' such that $(m')^3 \equiv m \mod N$. Since $x \mapsto x^3 \mod N$ is a bijection on $\mathbb{Z}_N^*$ we find that $m = m'$, and thus conclude that an efficient classical learner can indeed efficiently compute the solution to our discrete cube root instance.

To establish that the cube root identification concept class is quantumly proper learnable, we first note that using $\mathcal{O}(\mathrm{poly}(n))$ examples of a concept c_m under the

uniform distribution we can with high probability reconstruct the full binary representation of m^3. Next, since N is known we can use Shor's algorithm [176] to compute d such that $(m^3)^d \equiv m \bmod N$, which allows us to correctly identify the concept c_m. Note that the quantum learner needs access to the data to obtain a full reconstruction of the binary representation of m^3. We summarize the learning separation of the cube root identification concept class in Theorem 25, and we defer the proof to Appendix D.3.

Theorem 25. *$L_{\mathrm{DCRI}} = \left(\{\mathcal{C}_n^{\mathrm{DCRI}}\}_{n\in\mathbb{N}}, \{\mathcal{D}_n^U\}_{n\in\mathbb{N}}\right)$ exhibits a $\mathsf{C}_{\mathcal{H}}/\mathsf{Q}_{\mathcal{H}}$ separation, where $\mathcal{H} = \mathcal{C}^{\mathrm{DCRI}}$ and $\mathcal{D}_n^U$ denotes the uniform distribution over $\mathbb{Z}_N^*$.*

In conclusion, the cube root identification concept class exhibits a separation in the setting where the learner is constrained to only output hypotheses from a fixed hypothesis class. In fact, this is a separation in the *proper* efficient PAC learning framework, since the hypothesis class is the same as the concept class. Since in this setting it is not required to evaluate the concepts on unseen examples, the classical hardness has to lie in *identifying* rather than *evaluating* the concept generating the examples. We remark that for the cube root identification concept class it is not possible to obtain a similar learning separation for a singleton concept class where a learner does not require any data (see also Observation 1).

6.2 Learning separations without efficient data generation

In the quantum machine learning community there is an often-mentioned conjecture that quantum machine learning is most likely to have its advantages for data that is generated by a "genuine quantum process"[1]. We understand this to mean that the concepts generating the data are BQP-complete or perhaps DQC1-complete. It is worth noting that if concepts in BQP or DQC1 that are not in BPP are already considered a "genuine quantum process", then the discrete logarithm concept class discussed in Section 6.1.1 suffices. However, we aim to investigate learning separations beyond these concepts, i.e., where the concepts are BQP-complete.

A natural question that arises is, given a family of BQP-complete concepts, what additional assumptions are sufficient to prove that these concepts exhibit a learning separation? In Section 6.1, we discussed proofs of learning separations that were predicated on the data being efficiently generatable by a classical device. However, since there is no reason to believe that a family of BQP-complete concepts allow for efficient data generation, we will need to adopt a different proof-strategy.

To ensure quantum learnability of a family of BQP-complete concepts $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$, we can simply limit the size of each concept class $\mathcal{C}_n$ to be no more than a polynomial in n. When the size of the concept class is polynomial, a quantum learner can iterate over all concepts and identify the concept that best matches the examples from the oracle. In more technical terms, a quantum learner can efficiently perform empirical risk

[1] Recently, there have been notable developments that have yielded contrasting conclusions. For instance, in [107], surprisingly complex physics problems are efficiently learned by classical learners. We will briefly discuss this in Section 6.3.1.

minimization through brute-force fitting. From standard results in learning theory (e.g., Corollary 2.3 in [174]), it follows that this method results in a learner that satisfies the conditions of the PAC learning framework.

As discussed in Section 2.5.2, assuming that the concepts are not in HeurP/poly is sufficient to ensure that the concept class is not classically learnable. Intuitively, this is because if the concepts were classically learnable, the examples could be used to construct an advice string that, together with an efficient classical learning algorithm, would put the concepts in HeurP/poly. By combining this with our approach to ensure quantum learnability, we can show that if there exists a family of polynomially-sized concept classes consisting of BQP-complete concepts that are not in HeurP/poly, then this family of concept classes exhibits a CC/QQ separation. Moreover, in Section 6.2.1 we discuss how several of these separations can be build around data that is generated by a "genuine quantum process". The following theorem summarizes our findings, and we defer the proof to Appendix D.4.

Theorem 26. *Consider a family of concept classes* $\{\mathcal{C}_n\}_{n\in\mathbb{N}}$ *and distributions* $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$ *such that*

Quantum learnability:

(a) Every $c_n \in \mathcal{C}_n$ *can be evaluated quantumly in time* $\mathcal{O}(\text{poly}(n))$.

(b) There exists a polynomial p *such that for every* $n \in \mathbb{N}$ *we have*

$$|\mathcal{C}_n| \leq p(n).$$

Classical non-learnability:

(c) There exists a family $\{c_n\}_{n\in\mathbb{N}}$*, where* $c_n \in \mathcal{C}_n$*, such that*

$$(\{c_n\}_{n\in\mathbb{N}}, \{\mathcal{D}_n\}_{n\in\mathbb{N}}) \notin \mathsf{HeurP/poly}.$$

Then, $L = (\{\mathcal{C}_n\}_{n\in\mathbb{N}}, \{\mathcal{D}_n\}_{n\in\mathbb{N}})$ *exhibits a* CC/QQ *learning separation.*

At face value, it may not be clear whether there exist concept classes that satisfy both conditions (a) and (c), since condition (a) puts the concepts in BQP and it may not be clear how large HeurP/poly is relative to BQP. Notably, it is known that if the discrete logarithm is not in BPP, then it is also not in HeurP under certain distributions. Additionally, it is widely believed that a polynomial amount of advice does not significantly improve the computational complexity of solving the discrete logarithm problem. Hence, it is plausible to imagine the existence of problems $L \in \mathsf{BQP}$ for which there is a distribution $\mathcal{D}$ such that $(L, \mathcal{D}) \notin \mathsf{HeurP/poly}$. Moreover, it is interesting to observe that if there exists a single $L \in \mathsf{BQP}$ that is not in HeurP/poly under some distribution, then for every BQP-complete problem there exists a distribution under which it is not in HeurP/poly. We summarize this in the lemma below, and we defer the proof to Appendix D.4.1.

Lemma 27. *If there exists a* $(L, \mathcal{D}) \notin \mathsf{HeurP/poly}$ *with* $L \in \mathsf{BQP}$*, then for every* $L' \in \mathsf{BQP\text{-}complete}$[2] *there exists a family of distributions* $\mathcal{D}' = \{\mathcal{D}'_n\}_{n\in\mathbb{N}}$ *such that* $(L', \mathcal{D}') \notin \mathsf{HeurP/poly}$.

In summary, to obtain a learning separation for data generated by a "genuine quantum process", it is sufficient to have a single problem $L \in \mathsf{BQP}$ that lies outside $\mathsf{HeurP/poly}$ under some distribution. An example of such a problem is the discrete logarithm. However, the resulting distribution under which the BQP-complete problem lies outside of $\mathsf{HeurP/poly}$ is artificial as it comes from explicitly encoding the discrete logarithm into the learning problem through the reduction to the BQP-complete problem. Besides the discrete logarithm, little is known about the heuristic hardness of problems in BQP (especially those that are considered "genuinely quantum"). Therefore, the question arises as to what additional properties are required for a BQP-complete problem to lie outside $\mathsf{HeurP/poly}$ under some distribution. We show that a worst-case to average-case reduction combined with the assumption that $\mathsf{BQP} \not\subseteq \mathsf{P/poly}$ is sufficient for this purpose. While the question of $\mathsf{BQP} \not\subseteq \mathsf{P/poly}$ remains open, we proceed under this assumption based on its implications for cryptography. Specifically, if $\mathsf{BQP} \subseteq \mathsf{P/poly}$, then problems like the discrete logarithm would be in $\mathsf{P/poly}$, which would break cryptographic systems assumed to be secure[3]. Under the assumption that $\mathsf{BQP} \not\subseteq \mathsf{P/poly}$, the only missing piece is that our problem $L \in \mathsf{BQP}$ that lies outside $\mathsf{P/poly}$ is random self-reducible with respect to some distribution (i.e., it admits a worst-case to average case reduction as discussed in Section 2.5.2). We summarize these findings in the lemma below, and we defer the proof to Appendix D.4.2.

Lemma 28. *If* $L \notin \mathsf{P/poly}$ *and* L *is polynomially random self-reducible with respect to some distribution* $\mathcal{D}$*, then* $(L, \mathcal{D}) \notin \mathsf{HeurP/poly}$.

By combining Lemma 27 with Lemma 28, we obtain a set of assumptions that result in provable learning separations for data that could be generated by a genuine quantum process, as stated in Theorem 26 (see also Section 6.2.1). These assumptions include the existence of a problem $L \in \mathsf{BQP}$ that is not in $\mathsf{P/poly}$ which is polynomially random self-reducible with respect to some distribution.

Corollary 29. *If there exists an* $L \in \mathsf{BQP}$ *such that* $L \notin \mathsf{P/poly}$ *and it is random self-reducible, then every* BQP*-complete problem gives rise to a* $\mathsf{CC/QQ}$ *separation.*

Although establishing such learning separations is not straightforward, the criteria listed in Lemma 27, Lemma 28 and Corollary 29 suggest some challenges that when addressed lead to provable learning separations. In particular, they highlight the need for further investigation into the heuristic hardness of problems in BQP from the perspective of quantum machine learning.

[2] With respect to many-to-one reductions (as is the case for, e.g., quantum linear system solving [101]).

[3] In cryptography it is common to assume non-uniform adversaries (i.e., with computational resources of $\mathsf{P/poly}$), and even in this case most public-key cryptosystems such as RSA and Diffie-Hellman are still assumed to be safe).

Generative modeling To the authors it is not entirely clear precisely how strong the assumption that $\mathsf{BQP} \not\subseteq \mathsf{P/poly}$ is. It is worth noting though that for sampling problems arguably more iron-clad assumptions, such as the non-collapse of the polynomial hierarchy, could potentially lead to analogous conclusions. In particular, one possibility is to use quantum supremacy arguments [10, 42] to establish learning separations in generative modeling, where the task is to learn a distribution instead of a binary function. If the distribution to be learned is in $\mathsf{SampBQP}$ (i.e., sampling problems solvable by a polynomial-time quantum algorithm), then for classical non-learnability, the corresponding requirement is that not all $\mathsf{SampBQP}$ problems are in $\mathsf{SampBPP/poly}$ (i.e., sampling problems solvable by a polynomial-time classical algorithm with polynomial-sized advice)[4]. This is analogous to the supervised learning case, but for sampling problems we might have further evidence this is unlikely. Specifically, as sketched by Aaronson in [9], if $\mathsf{SampBQP} \subseteq \mathsf{SampBPP/poly}$, then this could cause the polynomial hierarchy to collapse. In other words, one could arguably use these arguments to show that a family of distributions is not classically learnable, under the assumption that the polynomial hierarchy does not collapse.

6.2.1 Learning separations from physical systems

Many quantum many-body problems are either BQP-complete or QMA-complete when appropriately formalized, making them suitable for defining concepts that are not classically learnable (recall that this also implies a learning separation, since quantum learnability can be ensured by considering polynomially-sized concept classes). To be more precise, recall that any problem in BQP that does not lie in $\mathsf{HeurP/poly}$ with respect to some distribution can be used to construct a distribution under which a hard quantum many-body problem defines a learning problem that is not classically learnable (as shown by Theorem 26 and Lemma 27). However, the induced distribution under which the physical system is not classically learnable is artificial, as it is induced by a particular choice of reduction, and there is no evidence that these induced distributions are relevant in practice.

Examples of physical systems For concreteness, let us discuss some examples. There are many physical systems that are in some sense universal for quantum computing, such as the Bose-Hubbard model [62], the antiferromagnetic Heisenberg and antiferromagnetic XY model [159], the Fermi-Hubbard model [150], supersymmetric systems [49], interacting bosons [197], and interacting fermions [125]. In particular, each of these physical systems defines a family of Hamiltonians and, for several of these Hamiltonian families, time-evolution is BQP-complete when appropriately formalized [96, 63]. That is, for several of these universal Hamiltonian families $H(\beta)$, where β denote the Hamiltonian parameters, we can define BQP-complete concepts

$$c_H(\beta, t) = \operatorname{sign}\left(\left|\langle 0^n|e^{iH(\beta)t} Z_1 e^{-iH(\beta)t}|0^n\rangle\right|^2 - \frac{1}{2}\right),$$

[4]For a formal definition of these complexity classes we refer to [10].

where Z_1 denotes the Pauli-Z operator on the first qubit and identity elsewhere. Additionally, one could also use BQP-complete problems in high energy physics, such as scattering in scalar quantum field theory [112].

As another example, we note that for any of the universal Hamiltonian families the problem of finding the ground state energy is QMA-complete. That is, for any universal Hamiltonian family $H(\beta)$, where β denote the Hamiltonian parameters, we can define QMA-complete concepts

$$c_H(\beta) = \text{sign}\left(\text{Tr}\left[H(\beta)\left|\psi_H(\beta)\right\rangle\right] - \frac{1}{2}\right),$$

where $|\psi_H(\beta)\rangle$ denotes the ground state of $H(\beta)$. Naturally, one worries that these concepts are too hard to evaluate on a quantum computer, but there are a few workarounds. Firstly, sometimes there is a natural special case of the problem that is BQP-complete (e.g., the subset of Hamiltonians obtained through a circuit-to-Hamiltonian mapping). Moreover, more generically it holds that any problem that is QMA-complete has a restriction that is BQP-complete (i.e., take any BQP-complete problem and consider the image of this problem under a many-to-one reduction). Finally, one could use recent results on the guided local Hamiltonian problem to relax the QMA-complete problems and obtain a BQP-complete problem [196, 50, 82].

In short, by exploiting a reduction from a problem that is in BQP which under a given distribution lies outside HeurP/poly onto a chosen BQP-complete problem (as in Lemma 27), any physical system that is in some sense universal for quantum computing can be used to construct a learning separation. Nonetheless, since the reduction is implicitely used to construct the distribution under which the physical system becomes not classically learnable, the distributions will be artificial and there is no reason to believe these have any relevance in practice.

6.3 Connections to other works on (quantum) learning tasks

In this section we discuss other topics of relevance. First, in Section 6.3.1, we discuss the implications and limitations of the milestone work of Huang et al. [107] on establishing learning separations from physical systems. Next, in Section 6.3.2, we discuss how having access to data radically enhances what can be efficiently evaluated by discussing the example of evaluating parameterized quantum circuits. Afterwards, in Section 6.3.3, we discuss how two physically-motivated problems (i.e., Hamiltonian learning, and identifying order parameters for phases of matter) fit in the PAC learning setting where the learner is constrained to output hypotheses from a fixed hypothesis class.

6.3.1 Provably efficient machine learning with classical shadows

In the milestone work of Huang et al. [107], the authors design classical machine learning methods (in part built around the *classical shadow* paradigm) that can efficiently learn quantum many-body problems. One of the problems studied in [107] is that of *predicting ground states of Hamiltonian*. More precisely, for a family of Hamiltonians $H(x)$ with ground states $\rho_H(x)$, one wants to predict the expectation value of some observable O when measured on $\rho_H(x)$. That is, one wants to efficiently learn to evaluate the function

$$f_{H,O}(x) = \mathrm{Tr}\left[\rho_H(x)O\right]. \tag{6.8}$$

One of the main things that [107] show is that given a polynomial number of data points, one is able to efficiently evaluate the functions in Eq. (6.8) with a constant expected error under certain criteria. Recall that in Section 6.2.1 we argued that concepts based on physical systems can be used as a source of learning separations. Since these concepts are of a similar form as the functions described in Eq. (6.8), one might wonder how the results of Huang et al. relate.

Let us take a closer look at the requirements of the methods described in [107]. Firstly, the Hamiltonians $H(x)$ must all be geometrically-local, and the observable O must be a sum of polynomially many local observables $O = \sum_{i=1}^{L} O_i$ such that $\sum_{i=1}^{L} ||O_i||$ is bounded by a constant. Additionally, the Hamiltonians $H(x)$ must all have a constant spectral gap (i.e., the difference between the smallest and the next smallest eigenvalue) and they must depend smoothly on x (or more precisely, the average gradient of the function in Eq. (6.8) must be bounded by a constant). One might wonder what will happen if we relax the above requirements, while simultaneously maintaining the fact that a quantum computer would still be able to evaluate the function in Eq. (6.8) (and hence build a learning separation around it based on Theorem 26).

Two possible relaxations of the requirements are the absence of a constant spectral gap (while maintaining an inverse polynomial spectral gap) and a reduced smoothness dependency of the Hamiltonian family on x (i.e., compared to what is required for the methods of [107]). It turns out that if one relaxes these requirements, then under cryptographic assumptions the methods proposed by Huang et al. are no longer capable of evaluating the function in Eq. (6.8) with constant expected error. More precisely, any classical machine learning method that would still be able to evaluate the function in Eq. (6.8) up to constant expected error under the relaxed assumptions would be able to solve DLP in P/poly, which contradicts certain cryptographic assumptions. We provide a formal statement of this in the following theorem, the proof of which is deferred to Appendix D.5.

Theorem 30. *Suppose there exists a polynomial-time randomized classical algorithm $\mathcal{A}$ with the following property: for every geometrically-local family of n-qubit Hamiltonians $H(x)$ there exist a dataset $\mathcal{T}_H \in \{0,1\}^{\mathrm{poly}(n)}$ such that for every sum $O = \sum_{i=1}^{L} O_i$ of $L \in \mathcal{O}(\mathrm{poly}(n))$ many local observables with $\sum_{i=1}^{L} ||O_i|| \leq B$ for*

some constant B, the function

$$\overline{f}_{H,O}(x) = \mathcal{A}(x, O, \mathcal{T}_H)$$

satisfies

$$\mathbb{E}_{x\sim[-1,1]^m}\left[\, |\overline{f}_{H,O}(x) - f_{H,O}(x)| \,\right] < \frac{1}{6},$$

where $f_{H,O}(x) = \mathrm{Tr}\left[\rho_H(x)O\right]$ and $\rho_H(x)$ denotes the ground state of $H(x)$. Then, $\mathsf{DLP} \in \mathsf{P/poly}$.

In conclusion, Theorem 30 shows that any method similar to that of [107] cannot learn to predict ground state properties of certain physical systems discussed in Section 6.2.1. Moreover, there are a few subtle differences between the setup of [107] and the one discussed in this thesis. Firstly, the classical shadow paradigm uses data that is different from the PAC learning setting (i.e., the data does not correspond to evaluations of the function it aims to predict). This distinction in setup makes the approach of [107] more versatile, as their data can be utilized to evaluate multiple different observables (moreover, their methods also work in the PAC setting). Secondly, the functions $f_{H,O}$ in Eq. (6.8) are real-valued, which differs from our setting where we investigate functions that map onto a discrete label space. It is possible to address this difference by applying a threshold function to $f_{H,O}$ after it is learned. However, this thresholding introduces a mismatch in the types of data, as it would involve using real-valued data to learn a function with discrete values (which is clearly different from the PAC setting).

6.3.2 Power of data

In [109] the authors show how having access to data radically enhances what can be efficiently evaluated. In this section we connect the ideas from their work to the formalism we introduce in this thesis. Specifically, we will discuss a family of functions inspired by [109] that from their description alone cannot be efficiently evaluated classically, yet access to a few examples (i.e., evaluations of the function) allows them to be efficiently evaluated classically. This highlights an important difference between complexity-theoretic separations and learning separations, since in the latter one has to deal with the learner having access to data when proving classical non-learnability.

Consider a polynomial-depth parameterized quantum circuit $U(\theta, \vec{\phi})$ – with two types of parameters $\theta \in \mathbb{R}$ parameterizing a single gate and $\vec{\phi} \in \mathbb{R}^\ell$ parameterizing multiple other gates – that is universal in the sense that for every polynomial-depth circuit V there exists parameters $\vec{\phi}^* \in \mathbb{R}^\ell$ such that

$$U(0, \vec{\phi}^*)\,|0^n\rangle = V\,|0^n\rangle\,.$$

Moreover, assume the gates in U are of the form $\exp\left(-\frac{i\theta}{2}A\right)$, with $A^2 = I$ (e.g., Z- or X-rotations). By measuring the output of the circuit we define a family of single parameter functions given by

$$f_{\vec{\phi}}(\theta) = \langle 0^n|\, U(\theta, \vec{\phi})^\dagger M U(\theta, \vec{\phi})\,|0^n\rangle\,.$$

Following an argument similar to [109], due to the universality of the parameterized quantum circuit no efficient randomized classical algorithm can take as input a $\vec{\phi} \in \mathbb{R}^\ell$ and compute the function $f_{\vec{\phi}}$ on a given point $\theta \in \mathbb{R}$ up to constant error in time $\mathcal{O}(\text{poly}(n))$, unless $\mathsf{BPP} = \mathsf{BQP}$. Intuitively, one might thus think that the concept class $\{f_{\vec{\phi}} \mid \vec{\phi} \in \mathbb{R}^\ell\}$ exhibits a separation between classical and quantum learners. However, it turns out that the examples given to a classical learner radically enhance what it can efficiently evaluate. In particular, given a few of evaluations of $f_{\vec{\phi}}$ for some fixed but arbitrary $\vec{\phi} \in \mathbb{R}^\ell$, a classical learner is suddenly able to efficiently evaluate the function. To see this, note that by [144] one can write the functions as

$$f_{\vec{\phi}}(\theta) = \alpha \cos(\theta - \beta) + \gamma, \quad \text{for } \alpha, \beta, \gamma \in \mathbb{R},$$

where the coefficients α, β and γ are all independent of θ (but they do depend on $\vec{\phi}$). From this we can see that any three distinct examples $\left\{\left(\theta_i, f_{\vec{\phi}}(\theta_i)\right)\right\}_{i=1}^3$ uniquely determine $f_{\vec{\phi}}(\theta)$ and one can simply fit α, β and γ to these three examples to learn how to evaluate $f_{\vec{\phi}}$ on unseen points. We would like to point that the BQP-hard problem in question is not evaluating $f_{\vec{\phi}}$ for a fixed $\vec{\phi} \in \mathbb{R}^\ell$, but rather evaluating $f_{\vec{\phi}}$ when $\vec{\phi} \in \mathbb{R}^\ell$ is part of the input. This approach can be generalized to settings with more than one free parameter θ, by using the fact that expectation values of parameterized quantum circuits can be written as a Fourier series [171]. Specifically, when the number of frequencies appearing in the Fourier series is polynomial, then a polynomial number of examples suffices to fit the Fourier series and learn how to evaluate the expectation value of the quantum circuits for an arbitrary choice of parameters.

As discussed in Section 6.1, one way to deal with the fact that data can radically enhance what can be efficiently evaluated is to ensure that the data itself is efficiently generatable. However, for the concepts discussed above, the examples are such that only a quantum computer can generate them efficiently. In other words, these functions exemplify how hard to generate data can radically enhance what a classical learner can efficiently evaluate. As discussed in Section 6.1, another way to deal with the fact that data can radically enhance what can be efficiently evaluated is to ensure that the concepts lie outside of $\mathsf{HeurP/poly}$. However, for the case discussed above, every $f_{\vec{\phi}}$ corresponds to a function in $\mathsf{HeurP/poly}$, since the coefficients α, β and γ suffice as the advice. Finally, we note that for certain circuits one could have exponentially many terms in the Fourier series [54, 53], in which case it is unclear how to classically learn them.

6.3.3 Physically-motivated PAC learning settings with fixed hypothesis classes

Throughout this thesis we mainly focused on the setting where the learner is allowed to output arbibtrary hypotheses (barring that they have to be tractable as discussed in Appendix D.1.1). However, we want to highlight that setting where the learner is constrained to only be able to output hypothesis from a fixed hypothesis class is also relevant from a practical perspective. In particular, in this section discuss how two well-studied problems (i.e., Hamiltonian learning, and identifying order parameters

for phases of matter) fit in this setting. Recall that in this setting, it is allowed and reasonable for the hypothesis class to be classically- or quantumly- intractable.

Hamiltonian learning In Hamiltonian learning one is given measurement data from a quantum experiment, and the goal is to recover the Hamiltonian that best matches the data. Throughout the literature, various different types of measurement data have been considered. For example, it could be measurement data from ground states, (non-zero temperature) thermal sates, or time-evolved states. In our case, the data will be measurement data from time-evolved states and we formulate Hamiltonian learning in terms of a hypothesis class as follows. First, we fix a (polynomially-sized) set of Hermitian operators $\{H_\ell\}_{\ell=1}^{L}$. Next, we consider a family of Hamiltonians $\{H_\beta\}_{\beta\in\mathbb{R}^L}$, where

$$H_\beta = \sum_{\ell=1}^{L} \beta_\ell H_\ell. \tag{6.9}$$

Finally, we define the hypothesis class $\mathcal{H}^{\mathrm{HL}} = \{h_\beta\}_{\beta\in\mathbb{R}^L}$, with concepts defined as

$$h_\beta(z,t) = \mathrm{sign}\big(\mathrm{Tr}\big[U^\dagger(t)\rho_z U(t) O_z\big]\big), \quad U(t) = e^{itH_\beta}. \tag{6.10}$$

Here z describes the experimental setup, specifying the starting state (that will evolve under H_β for time t) and the observable measured at the end. A natural specification of the concepts that a learner could output are the parameters β. In particular, in Hamiltonian learning we are only concerned with identifying which concept generated the data (i.e., what is the specification of the underlying Hamiltonian), as opposed to finding a hypothesis that closely matches the data. In other words, the problem of Hamiltonian learning can naturally be formulated as PAC learning setting where the learner is constrained to only be able to output hypotheses described in Eq. (6.10).

With respect to learning separations, one might think that the above setting is a good candidate to exhibit a $\mathsf{C}_{\mathcal{H}^{\mathrm{HL}}}/\mathsf{Q}_{\mathcal{H}^{\mathrm{HL}}}$ separation, since the hypotheses are classically intractable and quantumly efficient to evaluate (assuming $\mathsf{BPP} \neq \mathsf{BQP}$). Moreover, according to the folklore, quantum learners are most likely to have its advantages for data that is "quantum-generated", which certainly seems to be the case here. However, recall that in the setting where the learner is constrained to output hypotheses from a fixed hypothesis class the task is not to evaluate, but rather to identify the concept generating the examples. Therefore, the arguments we used throughout this thesis do not directly apply. In fact, it turns out that classical learners can efficiently identify the parameters of the Hamiltonian generating the data in many natural settings [20, 98, 108], eliminating the possibility of a $\mathsf{C}_{\mathcal{H}^{\mathrm{HL}}}/\mathsf{Q}_{\mathcal{H}^{\mathrm{HL}}}$ separation.

Order parameters and phases of matter When studying phases of matter one might want to identify what physical properties characterize the phase. One can formulate this problem as finding a specification of the correct hypothesis selected from a hypothesis class consisting of possible *order parameters*. In particular, we fix the hypotheses $\mathcal{H}^{\mathrm{order}} = \{h_\alpha\}$ to be of a very special form, which compute certain

expectation values of ground states given a specification of a Hamiltonian. That is, we formally define the hypotheses as

$$h_\alpha(\beta) = \text{sign}\left(\text{Tr}\left[O_\alpha \rho_\beta\right]\right), \tag{6.11}$$

where ρ_β denotes the ground state of some Hamiltonian specified by β (e.g., using the parameterization in Eq. (6.9)), and α specifies an observable O_α drawn from a set of observables that are deemed potential candidates for the order parameter that characterize the phase. In this setting, one might not necessarily want to evaluate the hypotheses, as they might require one to prepare the ground state, which is generally intractable (even for a quantum computer). However, one might still want to identify the observable O_α that correctly characterizes the phase of the physical system specified by β (i.e., the corresponding *order parameter*). In other words, the problem of identifying order parameters naturally fits in the PAC learning setting where the learner is constrained to only be able to output hypotheses described in Eq. (6.11).

As in the case of Hamiltonian learning, one might think that the above concepts are good candidates to exhibit a $\mathsf{C}_{\mathcal{H}^{\text{order}}}/\mathsf{Q}_{\mathcal{H}^{\text{order}}}$ separation, since the hypotheses are classically intractable and quantumly efficient to evaluate (assuming $\mathsf{BPP} \neq \mathsf{BQP}$). In fact, according to the folklore, quantum learners are most likely to have advantages for data that is "quantum-generated", which certainly seems to also be the case here. However, as already mentioned, in the setting where the learner is constrained to output hypotheses from a fixed hypothesis class the goal is only to identify the correct hypothesis, and it is therefore not enough to just have concepts that are classically intractable. We remark that the methods of [107] also apply to phase classification, but they are more aimed at the PAC learning setting where the learner can output arbitrary hypotheses (i.e., the main goal is to predict the phase of a given physical system). In particular, their methods do not directly allow one to obtain a physically-meaningful description of the order-parameter, which is the main goal in the setting where the learner is constrained to output hypotheses from a fixed hypothesis class (which is related to the popular theme of "explainability" in machine learning).

In conclusion, while there has been progress in studying separations in the setting where the learner is constrained to output hypotheses from a fixed hypothesis class, there is still much to be discovered. Note that if the hypothesis class is BQP-complete in the sense that it can perform arbitrary quantum computation, then a collapse similar to Lemma 3 happens and no separations are possible. All in all, we have yet to find an example of a learning setting where the data is generated by a genuine quantum process and where it is necessary to use a quantum algorithm to efficiently identify the process generating the data.