Perspective

Is Quantum Advantage the Right Goal for Quantum Machine Learning?

Maria Schuld* and Nathan Killoran
Xanadu, Toronto, Ontario M5G 2C8, Canada

(Received 18 November 2021; revised 16 May 2022; published 14 July 2022)

Machine learning is frequently listed among the most promising applications for quantum computing. This is in fact a curious choice: the machine-learning algorithms of today are notoriously powerful in practice but remain theoretically difficult to study. Quantum computing, in contrast, does not offer practical benchmarks on realistic scales and theory is the main tool we have to judge whether it could become relevant for a problem. In this perspective, we explain why it is so difficult to say something about the practical power of quantum computers for machine learning with the tools we are currently using. We argue that these challenges call for a critical debate on whether quantum advantage and that the narrative of "beating" classical machine learning should continue to dominate the literature in the way it does, and highlight examples for how other perspectives in existing research provide an important alternative to the focus on advantage.

DOI: 10.1103/PRXQuantum.3.030101

I. INTRODUCTION

The average number of papers on the arXiv quantum section that relate to machine learning has increased from a handful of contributions per year in the early 2000s to a few papers *per day* in 2021 [1]. A large share of this literature can be attributed to the field of quantum machine learning, which investigates how quantum computers can be used to solve machine-learning problems [2–6], stemming from both conventional and "quantum" [7,8] data. The dominant goal in quantum machine learning is to show that quantum computers, with their properties such as entanglement and interference, offer advantages for machine-learning tasks of practical relevance [9]. This question is particularly important to the emerging quantum technology industry, which has been driving the discipline right from the start [10,11], and which often names machine learning as one of the core application areas for quantum computers.

In this perspective, we want to put the goal of beating classical machine learning under critical scrutiny and argue that the scale of progress we seek may require at least a partial liberation from the "tunnel vision of quantum advantage." First, in Sec. II we explain why—contrary to commercial expectations—machine learning may turn out to be one of the hardest applications for which to show a practical quantum advantage (see also Table I): (a) machine learning is famous for notoriously powerful algorithms that set a challenging baseline for quantum algorithms; (b) the inputs to training algorithms are increasingly big and therefore hard to handle by early quantum computers; (c) the problems tend to stem from the human domain and are much messier than the tasks solved by standard quantum algorithms; (d) machine-learning theory provides a shifting ground to work with, since past assumptions and intuition is currently being upheaved by *deep learning*; and (e) we only have limited options to practically evaluate our methods with benchmarks. To state it in simple terms, quantum machine-learning research is trying to beat large high-performing algorithms for problems that are conceptually hard to study.

At the same time, the tools that quantum computing offers to think about advantages—essentially, experiments on prototype quantum devices and over 30 years' worth of knowledge on provable asymptotic speed-ups—are severely limited. Consequently, it may only be possible to show that quantum can beat classical machine learning in highly abstract settings or on very small scales at this stage. A focus on quantum advantage therefore means focusing only on a biased subset of models, data sets, and theoretical approaches, namely the ones we can tackle under these difficult conditions—a fact that we should discuss more critically. It is important to be clear that these challenges do not mean that we should stop trying to figure out what quantum computers can offer for machine learning. But judging the value of research from the limited lens of speed-ups could prevent important research

*maria@xanadu.ai

TABLE I. A comparison of typical properties of problems studied in quantum computing versus problems solved by machine-learning algorithms. Looking at this table, it is no surprise that quantum machine learning is a tough candidate for applications with a quantum advantage.

Property	Problems studied in quantum computing	Problems solved by machine learning
Classical performance	*Low*—problems are carefully selected to be provably difficult for classical computers	*High*—machine learning is applied on an industrial scale and many algorithms run in linear time in practice
Size of inputs	*Small*—near-term algorithms are limited by small qubit numbers, while fault-tolerant algorithms usually take short bit strings	*Very large*—may be millions of tensors with millions of entries each
Problem structure	*Very structured*—often exhibiting a periodic structure that can be exploited by interference	"*Messy*"—problems are derived from the human or "real-world" domain and are naturally complex to state and analyze
Theoretical accessibility	*High*—there is a large bias toward problems about which we can theoretically reason	*Shifting*—theory is currently being rebuilt around the empirical success of deep learning
Evaluating performance	*Computational complexity*—the dominant measure to assess the performance of an algorithm is asymptotic run-time scaling	*Practical benchmarks*—machine-learning research puts a strong emphasis on empirical comparisons between methods

areas from emerging and, in the worst case, it may even hinder the innovation needed to find use cases for quantum computers in the future. This is what we argue in Sec. III.

However, if we decide to let go of the goal of beating classical machine learning for a moment, what other meaningful questions can we ask? In Sec. IV, we discuss how existing research already yields rich results without having to only look at quantum advantage. First, we use the discussion around *quantum perceptrons* [12–19] to motivate the question *What are good building blocks for quantum models?* in which "good" can have much more diverse interpretations than speed-ups and beating benchmarks. Second, we explain how the connection between a large class of quantum machine-learning models and kernel methods probes the important question *How can we bridge quantum computing and classical learning theory to gain a better understanding of quantum machine learning?* rather than only finding classically intractable quantum kernels. Third, we use the technique of computing gradients on a quantum computer as an example of a successful subfield that has not been purely driven by an advantage from the very beginning but by the question *How can we make quantum software ready for machine-learning applications?* We believe that all three ingredients—the right quantum models to study, theoretical tools with which we can study them, and software solutions to scale experiments—are important for a meaningful attempt to explore the benefit of quantum computing for machine learning in future. But, somewhat paradoxically, limiting what we deem worth researching in these and other areas by whether or not a paper can demonstrate that "quantum is better" may actually *prevent* us from laying these much-needed foundations.

II. WHY MACHINE LEARNING IS SUCH A CHALLENGING PROBLEM

To be more precise about why machine learning may be a challenging application for the state that quantum computing is in, we have to become a bit more technical and look at how a machine-learning task can actually be formulated as a mathematical problem using the framework of empirical risk minimization. This may contain material that is familiar to some readers but it helps us to make the argument of the following section more explicit.

A. How to formalize learning

Intuitively, learning is the acquisition of skills from examples [20] (some useful textbooks are Refs. [21–23]). In machine learning, computers are the "agents" that learn and examples are represented by *data*. Skills can be as diverse as navigating a physical body in an environment, playing chess, generating artificial images, or translating languages. These situations have been captured by the famous distinction into supervised, unsupervised, and reinforcement learning. It is a bit surprising at first that most of machine-learning theory only focuses on supervised learning—which is not so much a reflection of importance but of the fact that supervision, or the provision of some information on the "ground truth," makes it easier to define what it means for a problem to be solved. At the same time, supervised learning does not have to deal with interaction between the learner and the data, as is common in reinforcement learning.

A rather general version of a supervised learning problem can be stated as follows [24]:

Definition 1 (Supervised learning task). *Consider a suitable data input domain $\mathcal{X}$ and a label domain $\mathcal{Y}$, as well*

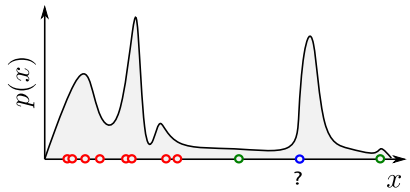

FIG. 1. A central problem in machine learning is how to find a model that performs well with regard to a distribution $p(x)$ over data points x if only a small set of samples from that distribution is given. In supervised learning, the data samples are labeled (red and green dots) and the goal is to label a new sample (blue dot). Especially in high dimensions, the samples will not be able to provide information on the entire data space (here, indicated by the region of high density with no samples). Learning is only possible if the distribution, model, and/or model-selection strategy contains a lot of structure, which is not always easy to analyze theoretically.

as a probability distribution $p(x)$ over inputs $x \in \mathcal{X}$. We assume that there is some ground truth mapping $f^* : \mathcal{X} \to \mathcal{Y}$ of inputs to target labels. We are given a finite set of inputs sampled from $p(x)$, together with their target labels $\mathcal{D} = \{(x^1, y^1), \ldots, (x^M, y^M)\}$ with $(x, y) \in \mathcal{X} \otimes \mathcal{Y}$, as well as a loss $l : \mathcal{Y} \otimes \mathcal{Y} \to \mathbb{R}$ that tells us how well a label predicted by a function $f : \mathcal{X} \to \mathcal{Y}$ compares to the target label. The task is to find a model f from a class of model functions $\mathcal{F}$ that minimizes the expected loss over the data distribution

$$\hat{f} = \mathrm{argmin}_{f \in \mathcal{F}} \int_{\mathcal{X}} p(x)\, l[f(x), f^*(x)]dx. \tag{1}$$

For image recognition, which has been one of the early success stories of modern machine learning, the inputs are numerical representations of images and the labels could be binary tags that indicate whether the images contain harmful content. The distribution $p(x)$ describes the probability with which we can expect to be given certain images in the problem but is necessarily unknown in a real-life task. Instead, we are given a subset of example images drawn from this hypothetical distribution (see Fig. 1), as well as information on which contain harmful content ($y = 1$) and which do not ($y = 0$). A typical loss function is simply an indicator function [25]

$$l(y, y') = \begin{cases} 1 \text{ if } y \neq y' \\ 0 \text{ else.} \end{cases} \tag{2}$$

Minimization of the expected loss over the data distribution is another way of saying that we want our model f to do well with regard to the loss over all the data that we can expect to see.

We can now pinpoint more precisely why machine-learning applications are so hard to access from a theory point of view: in all but pathological examples, the probability distribution $p(x)$ as well as the target function f^* in Definition 1—and hence an important part of Eq. (1)—is unknown. Even if we could model it, the integral in Eq. (1) would be hard to compute for all but special cases. In other words, even a very basic formalization of machine learning translates to a mathematical problem that is usually unsolvable.

B. Solving the problem in practice

Even though surprisingly few beginners to machine learning are aware of this correspondence, the standard approach to how to deal with this predicament is to solve a *proxy problem* to Definition 1 and hope that it translates well to the original one. The proxy problem is known as *empirical risk minimization* and it prescribes evaluation of the model performance using the finite set of data samples $\mathcal{D}$:

$$\hat{f}_{\mathrm{emp}} = \mathrm{argmin}_{f \in \mathcal{F}} \frac{1}{|\mathcal{D}|} \sum_{(x,y) \in \mathcal{D}} l(f(x), y). \tag{3}$$

Much of learning theory tries to find guarantees on how solving the empirical proxy will *generalize* to the original problem or how solutions found with a finite sample size perform on the original distribution [26].

The performance of a model on unseen data is usually measured on a test set of further data samples that have not been used for training and most papers in the machine-learning literature report the error on the test set by running benchmarks on famous data sets. While this sounds straightforward, it is hard to get high-quality results that do not depend on implementation details. For quantum machine-learning research, parts of which try to adopt the culture of benchmark comparisons, the current hardware size limitations make it an even more challenging tool to use and interpret.

In summary, while an important component of machine learning is optimization, its central aim is generalization, which is nontrivial to formalize and measure—even more so when we want to add quantumness into the mix.

C. Deep learning turns learning theory upside down

Many of the standard tools in machine learning, such as cross-validation and regularization, are trying to fulfill the balancing act of not solving Eq. (3) "too well": we want to use the information provided by the finite data sample but we do not want to pick up its particularities [which may not be present if we are given a different data set $\mathcal{D}$ sampled from $p(x)$]. For example, if all the images in our data set that have a black pixel in one position are, coincidentally, images with harmful content, we do not want to learn the spurious relation that if the pixel is black, the image is harmful.

For the longest time, "picking up too much information" was thought to be identical to interpolating the training data perfectly well (i.e., getting a zero average loss over the data). Examples of mitigation strategies are to choose a simple function class $\mathcal{F}$, to add terms to the loss that penalize nonsmooth models from that class, or to stop iterative optimization before it converges to a minimum. But since more than a decade ago, we have consistently been getting empirical evidence that challenges this assumption: very large models can fit any function perfectly well but still generalize beyond the data used for training—even in the presence of noise. This phenomenon has first been attributed to a kind of hierarchical model called a *deep neural network* but has been observed in other settings as well and is now understood to be a main characteristic of the regime of so-called *deep learning* [27,28].

One of the most important goals in machine-learning research today is to unite the evidence presented by deep learning with learning theory. This is a formidable challenge due to the mathematical structure of neural networks as long sequences of linear and nonlinear transformations, which make them unwieldy for mathematically modelling. Furthermore, it is by now largely uncontested that the algorithm with which neural networks are trained, as well as the data themselves, play crucial roles in the phenomena we observe in deep learning [27,29,30]. Therefore, a viable theory cannot just make statements about the model class $\mathcal{F}$ but has to describe the solutions $\hat{f}$ to an optimization problem, as well as the data distribution p. This means that even the simplest of toy models has to capture many moving parts, each of which is already difficult to analyze in the first place.

This ongoing revolution in machine-learning theory building, as well as the practical success of deep learning itself, obviously poses even more challenges for a theory of quantum machine learning, where we want to add quantum theory as another moving part. At the same time, we only have minimal access to empirical results from "just running the algorithm." And even if few-qubit proof-of-principle circuits can be simulated (or even run on real hardware), the learning regimes we are trying to understand are not observed on these small scales—which means that we cannot say much about the behavior that quantum models will exhibit on a realistic problem scale.

III. A CRITICAL LOOK AT QUANTUM ADVANTAGE

The previous section motivated why machine learning is a challenging problem to improve by quantum computers due to the good performance of existing algorithms, large inputs in many applications, the complex mathematical structure of the basic problems, and the little we know about why the best models perform so well, forcing us to gather evidence by benchmarks rather than guiding it by theory.

In this section (after a short overview of the research field itself), we motivate why in the context of machine learning, the tools we currently use to investigate quantum advantage substantially limit and bias the statements we can make about the practical use of quantum computers.

A. Progress in quantum machine learning

While sporadic papers at the intersection of quantum computing and machine learning have been published since the 1990s [31–39], quantum machine learning—here, defined as research on how to use quantum computers for machine-learning tasks from the classical or quantum domain—only gained momentum in around 2013 (see references in Refs. [2,40]).

Since then, one can distinguish two popular approaches to quantum machine learning. In the first years, a common goal has been to speed up existing machine-learning algorithms by solving (sub)tasks such as matrix inversion [41–43], Gibbs sampling [44,45], singular-value estimation [46], or search [12,47] on a quantum computer. Since this agenda is pretty much borrowed from the *modus operandi* of traditional quantum computing, it may not be surprising that the studies are firmly rooted in this parent discipline and touch upon the intricacies of machine-learning research only in the most basic strokes.

The advent of near-term quantum computers has led to a growing popularity of the second approach, which considers *parametrized* or *variational* quantum circuits as machine-learning models [48–51]. In these proposals, training is done similarly to neural networks: gradient-descent-type algorithms iteratively find better physical parameters of the "quantum model." Central questions in this branch of research are what architectures to choose [52,53] and how to compute gradients [54,55], as well as the trainability [56,57], expressivity [58,59] and generalization power [60–63] of such models using insights from machine learning.

Apart from these two active fields of research, there are many other contributions that try to formulate quantum versions of classical learning problems and analyze their scaling. For example, we can ask how quantum data distributions change the sample complexity of learning [8,31,64], how classification problems change in a quantum setting [63,65,66], how quantum agents learn from interacting with an environment [67,68], or how quantum Ising models compare to Ising-based machine-learning models such as Boltzmann machines [11] or Hopfield networks [69].

B. Quantum advantage

Almost all branches of quantum machine-learning research have been heavily framed by the question of

"beating" classical machine learning in some figure of merit, such as:

(a) The asymptotic run time of a particular machine-learning algorithm; for example, an optimizer used to solve the empirical risk-minimization problem in Eq. (3) [41–43,46]
(b) Whether or not a learning problem (such as the one in Definition 1) is efficiently solvable for a particular data distribution $p(x)$ [70,71]
(c) The expressivity of a model class $\mathcal{F}$ [59,72]
(d) The number M of samples needed to learn [64,73]
(e) Average or worst-case generalization errors (which measure the difference between expected and empirical loss) [60–62]
(f) The structure of the optimization landscape, giving us an idea of how easy it is to solve Eq. (3) with gradient-based methods [56,57]
(g) The test error on some small-scale practical benchmark [7,74,75].

The positive publication bias known from other areas of science [76] is strongly prevalent in most areas of quantum machine learning and a number of "positive" results have been put forward that either "prove" theoretically or "show" empirically that quantum computers are better at something. A few examples of the typical phrasing in abstracts and introductions are as follows:

(a) "[W]e establish a rigorous quantum speed-up for supervised classification" [70]
(b) "We prove that [$\cdots$] quantum machines can learn from exponentially fewer experiments than those required in conventional experiments" [73]
(c) "[W]e prove that [parametrized quantum circuits] with a simple structure already outperform any classical neural network for generative tasks" [72]
(d) "[F]or achieving accurate prediction on all inputs, we prove that exponential quantum advantage is possible" [8]
(e) "We show that our quantum-inspired generative models [..] generalize to unseen candidates with lower cost function values than any of the candidates seen by the classical solvers" [77]
(f) "Our simulation results show that our quantum-inspired models have up to a 68$\times$ enhancement in generating unseen [..] samples compared to GANs" (where GAN refers to "general adversarial network") [74]

Some areas, most notably the trainability of quantum models and the sample complexity in certain learning frameworks, actively discuss results that show which approaches do *not* lead to advantages or that could be problematic for practical quantum machine learning (ML):

(a) "We prove that for any input distribution [$\cdots$], a classical ML model can provide accurate predictions on average by accessing [the quantum process generating data] a number of times comparable to the optimal quantum ML model" [8]
(b) "[D]espite concerns about gradient-based methods in classical deep neural networks [$\cdots$], they are successful [$\cdots$ meanwhile, we show that for a large class of random circuits, the average value of the gradient of the objective function is zero" [56]
(c) "Our main result is that quantum and classical sample complexity are in fact equal up to constant factors in both the PAC and agnostic models" (where PAC refers to "probably approximately correct") [64].

If so much progress is being made in understanding quantum advantage, why do we think there is a problem? Will this research not eventually narrow down on areas where quantum computers could have a practical impact on machine-learning applications—or show by overwhelming evidence that the case is hopeless? We believe that there is a deeper structural issue: the tools we currently have in quantum computing are not sufficient to make meaningful statements about this question. Let us motivate this statement with a few points.

Proving exponential speed-ups for artificially constructed settings [50,71], on the one hand, is interesting from an academic point of view but does not say much about possible quantum applications. In the language of machine learning, this approach picks problems that have a heavy bias in the data distribution $p(x)$ and/or the ground truth of the problem of what quantum computers solve well—or, as remarked in Ref. [78], "quantum machine-learning models can offer speed-ups only if we manage to encode knowledge about the problem at hand into quantum circuits, while encoding the same bias into a classical model would be hard." But as we explain in the previous section, the success of machine learning does not stem from solving a structured well-understood problem and hand coding it into the solution method. On the contrary, machine learning is famous for being agnostically applied to a range of problems for which we do not know the exact inductive bias that would suit the data. Furthermore, the principle of *no free lunch* [79] in machine learning states that for any algorithm performing well on one problem, there will always be another problem on which it does not perform well. Good performance on selected hand-crafted examples therefore does not tell us anything about quantum computers as general learning tools. Note that relaxation of the need for exponential speed-ups can lead to provable advantages in more general settings (see, e.g., Refs. [67,80]) but it is currently questioned whether such advantages will have an impact considering

the overhead of error correction in fault-tolerant quantum computers [81].

The "traditional approach" to quantum machine learning mentioned in the previous section follows a different logic and looks for exponential speed-ups to widely applicable algorithms such as support vector machines and neural nets [42,43]. Since these algorithms evidently *have* efficient run times, the goal is usually to reach sublinear scaling. Here we find another issue, namely that we need extreme assumptions about data loading and readout settings, and fair comparisons to classical models have been challenged in the past [82,83].

Another potential issue of proving or disproving whether quantum machine learning "works" is the tendency to make statements about average or worst-case properties of extremely large model families, such as the class of models that we can express as $f(x) = \mathrm{tr}\{\rho(x)M\}$ (where ρ is a quantum state depending on x and M is any observable) or models constructed from circuits sampled according to the Haar measure [56,62,78]. Such statements do not preclude a more specific subclass of quantum models from having entirely different statistical properties. As a comparison, we may be able to prove that *all* models we can express on a classical computer have certain average or even worst-case properties for learning, which does not prevent *specific* models such as boosting or GANs from performing in an entirely different manner.

Empirical studies, on the other hand, tend to compare with very specific classical models on (necessarily) small data sets [49,61] and it is consequently hard to tell if advantages are due to the careful selection of the hyperparameters, benchmarks, and comparisons, or if they are structural observations. Small changes in the—often *ad hoc* designed—architecture of the circuits can cause the results to vary significantly [75]. Only a few studies try to reproduce existing results [84] or critically ask what measures we should apply [74] in our benchmarks other than borrowing concepts from classical machine learning. We also know very little about the scaling of empirical results to larger problem sizes, which will still be a challenge for experiments in years to come.

In our view, the question about whether quantum computers can really play a role in identifying *practical* machine-learning applications is therefore still wide open and is unlikely to be decided by theoretical proofs or small-scale experiments. These tools should be considered more as a means of fostering our understanding and test hypotheses in a well-defined setting. This is very relevant at the current state of quantum machine learning, where we observe an increasing tone of resignation in informal conversations with colleagues and students as quantum machine learning fails to produce immediate commercial use cases. The frequently repeated solution is to discard quantum computers for classical data processing [78] and instead see the future of quantum machine learning in analyzing data in the form of quantum states [7,73]. But following the thoughts laid out in this section, we should ask ourselves if switching our attention to "quantum data" is subconsciously motivated by the hope that it suits our traditional proof techniques better, rather than providing a mature use case.

IV. ALTERNATIVE RESEARCH AGENDAS

In acknowledging the current difficulty of proposing quantum algorithms that improve the performance of machine learning, we do not mean that quantum machine-learning research is at a dead end. Quite the contrary—recent years have shown a lot of interesting and fruitful research areas that have increased our understanding of the intersection without focusing on advantages only. We now want to illustrate this with three examples. The first two examples—the search for a quantum perceptron and the link between quantum circuits and kernel methods—show how a research area can or has been framed from both an "advantage" and "nonadvantage" perspective; either approach leads to different kinds of investigations that can mutually benefit from each other. The third example, the training of quantum circuits using gradients and automatic differentiation software, highlights an area that has enabled quantum applications research without directly trying to improve classical algorithms in the first place.

A. Quantum perceptrons or the search for building blocks of quantum models

A perceptron [85] is a simple function

$$f(\mathbf{x}) = \varphi(\mathbf{w}^T\mathbf{x}), \tag{4}$$

where $\mathbf{x}$ is an input vector, $\mathbf{w}$ is a vector of trainable weights, and φ is a nonlinear scalar function. The perceptron has a long history that connects machine learning with biological models of the brain. It is the basic building block of neural networks and hence of most of the modern deep-learning models used in practice today. Ways of constructing quantum versions of perceptrons have sparked the imagination of researchers for more than 25 years and quantum machine learning consequently contains a huge variety of proposals (see, e.g., Refs. [12,13,16–19,86–89], to mention just a few).

Implicitly, quantum perceptrons are motivated by the success of classical perceptrons and the desire to port this success over to the quantum domain. Depending on whether or not we want to prove a quantum advantage, very different study designs emerge. An advantage focus would require a comparison of quantum and classical versions with respect to run time or performance in learning tasks. The design would have to focus on enabling this

advantage (a feat that, to our knowledge, has not been convincingly performed yet).

Shifting the motivation to other figures of merit allows us to shed a different light on the search for a quantum perceptron. As is done in many studies, we could ask what the most natural equivalent of a nonlinear activation function would be in quantum algorithms. But we could also try to find an efficiently trainable unit for quantum machine-learning models that quantum hardware can easily implement. Other figures of merit are the simplicity of the model to allow theoretical investigations into training and generalization behavior or whether it allows us to pinpoint "nonclassicality" or "quantumness," so that we can directly study its influence on learning. All these alternative figures of merit lead to very different design choices.

We want to remark in passing that without critical reflection, the role of a universal building block has been filled by the ubiquitous Pauli rotations that we are so used to from quantum computing textbooks. But are we able to do better? Is there another "unit" that can provide a playground for theoretical insight and direct us toward the right practical implementations, such as the Ising model did for many-body physics [90] or linear models for deep learning [91,92]? Ironically, this change of perspective is not unlike the development of the perceptron itself: while researchers originally wanted to mimic a powerful concept for learning, namely the brain, porting it over to the computational domain required finding the right abstraction rather than emulating the original. Likewise, quantum researchers are trying to mimic the perceptron, which has proven to be a powerful concept in classical machine learning, but it may turn out that rather than emulation, we ought to distill the crucial properties of this model to make it suitable for the quantum computing domain.

B. Quantum kernels as a bridge between quantum computing and learning theory

The second example we want to bring forward is that of quantum kernel methods. The research area of quantum machine learning has grown out of the realization that data encoding is what machine-learning researchers call a "feature map" [42,93], which means that many quantum circuits can be understood as a linear model in a feature space of the data [94,95]. Again, part of this research area has been framed by (and used for) the search for quantum advantage [60,70,95]. But there is a complementary angle: we can see this research area as an attempt to find formal connections between quantum and machine-learning theory, connections that help us to apply results from one field to the other (see also Ref. [96]). We want to briefly introduce the basic concepts of quantum kernel research (see Fig. 2) to compare these two angles in more detail.

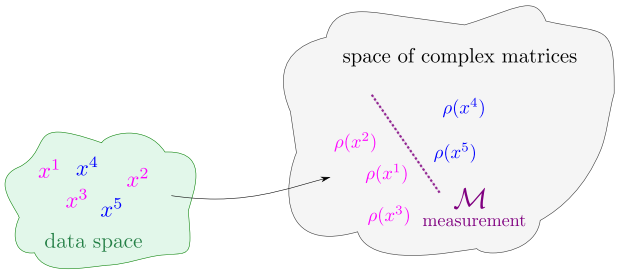

FIG. 2. Many quantum circuits used as supervised machine-learning models can be understood as mapping data to quantum states and then distinguishing these quantum states via hyperplanes defined by measurement observables. Such linear models in high-dimensional spaces are known as kernel methods in classical machine learning and connect quantum machine learning to a rich set of tools to analyze optimization, learning, and generalization.

In a nutshell, quantum kernel research is based on the insight that if we encode a data input $x \in \mathcal{X}$ into a quantum state $\rho(x)$ (e.g., via a quantum state preparation routine), the expectation of an observable $\mathcal{M}$ can be interpreted as a machine-learning model of the form

$$f(x) = \mathrm{tr}\{\rho(x)\mathcal{M}\}. \tag{5}$$

Upon realizing that the trace is an inner product (known as the Hilbert-Schmidt inner product) in the space of complex-valued matrices and that $\rho(x)$ maps the input x into this space, we can state that the "quantum model" in Eq. (5) is a *linear model* of the form

$$f(x) = \langle \phi(x), w \rangle_{\mathcal{H}}, \tag{6}$$

where $\phi(x)$ is a *feature map* from the data space to a feature space $\mathcal{H}$, w is a weight vector, and $\langle \cdot, \cdot \rangle$ is the inner product in $\mathcal{H}$. Most often, $\mathcal{H}$ is simply $\mathbb{R}^N$. The weight vector then contains trainable parameters and defines a linear hyperplane that can be used to separate classes of data in a supervised learning problem. Likewise, in many variational quantum models, $\mathcal{M} = \mathcal{M}(\theta)$ from Eq. (5) is trainable: by optimizing a parametrized circuit before a fixed measurement, we effectively choose a measurement basis (and hence the discriminating hyperplane) via optimization.

This innocent link has immense consequences. Linear models in high-dimensional spaces are the core of one of the richest corners of machine-learning theory, namely kernel theory, which we can now use to understand "quantum models" [96]. For example, kernel theory tells us that quantum models of the form given in Eq. (5) can be rewritten as a linear combination of the "distances" between quantum states encoding the training data points x^m and the

quantum state encoding the input x that we seek to classify,

$$f(x) = \sum_{m=1}^{M} a_m \mathrm{tr}\{\rho(x)\rho(x^m)\},\ x_m \in \mathcal{D}, a_m \in \mathbb{R}. \quad (7)$$

Instead of learning the parameters θ in a variational circuit, we can learn the coefficients a_m and we only need the quantum computer to evaluate the trace term (which for pure states reduces to the overlap $|\langle\psi(x)|\psi(x^m)\rangle|^2$). Furthermore, we are guaranteed that the optimal coefficients a_m construct a model that is also the global minimum of the empirical risk-minimization problem in Eq. (3). In other words, while Eq. (7) may define a smaller function class compared to Eq. (5), it still contains the solution that we want to find. If the loss used to compare predictions with target labels is convex, the entire optimization problem is convex and hence conceptually simple to analyze. This also guarantees that we can find the optimal solution.

An understanding of quantum computers as "kernel evaluators" can tell us something about quantum advantage. In situations where this link holds, potential speedups have to be located in the evaluation of the kernel $\mathrm{tr}\{\rho(x), \rho(x')\}$ [60]. We can investigate kernels based on circuits that are believed to be classically intractable [95] and prove end-to-end quantum advantages for very specific learning problems [70]. At the same time, the necessity of estimating the value of the kernel function using finite shots introduces an overhead [97,98]. While this research is certainly valuable, a classically intractable kernel that is *useful* for practical machine-learning tasks has yet to be found.

On the other hand, we can view quantum kernel theory purely as a tool for theory building. For example, the link connects quantum circuits to linear representations of neural networks such as neural tangent kernels [99] and random Fourier features [100], which are central to current investigations of deep learning—a fact that has been explored in a series of recent papers [101–103]. The theory of kernel methods also allows us to study generalization by making statements about the margin between the data-encoding states $\rho(x)$ for two different classes of data [63] or the regularization properties of a model [104]. Finally, it allows us to port over insights from quantum state discrimination as to what constitutes optimal decision boundaries [105]. None of these studies directly tries to answer the question of whether or not quantum computers could be superior for learning and they lead to very different kinds of results.

There are many similar points of contact between quantum and machine-learning theory, such as the interpretation of quantum measurements as samples from a generative model [106] or the proximity of quantum computers to machine-learning models inspired by many-body-physics [11] and the usefulness of neural networks in representing quantum states [107].

C. Quantum gradients and making quantum software ready for machine-learning applications

The last example highlights an area of research that has massively increased our capability of performing experiments and building software around quantum machine learning without having quantum advantages as an immediate goal. It is the study of gradients of quantum computations and of how to retrieve them from performing other efficient quantum computations. In fact, it is well known that so-called parameter-shift rules [54,55] put forward for this task are *less* efficient than classical backpropagation, since they require a full model estimation per model parameter (while each estimation requires many shots or runs of the model circuit).

Historically, the dominant representation of quantum computations has involved static algorithms that have been hand designed by expert theorists to maximally leverage coherent effects. More recently, there has been a growing recognition that the addition of free parameters to quantum circuits allows them to represent an entire family of functions, while retaining the unique coherence properties that make quantum algorithms distinct [108]. The best value of these parameters for a particular task can then be determined variationally. This expansion makes it easier for researchers to quickly test out new ideas and discover new quantum algorithms—as evidenced by the recent explosion of works on variational quantum circuits—but comes with the caveat that such classes of circuits may be harder to pin down theoretically (compared to, e.g., the kernel methods discussed in Sec. IV B). Notably, this dichotomy mirrors the present-day situation in deep learning.

In the variational framework, a quantum circuit implements a function of the form [109]

$$f(x,\theta) = \mathrm{tr}\{\rho(x,\theta)\mathcal{M}(\theta)\}, \quad (8)$$

where, in contrast to Eq. (6), we include the free parameters θ in the measurement and also allow for a trainable state $\rho(x,\theta)$. Typically, the free parameters correspond to rotation angles of gates in a quantum circuit. This presents us with a new task: given a parametrized circuit, how should we adjust the parameter values to "train" the circuit to minimize some loss function l that measures the quality of $f(x,\theta)$?

While many options are available for training, there are very intriguing links with the workhorse algorithm used to train deep-learning models: gradient descent. In gradient descent, we optimize a loss function by computing its gradient with respect to the free parameters and iteratively updating the parameters in the direction of the gradient. From the chain rule, we must therefore determine the gradient of the model function, $\nabla_\theta f\ (x,\theta)$, with respect

to the free parameters θ of the circuit. Modern software tools such as TensorFlow [110] or PyTorch [111] largely automate the gradient computation of deep-learning models using the back-propagation algorithm [112]. These libraries even let a user optimize custom functions—such as an expectation value produced from calling quantum computing hardware—provided that the user also supplies the gradient of this function.

Drawing on insights originally developed from quantum optimal control [113], it turns out to be remarkably simple to compute the gradients of (many) quantum circuits. Using a technique now known as the parameter-shift rule [54,55], we can evaluate the derivatives $\partial f/\partial\theta_i$ of a parametrized circuit [114]—and hence the gradient as well—by running the same circuit with parameter θ_i shifted forward and backward by a fixed amount,

$$\frac{\partial f}{\partial \theta_i} = \frac{f(x,\theta + s\hat{e}_i) - f(x,\theta - s\hat{e}_i)}{2\sin(s)}. \tag{9}$$

This technique, which has since been generalized to more and more cases [115–122], has a similar form to the numerical finite-difference approximator but in fact provides an analytically exact expression [123] for any shift value $s \neq 0, \pi$. And although it does not match the efficiency of back-propagation [124], the simplicity of the parameter-shift rule makes it a very hardware-friendly mechanism for computing quantum circuit gradients.

Parameter-shift rules, quantum gradients, and the resulting surge in quantum software for automatic differentiation are a prime example of research that strives by *enabling* quantum machine-learning applications, rather than demanding superiority of quantum algorithms. Armed with the ability to evaluate quantum models and compute their gradients, we can directly "plug and play" with existing deep-learning tools and train quantum circuits in the same way as we train neural networks. We can connect differentiable quantum subroutines into larger hybrid quantum-classical models and train the whole pipeline end to end using any of the specialized gradient-based optimizers developed in deep learning, such as Momentum or Adam [125] (see Fig. 3). Finally, the links unveiled through the study of quantum gradients and training quantum models open up a rich opportunity for cross-pollination of ideas between quantum computing and deep learning. For example, we have already seen the arrival of "quantum-aware" optimizers [126–129], which tweak ideas from deep learning to make them more native to the quantum setting. On the theory side, we can leverage the latest (admittedly, still evolving) theoretical insights on optimization landscapes and generalizations coming from deep learning and potentially adapt them to better understand phenomena such as barren plateaus [56]. As our understanding increases, ideas and techniques from quantum computing can even find their way back into deep learning. A recent example is the use of tensor-network-based models in place of standard neural networks [130].

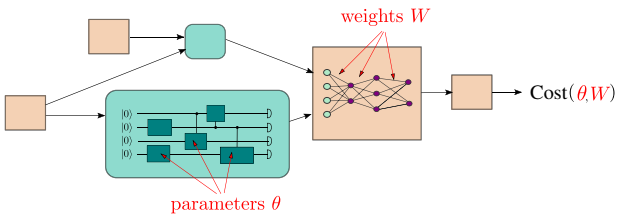

FIG. 3. Parametrized quantum circuits can be trained as parts of larger machine-learning pipelines by making use of automatic differentiation and the fact that we know in many settings how to estimate analytic gradients of cost functions with respect to circuit parameters.

V. MOVING FORWARD

This perspective advocates a shift in the research agenda of quantum machine learning away from investing all our resources into the notion of "beating" classical algorithms. Sections II and III try to motivate such a shift by arguing that the goal of showing quantum advantages forces us to limit our analytical focus to the very few problems we can actually study in a setting as complex as machine learning, while Sec. IV showcases existing areas framed by alternative research questions. Until quantum computers become available to do large-scale benchmarking, asking more fundamental questions may be a very good use of our time but requires a bit of courage to withstand the narrative of trying to find the billion-dollar quantum "supremacy," or to resist catchy expressions such as "the power of deep quantum neural networks."

A paradigm shift is never easy and will require the community to make subtle but crucial adjustments; for example, to the way in which supervisors guide students, how science journalists portray the topic, how companies formulate their deliverables, and how reviewers judge publication worthiness. However, in the end this may be exactly what is needed to push quantum machine-learning research to the level that leads to future industrial-scale applications.

[1] This observation is based on an arXiv API search in the *quant-ph* category for papers using the term "learn" as substring in title or abstract. A similar trend is found for usage of the term "neural network."

[2] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, 2014).

[3] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[4] C. Ciliberto, M. Herbster, A. D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, Quantum machine learning: A classical perspective, Proc. R. Soc. A: Math., Phys. Eng. Sci. **474**, 20170551 (2018).

[5] V. Dunjko and P. Wittek, A non-review of quantum machine learning: Trends and explorations, Quantum Views **4**, 32 (2020).

[6] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers* (Springer, 2021).

[7] L. Schatzki, A. Arrasmith, P. J. Coles, and M. Cerezo, Entangled datasets for quantum machine learning (2021), arXiv preprint ArXiv:2109.03400.

[8] H.-Y. Huang, R. Kueng, and J. Preskill, Information-theoretic bounds on quantum advantage in machine learning (2021), arXiv preprint ArXiv:2101.02464.

[9] A. Perdomo-Ortiz, M. Benedetti, J. Realpe-Gómez, and R. Biswas, Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers, Quantum Sci. Technol. **3**, 030502 (2018).

[10] H. Neven, V. S. Denchev, G. Rose, and W. G. Macready, Training a binary classifier with the quantum adiabatic algorithm (2008), arXiv preprint ArXiv:0811.0416.

[11] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann machine, Phys. Rev. X **8**, 021050 (2018).

[12] A. Kapoor, N. Wiebe, and K. Svore, in *Advances In Neural Information Processing Systems* (2016), p. 3999.

[13] M. Andrecut and M. Ali, A quantum perceptron, Int. J. Mod. Phys. B **16**, 639 (2002).

[14] E. Torrontegui and J. J. García-Ripoll, Unitary quantum perceptron as efficient universal approximator, EPL (Europhys. Lett.) **125**, 30004 (2019).

[15] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, Continuous-variable quantum neural networks, Phys. Rev. Res. **1**, 033063 (2019).

[16] M. Schuld, I. Sinayskiy, and F. Petruccione, Simulating a perceptron on a quantum computer, Phys. Lett. A **379**, 660 (2015).

[17] T. Espinosa-Ortega and T. Liew, Perceptrons with Hebbian Learning Based on Wave Ensembles in Spatially Patterned Potentials, Phys. Rev. Lett. **114**, 118101 (2015).

[18] F. Benatti, S. Mancini, and S. Mangini, Continuous variable quantum perceptron, Int. J. Quantum Inf. **17**, 1941009 (2019).

[19] R. Bondesan and M. Welling, Quantum deformed neural networks (2020), arXiv preprint ArXiv:2010.11189.

[20] See Ref. [131] for an interesting debate that is challenging modern machine learning, arguing that learning is the *efficient* acquisition of skills from examples—rather than just fitting massive models to massive data sets.

[21] C. M. Bishop and N. M. Nasrabadi, *Pattern Recognition and Machine Learning* Vol. 4 (Springer, New York, 2006).

[22] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning* Vol. 112 (Springer, New York, 2013).

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

[24] There are many different ways to formalize the notion of learning and, as usual in science, there have been strong trends in what machine-learning research has considered to be a "relevant" setting. For example, in the past, *learning from membership queries* [31], where we can actively influence which data we are given, has often been considered. A popular alternative flavour to supervised learning as we set it up here is *probably approximately correct (PAC) learning*, where learning translates to finding a model so that with a high probability, the loss of an input x sampled from $p(x)$ has a loss smaller than a threshold.

[25] Note that optimization algorithms usually require continuous loss functions and that standard losses such as least squares can be understood as continuous surrogates for the indicator function.

[26] It may not come as a surprise that the tools and terms of machine learning are largely borrowed from statistics.

[27] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, Understanding deep learning (still) requires rethinking generalization, Commun. ACM **64**, 107 (2021).

[28] P. L. Bartlett, A. Montanari, and A. Rakhlin, Deep learning: A statistical viewpoint (2021), arXiv preprint ArXiv:2103.09177.

[29] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, The implicit bias of gradient descent on separable data, J. Mach. Learn. Res. **19**, 2822 (2018).

[30] S. Arora, N. Cohen, W. Hu, and Y. Luo, Implicit regularization in deep matrix factorization, Adv. Neural Inf. Process. Syst. **32**, 7413 (2019).

[31] R. A. Servedio and S. J. Gortler, Equivalences and separations between quantum and classical learnability, SIAM J. Comput. **33**, 1067 (2004).

[32] S. C. Kak, Quantum neural computing, Adv. Imaging Electron Phys. **94**, 259 (1995).

[33] G. Bonnell and G. Papini, Quantum neural network, Int. J. Theor. Phys. **36**, 2855 (1997).

[34] D. Ventura and T. Martinez, Quantum associative memory, Inf. Sci. (Ny) **124**, 273 (2000).

[35] C. A. Trugenberger, Quantum pattern recognition, Quantum Inf. Process. **1**, 471 (2002).

[36] R. Schützhold, Pattern recognition on a quantum computer, Phys. Rev. A **67**, 062311 (2003).

[37] R. Bonner and R. Freivalds, in *Quantum Computation and Learning* (2003), p. 106.

[38] E. Aïmeur, G. Brassard, and S. Gambs, in *Advances in Artificial Intelligence* (Springer, 2006), p. 431.

[39] N. H. Bshouty and J. C. Jackson, Learning DNF over the uniform distribution using a quantum example oracle, SIAM J. Comput. **28**, 1136 (1998).

[40] M. Schuld, I. Sinayskiy, and F. Petruccione, Introduction to quantum machine learning, Contemp. Phys. **56**, 172 (2015).

[41] N. Wiebe, D. Braun, and S. Lloyd, Quantum Algorithm for Data Fitting, Phys. Rev. Lett. **109**, 050505 (2012).

[42] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum Support Vector Machine for Big Data Classification, Phys. Rev. Lett. **113**, 130503 (2014).

[43] A. Zlokapa, H. Neven, and S. Lloyd, A quantum algorithm for training wide and deep classical neural networks (2021), arXiv preprint ArXiv:2107.09200.

[44] N. Wiebe, A. Kapoor, and K. M. Svore, Quantum deep learning (2014), ArXiv:1412.3489v1.

[45] M. Denil and N. De Freitas, in *NIPS 2011 Deep Learning and Unsupervised Feature Learning Workshop* (2011).

[46] I. Kerenidis and A. Prakash, Quantum machine learning with subspace states (2022), arXiv preprint ArXiv:2202.00054.

[47] E. Aïmeur, G. Brassard, and S. Gambs, Quantum speed-up for unsupervised learning, Mach. Learn. **90**, 261 (2013).

[48] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors (2018), arXiv preprint ArXiv:1802.06002.

[49] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, Circuit-centric quantum classifiers (2018), arXiv preprint ArXiv:1804.00633.

[50] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit born machines, Phys. Rev. A **98**, 062324 (2018).

[51] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

[52] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[53] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[54] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[55] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[56] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 1 (2018).

[57] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting ansatz expressibility to gradient magnitudes and barren plateaus (2021), arXiv preprint ArXiv:2101.02138.

[58] Y. Wu, J. Yao, P. Zhang, and H. Zhai, Expressivity of quantum neural networks, Phys. Rev. Res. **3**, L032049 (2021).

[59] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[60] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, Nat. Commun. **12**, 1 (2021).

[61] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nat. Comput. Sci. **1**, 403 (2021).

[62] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data (2021), arXiv preprint ArXiv:2111.05292.

[63] L. Banchi, J. Pereira, and S. Pirandola, Generalization in quantum machine learning: A quantum information perspective (2021), arXiv preprint ArXiv:2102.08991.

[64] S. Arunachalam and R. de Wolf, Optimal quantum sample complexity of learning algorithms, J. Mach. Learn. Res. **19**, 2879 (2018).

[65] G. Sentís, J. Calsamiglia, R. Muñoz-Tapia, and E. Bagan, Quantum learning without quantum memory, Sci. Rep. **2**, 1 (2012).

[66] A. Monras, G. Sentís, and P. Wittek, Inductive Supervised Quantum Learning, Phys. Rev. Lett. **118**, 190503 (2017).

[67] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Quantum speedup for active learning agents, Phys. Rev. X **4**, 031002 (2014).

[68] V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-Enhanced Machine Learning, Phys. Rev. Lett. **117**, 130501 (2016).

[69] P. Rotondo, M. Marcuzzi, J. P. Garrahan, I. Lesanovsky, and M. Müller, Open quantum generalisation of Hopfield neural networks, J. Phys. A: Math. Theor. **51**, 115301 (2018).

[70] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **1**, 2021.

[71] R. Sweke, J.-P. Seifert, D. Hangleiter, and J. Eisert, On the quantum versus classical learnability of discrete distributions, Quantum **5**, 417 (2021).

[72] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, Expressive power of parametrized quantum circuits, Phys. Rev. Res. **2**, 033125 (2020).

[73] H.-Y. Huang, M. Broughton, J. Cotler, S. Chen, J. Li, M. Mohseni, H. Neven, R. Babbush, R. Kueng, and J. Preskill, *et al.*, Quantum advantage in learning from experiments (2021), arXiv preprint ArXiv:2112.00778.

[74] K. Gili, M. Mauri, and A. Perdomo-Ortiz, Evaluating generalization in classical and quantum generative models (2022), arXiv preprint ArXiv:2201.08770.

[75] P. Selig, N. Murphy, A. Sundareswaran, D. Redmond, and S. Caton, in *2021 International Conference on Rebooting Computing (ICRC)* (IEEE, 2021), p. 24.

[76] A. Mlinarić, M. Horvat, and V. Šupak Smolčić, Dealing with the positive publication bias: Why you should really publish your negative results, Biochem. Med. (Zagreb) **27**, 447 (2017).

[77] J. Alcazar and A. Perdomo-Ortiz, Enhancing combinatorial optimization with quantum generative models (2021), arXiv preprint ArXiv:2101.06250.

[78] J. Kübler, S. Buchholz, and B. Schölkopf, The inductive bias of quantum kernels, Adv. Neural Inf. Process. Syst. **34**, 1 (2021).

[79] D. H. Wolpert, The lack of a priori distinctions between learning algorithms, Neural Comput. **8**, 1341 (1996).

[80] G. H. Low, T. J. Yoder, and I. L. Chuang, Quantum inference on Bayesian networks, Phys. Rev. A **89**, 062315 (2014).

[81] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, Focus beyond Quadratic Speedups for Error-Corrected Quantum Advantage, PRX Quantum **2**, 010103 (2021).

[82] S. Aaronson, Read the fine print, Nat. Phys. **11**, 291 (2015).

[83] E. Tang, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019), p. 217.

[84] M. Franz, L. Wolf, M. Periyasamy, C. Ufrecht, D. D. Scherer, A. Plinge, C. Mutschler, and W. Mauerer, Uncovering instabilities in variational-quantum deep Q-networks (2022), arXiv preprint ArXiv:2202.05195.

[85] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain., Psychol. Rev. **65**, 386 (1958).

[86] M. Lewenstein, Quantum perceptrons, J. Mod. Opt. **41**, 2491 (1994).

[87] Y. Cao, G. G. Guerreschi, and A. Aspuru-Guzik, Quantum neuron: An elementary building block for machine learning on quantum computers (2017), arXiv preprint ArXiv:1711.11240.

[88] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, Nat. Commun. **11**, 1 (2020).

[89] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, An artificial neuron implemented on an actual quantum processor, npj Quantum Inf. **5**, 1 (2019).

[90] A. Taroni, 90 years of the Ising model, Nat. Phys. **11**, 997 (2015).

[91] P. L. Bartlett, P. M. Long, G. Lugosi, and A. Tsigler, Benign overfitting in linear regression, Proc. Nat. Acad. Sci. **117**, 30063 (2020).

[92] A. M. Saxe, J. L. McClelland, and S. Ganguli, Exact solutions to the nonlinear dynamics of learning in deep linear neural networks (2013), arXiv preprint ArXiv:1312.6120.

[93] R. Chatterjee and T. Yu, Generalized coherent states, reproducing kernels, and quantum support vector machines (2016), arXiv preprint ArXiv:1612.03713.

[94] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[95] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[96] M. Schuld, Supervised quantum machine learning models are kernel methods (2021), arXiv preprint ArXiv:2101.11020.

[97] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, The complexity of quantum support vector machines (2022), arXiv preprint ArXiv:2203.00031.

[98] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, Machine learning of high dimensional data on a noisy quantum processor, npj Quantum Inf. **7**, 1 (2021).

[99] A. Rahimi and B. Recht, *et al.*, in *NIPS*, Vol. 3 (Citeseer, 2007), p. 5.

[100] A. Jacot, F. Gabriel, and C. Hongler, Neural tangent kernel: Convergence and generalization in neural networks (2018), arXiv preprint ArXiv:1806.07572.

[101] N. Shirai, K. Kubo, K. Mitarai, and K. Fujii, Quantum tangent kernel (2021), arXiv preprint ArXiv:2111.02951.

[102] K. Nakaji, H. Tezuka, and N. Yamamoto, Quantum-enhanced neural networks in the neural tangent kernel framework (2021), arXiv preprint ArXiv:2109.03786.

[103] J. Liu, F. Tacchino, J. R. Glick, L. Jiang, and A. Mezzacapo, Representation learning via quantum neural tangent kernels (2021), arXiv preprint ArXiv:2111.04225.

[104] I. Steinwart and A. Christmann, *Support Vector Machines* (Springer, New York, 2008), 1st ed.

[105] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning (2020), arXiv preprint ArXiv:2001.03622.

[106] S. Cheng, J. Chen, and L. Wang, Information perspective to probabilistic modeling: Boltzmann machines versus born machines, Entropy **20**, 583 (2018).

[107] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[108] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, and L. Cincio, *et al.*, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[109] In some settings, the input data x might be omitted. In these cases, since no data is present, the problem is more formally an optimization problem and not a machine-learning problem. Because the tools employed are exactly the same, these distinctions are often conflated.

[110] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, *et al.*, in *12th {USENIX} symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016), p. 265.

[111] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, and L. Antiga, *et al.*, PyTorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. **32**, 8026 (2019).

[112] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning representations by back-propagating errors, Nature **323**, 533 (1986).

[113] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid Quantum-Classical Approach to Quantum Optimal Control, Phys. Rev. Lett. **118**, 150503 (2017).

[114] In this case, the parameters should appear as the rotation angles of gates.

[115] G. E. Crooks, Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition (2019), arXiv preprint ArXiv:1905.13311.

[116] L. Banchi and G. E. Crooks, Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule, Quantum **5**, 386 (2021).

[117] J. S. Kottmann, A. Anand, and A. Aspuru-Guzik, A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers, Chem. Sci. **12**, 3497 (2021).

[118] J. G. Vidal and D. O. Theis, Calculus on parameterized quantum circuits (2018), arXiv preprint ArXiv:1812.06323.

[119] A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, Phys. Rev. A **103**, 012405 (2021).

[120] A. F. Izmaylov, R. A. Lang, and T.-C. Yen, Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations (2021), arXiv preprint ArXiv:2107.08131.

[121] O. Kyriienko and V. E. Elfving, Generalized quantum circuit differentiation rules (2021), arXiv preprint ArXiv:2108.01218.

[122] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients (2021), arXiv preprint ArXiv:2107.12390.

[123] In the case of expectation values obtained with a finite number of shots, the parameter-shift rule remains an unbiased estimator for the analytic gradient.

[124] In back-propagation, or automatic differentiation *through* models, we typically only step through the algorithm once forward and once backward—reusing computations cached from the forward pass—to obtain the entire gradient vector. In contrast, each entry in the gradient of a quantum circuit requires additional circuit evaluations on top of the evaluation of $f(x,\theta)$.

[125] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization (2014), arXiv preprint ArXiv:1412.6980.

[126] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).

[127] J. M. Kübler, A. Arrasmith, L. Cincio, and P. J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, Quantum **4**, 263 (2020).

[128] A. Arrasmith, L. Cincio, R. D. Somma, and P. J. Coles, Operator sampling for shot-frugal optimization in variational algorithms (2020), arXiv preprint ArXiv:2004.06252.

[129] M. Ostaszewski, E. Grant, and M. Benedetti, Structure optimization for parameterized quantum circuits, Quantum **5**, 391 (2021).

[130] E. M. Stoudenmire, Learning relevant features of data with multi-scale tensor networks, Quantum Sci. Technol. **3**, 034003 (2018).

[131] F. Chollet, On the measure of intelligence (2019), arXiv preprint ArXiv:1911.01547.