Is quantum advantage the right goal for quantum machine learning?

Maria Schuld and Nathan Killoran
Xanadu, Toronto, ON, M5G 2C8, Canada[*]

(Dated: March 4, 2022)

Machine learning is frequently listed among the most promising applications for quantum computing. This is in fact a curious choice: Today's machine learning algorithms 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 the narrative of "beating" classical machine learning should continue to dominate the literature the way it does, and provide a few examples for alternative research questions.

The average number of papers on the arXiv's 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, stemming from both "classical" or "quantum" data. The dominant goal in quantum machine learning is to show that quantum computers, with their properties like entanglement and interference, offer advantages for machine learning tasks of practical relevance. This question is particularly important to the emerging quantum technology industry, which has been driving the discipline right from the start[2], 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 a liberation from the "tunnel vision of quantum advantage". First, in Section II we explain why – contrary to commercial expectations – machine learning may turn out to be one of the hardest applications to show a practical quantum advantage for (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, armed with a few prototype quantum devices and thirty year's worth of computational complexity theory.

These challenges do not mean we should stop trying to figure out what quantum computers can offer for machine learning. But they suggest that at this stage, showing that quantum can beat classical machine learning in some general sense may only be possible in highly abstract settings or on very small scales – a fact that we should discuss more critically. Focusing on quantum advantage therefore means focusing only on a biased, limited subset of models, datasets and theoretical approaches, namely the ones we can tackle under these difficult conditions. This could prevent important research 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.

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 Section III we discuss three alternative research goals and illustrate each with an example. First, we use the discussion around *quantum neural networks* and *quantum perceptrons* to motivate the question *What are good building blocks for quantum models?*. Second, we explain how the connection between a large class of quantum machine learning models and support vector machines or kernel methods allows us to probe the question *How can we bridge quantum computing and classical learning theory to gain a better understanding of quantum machine learning?*. Thirdly, we use the technique of computing gradients on a quantum computer to illustrate the question *How can we make quantum software ready for machine learning applications?*. Such alternative agendas have already proven to unlock further research – interestingly, a lot of it being research that searches for quantum advantages.

[*] maria@xanadu.ai

[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] Some of the earliest papers in quantum machine learning stem from research teams at *Google* [1] and *D-Wave* [2].

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 naturally complex to state and analyse
theoretical accessibility	**high** – there is a large bias towards problems about which we can theoretically reason	**shifting** – theory is currently been re-built around the empirical success of deep learning
evaluating performance	**computational complexity** – the dominant measure to assess the performance of an algorithm is asymptotic runtime scaling	**practical benchmarks** – machine learning research puts a strong emphasis on empirical comparisons between methods

TABLE I. 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.

I. A BRIEF OVERVIEW OF QUANTUM MACHINE LEARNING

While sporadic papers at the intersection of quantum computing and machine learning were published since the 1990s [3–10], quantum machine learning – here defined as research on how to use quantum computers for machine learning tasks from the classical or quantum physics domain – only gained momentum around 2013 (see references in [11, 12]).

Since then one can distinguish two popular approaches to quantum machine learning. In the first years, a common goal was to speed up existing machine learning algorithms by solving (sub)tasks such as matrix inversion [13–15], Gibbs sampling [16, 17] or search [18, 19] 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 led to a growing popularity of the second approach, which considers *parametrized* or *variational* quantum circuits as machine learning models [20–23]. In these proposals, training is done similarly to neural networks: gradient-descent-type algorithms iteratively find better physical parameters of the "quantum model". One of the central questions in this branch of research is how we can make statements about the trainability [24, 25] and generalization power [26, 27] 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 analyse their scaling. For example we can ask how quantum data distributions change the sample complexity of learning [9, 28, 29], how classification problems change in a quantum setting [30–32], how quantum agents learn from interacting with an environment [33, 34], or how quantum Ising models compare to Ising-based machine learning models such as Boltzmann machines [2] or Hopfield networks [35].

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 computational complexity, the expressivity of model classes, the number of samples needed to learn, a worst-case generalization bound, or the empirical generalization error on some small-scale practical benchmark. The picture that emerges is that of a mixed bag: Empirical studies routinely report that their proposed quantum model with some variation of parametrized ansatz performs "better" than the chosen classical competitor, but it is hard to tell if this is due to the careful selection of the hyperparameters, benchmarks and comparisons, or if it is a structural observation. We also know very little about the scaling of these results to relevant problem sizes, which will still be a challenge for experiments in years to come. On the other hand, a range of provable quantum advantages were reported, but they construct artificial problems from reformulations of known quantum routines [36, 37], or make extreme assumptions about data loading and read-out settings [14, 38, 39], both of which may turn out to have little impact on the future use of quantum computers as a wide-spread technology alternative.

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 will have to become a bit more technical, and look at how a machine learning task can actually be formulated as a mathematical problem.

A. How to formalize learning

Intuitively, learning is the acquisition of skills from examples[3]. 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:[4]:

Definition 1 ***(Supervised learning task)*** *Consider a suitable data input domain $\mathcal{X}$ and a label domain $\mathcal{Y}$, as well as a probability distribution $p(x)$ over inputs $x \in \mathcal{X}$. We assume that there is some ground truth mapping $f^*: \mathcal{X} \rightarrow \mathcal{Y}$ of inputs to target labels. We are given a finite set of inputs sampled from $p(x)$, together with their target labels $\mathcal{D} = \{(x^1, y^1), \dots, (x^M, y^M)\}$ with $(x, y) \in \mathcal{X} \otimes \mathcal{Y}$, as well as a loss $l : \mathcal{Y} \otimes \mathcal{Y} \rightarrow \mathbb{R}$ that tells us how well a label predicted by a function $f : \mathcal{X} \rightarrow \mathcal{Y}$ 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,*

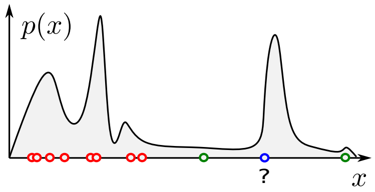

FIG. 1. A central problem in machine learning is how to find a model that performs well with regards to a distribution $p(x)$ over datapoints x if only a small set of samples from that distribution is given. In supervised learning, the data samples are labeled (red and green dots) and the goal is to 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 analyse theoretically.

$$\hat{f} = \min_{f \in \mathcal{F}} \int_{\mathcal{X}} p(x)\; l(f(x), f^*(x)) dx. \quad (1)$$

For image recognition, which was one of the early success stories of modern machine learning, the inputs are numerical representations of images, and the labels could be a binary tag that indicates whether the image contains 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[5]

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

Minimising the expected loss over the data distribution is another way of saying that we want our model f to do well with regards to the loss over all data 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) will 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.

[3] See [40] 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 datasets.

[4] There are many different ways to formalise the notion of learning, and as usual in science, there were strong trends in what machine learning research considered to be a "relevant" setting. For example, in the past *learning from membership queries* [9], where we can actively influence which data we are given, was often considered. A popular alternative flavour to supervised learning as we set it up here is *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.

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

B. Solving the problem in practice

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

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

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.[6]

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 datasets. While this sounds straight-forward, getting high-quality results whose dependence on implementation details are well-controlled is hard. For quantum machine learning research, parts of which try to adopt the culture of benchmark comparisons, the current limitations of hardware size 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 non-trivial 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 fulfil 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 were given a different data set $\mathcal{D}$ sampled from $p(x)$). For example, if coincidentally all images in our data set that have a black pixel in one position are 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 penalise non-smooth models from that class, or to stop iterative optimization before it converges to a minimum. But since more than a decade, we are consistently 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 was first 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* [41, 42].

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 itself, plays a crucial role in the phenomena we observe in deep learning [41, 43, 44]. A viable theory therefore 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 analyse in the first place.

This ongoing revolution in machine learning theory-building, as well as the practical success of deep learning itself, obviously pose 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 have only little 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 behaviour that quantum models will exhibit on a realistic problem scale.

D. Unpacking quantum advantage

Despite these challenges, an impressive amount of progress has been made in the young discipline of quantum machine learning, and we want give a rough idea of how the two approaches sketched in Section I tackle the problem of “beating” classical machine learning.

The traditional approach of speeding up existing machine learning algorithms by quantum subroutines usually tries to speed up the evaluation of the model f, or the optimization algorithm that finds $\min_{f\in\mathcal{F}}$ (or both). As such, this approach does not affect the problem of generalization and learning itself, but only proposes a

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

different implementation of the algorithms that solve it, in the hope that quantum hardware can improve the runtime of the overall algorithm. Since we do not yet know the runtime of quantum algorithms, speed is measured in terms of the *scaling* of the runtime with the size of the inputs, and unless there is a guarantee that the classical algorithm was the best in the first place, such a speedup relates only to the particular algorithm, and not to the learning problem itself. It is also becoming increasingly clear that low-order polynomial speedups may not be enough to counteract the overheads of quantum error correction [45].

A possible reason for the waning popularity of the "speedup" approach in quantum machine learning is the fact that quantum subroutines only guarantee significantly better scaling if we pose very strict assumptions on the data distribution $p(x)$ (for example, that a matrix of data vectors will be low-rank [14] or sparse [13]) and readout of the results [39]. But such assumptions have to be imposed on the classical algorithms as well, which can sometimes be shown to catch up with the quantum scaling [38]. In other cases, it is simply not known what effect the assumptions have on the performance of the classical competitor. Furthermore, even if there are few proofs on paper, classical machine learning exhibits linear runtimes in the number of data points in practice (and anything worse would be prohibitive for larger datasets). Exponential speedups are therefore likely limited to domains where machine learning fails – domains that we consequently know very little about.

The second, and currently dominating, approach of quantum machine learning replaces $\mathcal{F}$ with a class of "quantum models", and optimization is usually performed using similar tools to classical machine learning (i.e., by stochastic gradient descent [20–22, 46]). Here, quantum advantages are a bit more complex to study, since they relate to the full learning problem. To use the supervised learning task above, given a "quantum model class" $\mathcal{F}$ we have to show that the empirical risk minimization problem can be solved *and* leads to solutions that generalize well, while no classical algorithm can achieve the same feat.

We know that there are situations where all three boxes are ticked. For example, the authors in [36] can construct the ground truth for a supervised learning problem from a discrete logarithm problem, that we know a quantum computer can solve efficiently while a classical computer is unlikely to. Using the link between quantum circuits and kernel methods (see Section III) they can show that the ability to solve discrete logarithm automatically leads to the ability to both learn the optimal solution, and to ensure generalization from the proxy problem to the "real" task. Another beautiful example, which looks at the task of learning a model that can sample from the data distribution $p(x)$ in an unsupervised fashion, uses distributions that are known from quantum cryptography to be classically hard to sample from, while quantum computers can reconstruct them from samples [37]. These examples are witness to the fact that we *can* make statements about quantum advantages for learning. At the same time, it is no coincidence that they both heavily bias the data distribution $p(x)$ and/or the ground truth of the problem to a rather artificial problem that we know well in quantum computing (see also [47]). The question about whether quantum computers can really play a role in practical machine learning applications is therefore still wide open. In fact, an increasing number of researchers sees the future of quantum machine learning in analysing data in the form of quantum states, where speedups are a much more natural tool to construct arguments of "superiority".

III. ALTERNATIVE RESEARCH AGENDAS

Let us now drop the question of how quantum computing can make machine learning *better*, and look for other meaningful questions that can help to lay the foundation of quantum machine learning.

We will use examples from existing research to highlight a selection of three:

- The search for a quantum perceptron: What are natural building blocks for quantum machine learning algorithms?

- The link between quantum circuits and kernel methods: How can we construct useful bridges between quantum computing and learning theory?

- Gradients of quantum circuits: How can we make quantum software ready for machine learning applications?

Especially the last two have played an important role in searching for quantum advantage: some of the first comprehensive studies of a learning advantage have been based on the quantum kernel framework [26, 36], and practical benchmarks in a vast majority of the papers rely on software that automatically trains quantum circuits.

A. What are natural building blocks for quantum machine learning algorithms?

A perceptron [48] is a simple function

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

where $\mathbf{x}$ is an input vector, $\mathbf{w}$ a vector of trainable weights, and φ a nonlinear scalar function. The perceptron has a long history that connects machine learning with biological models of the brain, and is the basic building block of neural networks, and hence most of the modern deep learning models used in practice today. Ways of constructing quantum versions of perceptrons have sparked the imagination of researchers since more than 25 years, and quantum machine learning consequently contains a huge variety of proposals (see for example [18, 49–54], to only mention a few). Most of these papers – including our own humble contributions – artfully dance around the question of how their implementation could lead to a quantum advantage, and to our knowledge, no widely accepted argument has been brought forward yet. This comes as no surprise: following the arguments above, it is very hard to prove on paper or with benchmarks that a "quantum perceptron" is better (and one would have to first put substantial effort into developing a clear framework in which we can reason about its performance).

Leaving the idea of a quantum advantage behind allows us to shed a different light on the search for a quantum perceptron. At its core is the question whether or not we can find a building block in quantum computing that *plays the role* that a perceptron plays for neural networks. Such a building block would be a simple, modular and efficiently trainable unit for quantum machine learning models that quantum hardware can easily implement. It would allow theoretical investigations into training and generalisation behaviour. Preferably, we would also be able to pinpoint "non-classicality" or "quantumness" in such a building block, so we can directly study its influence on learning. At the moment, and without much critical reflection, this 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 towards the right practical implementations, such as the Ising model did for many-body physics?

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 that 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. How can we construct useful bridges between quantum computing and learning theory?

An example for a useful theoretical insight that builds bridges between quantum machine learning and classical learning theory was the realization that data encoding is what machine learning researchers call a "feature map" [14, 55], which means that many quantum circuits can be understood as a linear model in a feature space of the data [56, 57].

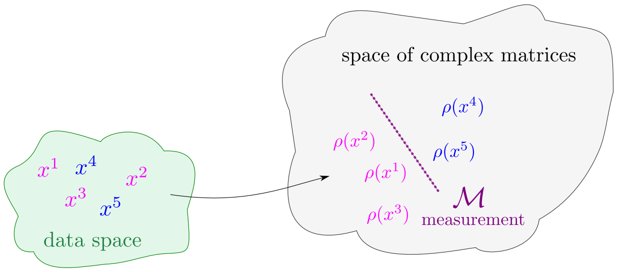

FIG. 2. Many quantum circuits used as supervised machine learning models can be understood as mapping data to quantum states and then distinguishing these quantum states via hyperplanes defined by measurement observables. Such linear models in high-dimensional spaces are known as kernel methods in classical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

In a nutshell, if we encode a data input $x \in \mathcal{X}$ into a quantum state $\rho(x)$ (for example via a quantum state preparation routine), the expectation of an observable $\mathcal{M}$ can be interpreted as a machine learning model of the form

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

Realising 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 a weight vector, and $\langle \cdot, \cdot \rangle$ the inner product in $\mathcal{H}$ (see Fig. 2). 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 optimising 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" [58]. For example, kernel theory tells us that quantum models of the form 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 we seek to classify,

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

Instead of learning the parameters θ in a variational circuit, we can learn the coefficients a_m and only need the quantum computer to evaluate the trace term (which for pure states reduces to the overlap $|\langle\psi(x)|\psi(x^m)\rangle|^2$).

Finding a common ground between kernels and quantum circuits comes with a number of perks:

- 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 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. It also guarantees that we can find the optimal solution.
- Separating the trainable parameters and optimization process from the terms $\mathrm{tr}\{\rho(\mathrm{x}), \rho(\mathrm{x}')\}$ that a quantum computer has to evaluate can help to isolate where to look for quantum advantages [26].
- The theory of kernel methods allows us to study generalization. For example, we know that it is connected to the margin between the data-encoding states $\rho(x)$ for two different classes of data [32], and that the kernel controls regularization properties of the model [59].
- Finally, the link connects quantum circuits to linear representations of neural networks such as neural tangent kernels [60] and random Fourier features [61], which are central to current investigations of deep learning - a fact that has been explored in a series of recent papers such as [62–64].

The point of this example is that the kernel method perspective does *not* immediately answer questions about how to beat classical machine learning (although a lot of papers try to use it for that purpose). Likewise, the question of whether there exist useful quantum kernels is still left wide open. Instead, it creates a foundation that *enables* such research, but first and foremost gives us an understanding of the models we are dealing with, and provides the tools to handle them.

There are many other fruitful points of contact of a similar flavour, such as the interpretation of quantum measurements as samples from a generative model [65], or the proximity of quantum computers to machine learning models inspired by many-body-physics [2], and the usefulness of neural networks in representing quantum states [66].

C. How can we make quantum software ready for machine learning applications?

The last example highlights an area of research that massively increased our capability of performing experiments and building software around quantum machine learning. Looking at classical machine learning, it is hard to understate the impact of computational platforms for the growth of the field.

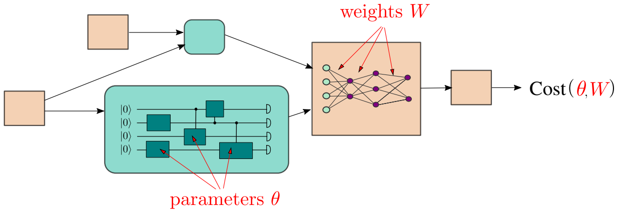

FIG. 3. Parametrized quantum circuits can be trained as parts of larger machine learning pipelines by making use of automatic differentiation and the fact that we know in many settings how to estimate analytic gradients of cost functions with respect to circuit parameters.

Historically, the dominant representation of quantum computations involved static algorithms that were hand-designed by expert theorists to maximally leverage coherent effects. More recently, there has been growing recognition that adding 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. 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, for example, kernel methods discussed in the previous section). Notably, this dichotomy mirrors the present-day situation in deep learning.

In the variational framework, a quantum circuit implements a function of the form[7]

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

where in contrast to Eq. (6) we included the free parameters θ in the measurement, and also allowed 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

[7] 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.

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 circuit's free parameters θ. Modern software tools like TensorFlow [67] or PyTorch [68] largely automate the gradient computation of deep learning models using the backpropagation algorithm [69]. These libraries even let a user optimize custom functions — such as an expectation value produced from calling quantum computing hardware — provided the user also supplies the gradient of this function.

Drawing on insights originally developed from quantum optimal control [70], 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 [46, 71], we can evaluate the derivatives $\frac{\partial f}{\partial \theta_i}$ of a parametrized circuit[8] — 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 [72–79], has a similar form to the numerical finite-difference approximator, but in fact provides an analytically exact expression[9] for any shift value $s \neq 0, \pi$. And although it does not match the efficiency of backpropagation[10], the simplicity of the parameter-shift rule makes it a very hardware-friendly mechanism for computing quantum circuit gradients.

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 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 [80] (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 [81–84] 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 [24]. 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 [85].

IV. MOVING FORWARD

This perspective advocated a shift in the research agenda of quantum machine learning away from investing all our resources into the notion of "beating" classical algorithms. Section II tried 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 Section III showed examples of alternative research questions that are important to lay the foundation of the field. Until quantum computers become available to do large-scale benchmarks, such fundamental questions may be a very good use of our time, but require a bit of courage to withstand the narrative of trying to find the billion-dollar quantum advantage, or to resist catchy terms like "deep quantum neural network".

A paradigm shift is never easy, and will require the community to make subtle but crucial adjustments, for example to the way that 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.

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

[9] 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.

[10] In backpropagation, 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 get 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 $f(x, \theta)$.

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

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

[3] S. C. Kak, Quantum neural computing, Advances in Imaging and Electron Physics **94**, 259 (1995).

[4] G. Bonnell and G. Papini, Quantum neural network, International Journal of Theoretical Physics **36**, 2855 (1997).

[5] D. Ventura and T. Martinez, Quantum associative mem-

ory, Information Sciences **124**, 273 (2000).

[6] C. A. Trugenberger, Quantum pattern recognition, Quantum Information Processing **1**, 471 (2002).

[7] R. Schützhold, Pattern recognition on a quantum computer, Physical Review A **67**, 062311 (2003).

[8] R. Bonner and R. Freivalds, A survey of quantum learning, in *Quantum Computation and Learning* (2003) p. 106.

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

[10] E. Aïmeur, G. Brassard, and S. Gambs, Machine learning in a quantum world, in *Advances in Artificial Intelligence* (Springer, 2006) pp. 431–442.

[11] P. Wittek, *Quantum machine learning: What quantum computing means to data mining* (Academic Press, 2014).

[12] M. Schuld, I. Sinayskiy, and F. Petruccione, Introduction to quantum machine learning, Contemporary Physics **56**, 172 (2015).

[13] N. Wiebe, D. Braun, and S. Lloyd, Quantum algorithm for data fitting, Physical Review Letters **109**, 050505 (2012).

[14] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Physcial Review Letters **113**, 130503 (2014).

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

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

[17] M. Denil and N. De Freitas, Toward the implementation of a quantum RBM, in *NIPS 2011 Deep Learning and Unsupervised Feature Learning Workshop* (2011).

[18] A. Kapoor, N. Wiebe, and K. Svore, Quantum perceptron models, in *Advances In Neural Information Processing Systems* (2016) pp. 3999–4007.

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

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

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

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

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

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

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

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

[27] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, The power of quantum neural networks, Nature Computational Science **1**, 403 (2021).

[28] S. Arunachalam and R. De Wolf, Optimal quantum sample complexity of learning algorithms, The Journal of Machine Learning Research **19**, 2879 (2018).

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

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

[31] A. Monras, G. Sentís, and P. Wittek, Inductive supervised quantum learning, Physical review letters **118**, 190503 (2017).

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

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

[34] V. Dunjko, J. M. Taylor, and H. J. Briegel, Quantum-enhanced machine learning, Physical Review Letters **117**, 130501 (2016).

[35] P. Rotondo, M. Marcuzzi, J. P. Garrahan, I. Lesanovsky, and M. Müller, Open quantum generalisation of hopfield neural networks, Journal of Physics A: Mathematical and Theoretical **51**, 115301 (2018).

[36] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nature Physics , 1 (2021).

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

[38] E. Tang, A quantum-inspired classical algorithm for recommendation systems, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019) pp. 217–228.

[39] S. Aaronson, Read the fine print, Nature Physics **11**, 291 (2015).

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

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

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

[43] D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro, The implicit bias of gradient descent on separable data, The Journal of Machine Learning Research **19**, 2822 (2018).

[44] S. Arora, N. Cohen, W. Hu, and Y. Luo, Implicit regularization in deep matrix factorization, Advances in Neural Information Processing Systems **32**, 7413 (2019).

[45] 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).

[46] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Physical Review A **98**, 032309 (2018).

[47] J. M. Kübler, S. Buchholz, and B. Schölkopf, The inductive bias of quantum kernels, arXiv preprint arXiv:2106.03747 (2021).

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

[49] F. Benatti, S. Mancini, and S. Mangini, Continuous variable quantum perceptron, International Journal of Quantum Information **17**, 1941009 (2019).

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

[51] M. Andrecut and M. Ali, A quantum perceptron, International Journal of Modern Physics B **16**, 639 (2002).

[52] M. Lewenstein, Quantum perceptrons, Journal of Modern Optics **41**, 2491 (1994).

[53] T. Espinosa-Ortega and T. Liew, Perceptrons with Hebbian learning based on wave ensembles in spatially patterned potentials, Physical Review Letters **114**, 118101 (2015).

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

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

[56] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, Physical review letters **122**, 040504 (2019).

[57] 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).

[58] M. Schuld, Quantum machine learning models are kernel methods, arXiv preprint arXiv:2101.11020 (2021).

[59] I. Steinwart and A. Christmann, *Support Vector Machines*, 1st ed. (Springer Publishing Company, Incorporated, 2008).

[60] A. Rahimi, B. Recht, *et al.*, Random features for large-scale kernel machines., in *NIPS*, Vol. 3 (Citeseer, 2007) p. 5.

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

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

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

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

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

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

[67] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, Tensorflow: A system for large-scale machine learning, in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (2016) pp. 265–283.

[68] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems **32**, 8026 (2019).

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

[70] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid quantum-classical approach to quantum optimal control, Physical review letters **118**, 150503 (2017).

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

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

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

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

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

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

[77] 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, arXiv preprint arXiv:2107.08131 (2021).

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

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

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

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

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

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

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

[85] E. M. Stoudenmire, Learning relevant features of data with multi-scale tensor networks, Quantum Science and Technology **3**, 034003 (2018).