Power of data in quantum machine learning

Hsin-Yuan Huang[1,2,3], Michael Broughton[1], Masoud Mohseni[1], Ryan Babbush[1], Sergio Boixo [1], Hartmut Neven[1] & Jarrod R. McClean [1]✉

The use of quantum computing for machine learning is among the most exciting prospective applications of quantum technologies. However, machine learning tasks where data is provided can be considerably different than commonly studied computational tasks. In this work, we show that some problems that are classically hard to compute can be easily predicted by classical machines learning from data. Using rigorous prediction error bounds as a foundation, we develop a methodology for assessing potential quantum advantage in learning tasks. The bounds are tight asymptotically and empirically predictive for a wide range of learning models. These constructions explain numerical results showing that with the help of data, classical machine learning models can be competitive with quantum models even if they are tailored to quantum problems. We then propose a projected quantum model that provides a simple and rigorous quantum speed-up for a learning problem in the fault-tolerant regime. For near-term implementations, we demonstrate a significant prediction advantage over some classical models on engineered data sets designed to demonstrate a maximal quantum advantage in one of the largest numerical tests for gate-based quantum machine learning to date, up to 30 qubits.

[1] Google Quantum AI, Venice, CA, USA. [2] Institute for Quantum Information and Matter, Caltech, Pasadena, CA, USA. [3] Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA. ✉email: jmcclean@google.com

As quantum technologies continue to rapidly advance, it becomes increasingly important to understand which applications can benefit from the power of these devices. At the same time, machine learning on classical computers has made great strides, revolutionizing applications in image recognition, text translation, and even physics applications, with more computational power leading to ever increasing performance[1]. As such, if quantum computers could accelerate machine learning, the potential for impact is enormous.

At least two paths towards quantum enhancement of machine learning have been considered. First, motivated by quantum applications in optimization[2–4], the power of quantum computing could, in principle, be used to help improve the training process of existing classical models[5,6], or enhance inference in graphical models[7]. This could include finding better optima in a training landscape or finding optima with fewer queries. However, without more structure known in the problem, the advantage along these lines may be limited to quadratic or small polynomial speedups[8,9].

The second vein of interest is the possibility of using quantum models to generate correlations between variables that are inefficient to represent through classical computation. The recent success both theoretically and experimentally for demonstrating quantum computations beyond classical tractability can be taken as evidence that quantum computers can sample from probability distributions that are exponentially difficult to sample from classically[10,11]. If these distributions were to coincide with real-world distributions, this would suggest the potential for significant advantage. This is typically the type of advantage that has been sought in recent work on both quantum neural networks[12–14], which seek to parameterize a distribution through some set of adjustable parameters, and quantum kernel methods[15] that use quantum computers to define a feature map that maps classical data into the quantum Hilbert space. The justification for the capability of these methods to exceed classical models often follows similar lines as refs. [10,11] or quantum simulation results. That is, if the model leverages a quantum circuit that is hard to sample results from classically, then there is potential for a quantum advantage.

In this work, we show quantitatively how this picture is incomplete in machine learning (ML) problems where some training data are provided. The provided data can elevate classical models to rival quantum models, even when the quantum circuits generating the data are hard to compute classically. We begin with a motivating example and complexity-theoretic argument showing how classical algorithms with data can match quantum output. Following this, we provide rigorous prediction error bounds for training classical and quantum ML methods based on kernel functions[15–24] to learn quantum mechanical models. We focus on kernel methods, as they not only provide provable guarantees, but are also very flexible in the functions they can learn. For example, recent advancements in theoretical machine learning show that training neural networks with large hidden layers is equivalent to training an ML model with a particular kernel, known as the neural tangent kernel[19–21]. Throughout, when we refer to classical ML models related to our theoretical developments, we will be referring to ML models that can be easily associated with a kernel, either explicitly as in kernel methods, or implicitly as in the neural tangent kernels. However, in the numerical section, we will also include performance comparisons to methods where direct association of a kernel is challenging, such as random forest methods. In the quantum case, we will also show how quantum ML based on kernels can be made equivalent to training an infinite depth quantum neural network.

We use our prediction error bounds to devise a flowchart for testing potential quantum prediction advantage, the separation between prediction errors of quantum and classical ML models for a fixed amount of training data. The most important test is a geometric difference between kernel functions defined by classical and quantum ML. Formally, the geometric difference is defined by the closest efficient classical ML model. In practice, one should consider the geometric difference with respect to a suite of optimized classical ML models. If the geometric difference is small, then a classical ML method is guaranteed to provide similar or better performance in prediction on the dataset, independent of the function values or labels. Hence this represents a powerful, function independent prescreening that allows one to evaluate if there is any possibility of better performance. On the other hand, if the geometry differs greatly, we show both the existence of a dataset that exhibits large prediction advantage using the quantum ML model and how one can construct it efficiently. While the tools we develop could be used to compare and construct hard classical models like hash functions, we enforce restrictions that allow us to say something about a quantum separation. In particular, the feature map will be white box, in that a quantum circuit specification is available for the ideal feature map, and that feature map can be made computationally hard to evaluate classically. A constructive example of this is a discrete log feature map, where a provable separation for our kernel is given in Supplementary Section 11. Additionally, the minimum over classical models means that classical hash functions are reproduced formally by definition.

Moreover, application of these tools to existing models in the literature rules many of them out immediately, providing a powerful sieve for focusing development of new data encodings. Following these constructions, in numerical experiments, we find that a variety of common quantum models in the literature perform similarly or worse than classical ML on both classical and quantum datasets due to a small geometric difference. The small geometric difference is a consequence of the exponentially large Hilbert space employed by existing quantum models, where all inputs are too far apart. To circumvent the setback, we propose an improvement, which enlarges the geometric difference by projecting quantum states embedded from classical data back to approximate classical representation[25–27]. With the large geometric difference endowed by the projected quantum model, we are able to construct engineered datasets to demonstrate large prediction advantage over common classical ML models in numerical experiments up to 30 qubits. Despite our constructions being based on methods with associated kernels, we find empirically that the prediction advantage remains robust across tested classical methods, including those without an easily determined kernel. This opens the possibility to use a small quantum computer to generate efficiently verifiable machine learning problems that could be challenging for classical ML models.

Results

Setup and motivating example. We begin by setting up the problems and methods of interest for classical and quantum models, and then provide a simple motivating example for studying how data can increase the power of classical models on quantum data. The focus will be a supervised learning task with a collection of N training examples $\{(x_i, y_i)\}$, where x_i is the input data and y_i is an associated label or value. We assume that x_i are sampled independently from a data distribution $\mathcal{D}$.

In our theoretical analysis, we will consider $y_i \in \mathbb{R}$ to be generated by some quantum model. In particular, we consider a

continuous encoding unitary that maps a classical input data x_i into quantum state $|x_i\rangle = U_{\mathrm{enc}}(x_i)|0\rangle^{\otimes n}$ and refer to the corresponding density matrix as $\rho(x_i)$. The expressive power of these embeddings have been investigated from a functional analysis point of view[28,29]; however, the setting where data are provided requires special attention. The encoding unitary is followed by a unitary $U_{\mathrm{QNN}}(\theta)$. We then measure an observable O after the quantum neural network. This produces the label/value for input x_i given as $y_i = f(x_i) = \langle x_i|U_{\mathrm{QNN}}^\dagger O U_{\mathrm{QNN}}|x_i\rangle$. The quantum model considered here is also referred to as a quantum neural network (QNN) in the literature[14,30]. The goal is to understand when it is easy to predict the function $f(x)$ by training classical/quantum machine learning models.

With notation in place, we turn to a simple motivating example to understand how the availability of data in machine learning tasks can change computational hardness. Consider data points $\{\mathbf{x}_i\}_{i=1}^N$ that are p-dimensional classical vectors with $\|\mathbf{x}_i\|_2 = 1$, and use amplitude encoding[31–33] to encode the data into an n-qubit state $|\mathbf{x}_i\rangle = \sum_{k=1}^p x_i^k|k\rangle$, where x_i^k is the individual coordinate of the vector $\mathbf{x}_i$. If U_{QNN} is a time-evolution under a many-body Hamiltonian, then the function $f(\mathbf{x}) = \langle \mathbf{x}|U_{\mathrm{QNN}}^\dagger O U_{\mathrm{QNN}}|\mathbf{x}\rangle$ is in general hard to compute classically[34], even for a single input state. In particular, we have the following proposition showing that if a classical algorithm can compute $f(\mathbf{x})$ efficiently, then quantum computers will be no more powerful than classical computers; see Supplementary Section 1 for a proof.

Proposition 1. If a classical algorithm without training data can compute $f(\mathbf{x})$ efficiently for any U_{QNN} and O, then $\mathrm{BPP} = \mathrm{BQP}$.

Nevertheless, it is incorrect to conclude that training a classical model from data to learn this evolution is hard. To see this, we write out the expectation value as

$$\begin{aligned} f(x_i) &= \left(\sum_{k=1}^p x_i^{k*}\langle k|\right) U_{\mathrm{QNN}}^\dagger O U_{\mathrm{QNN}} \left(\sum_{l=1}^p x_i^l|l\rangle\right) \\ &= \sum_{k=1}^p \sum_{l=1}^p B_{kl} x_i^{k*} x_i^l, \end{aligned} \tag{1}$$

which is a quadratic function with p^2 coefficients $B_{kl} = \langle k|U_{\mathrm{QNN}}^\dagger O U_{\mathrm{QNN}}|l\rangle$. Using the theory developed later in this work, we can show that, for any U_{QNN} and O, training a specific classical ML model on a collection of N training examples $\{(\mathbf{x}_i, y_i = f(\mathbf{x}_i))\}$ would give rise to a prediction model $h(\mathbf{x}_i)$ with

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}|h(\mathbf{x}) - f(\mathbf{x})| \le c\sqrt{\frac{p^2}{N}}, \tag{2}$$

for a constant $c > 0$. We refer to Supplementary Section 1 for the proof of this result. Hence, with $N \propto p^2/\epsilon^2$ training data, one can train a classical ML model to predict the function $f(\mathbf{x})$ up to an additive prediction error ϵ. This elevation of classical models through some training samples is illustrative of the power of data. In Supplementary Section 2, we give a rigorous complexity-theoretic argument on the computational power provided by data. A cartoon depiction of the complexity separation induced by data is provided in Fig. 1(a).

While this simple example makes the basic point that sufficient data can change complexity considerations, it perhaps opens more questions than it answers. For example, it uses a rather weak encoding into amplitudes and assumes one has access to an amount of data that is on par with the dimension of the model. The more interesting cases occur if we strengthen the data encoding, include modern classical ML models, and consider the number of data N much less than the dimension of the model. These more interesting cases are the ones we quantitatively answer.
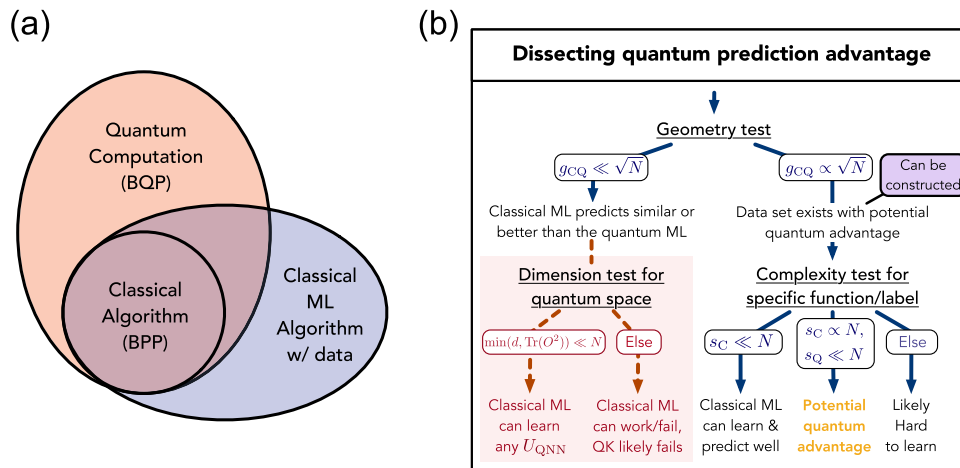

Fig. 1 Illustration of the relation between complexity classes and a flowchart for understanding and prescreening potential quantum advantage. a We cartoon the separation between problem complexities that are created by the addition of data to a problem. Classical algorithms that can learn from data define a complexity class that can solve problems beyond classical computation (BPP), but it is still expected that quantum computation can efficiently solve problems that classical ML algorithms with data cannot. A rigorous definition and proof for the separation between classical algorithms that can learn from data and BPP/BQP is given in Supplementary Section 2. **b** The flowchart we develop for understanding the potential for quantum prediction advantage. N samples of data from a potentially infinite depth QNN made with encoding and function circuits U_{enc} and U_{QNN} are provided as input along with quantum and classical methods with associated kernels. Tests are given as functions of N to emphasize the role of data in the possibility of a prediction advantage. One can first evaluate a geometric quantity g_{CQ} that measures the possibility of an advantageous quantum/classical prediction separation without yet considering the actual function to learn. We show how one can efficiently construct an adversarial function that saturates this limit if the test is passed, otherwise the classical approach is guaranteed to match performance for any function of the data. To subsequently consider the actual function provided, a label/function-specific test may be run using the model complexities s_C and s_Q. If one specifically uses the quantum kernel (QK) method, the red dashed arrows can evaluate if all possible choices of U_{QNN} lead to an easy classical function for the chosen encoding of the data.

Our primary interest will be ML algorithms that are much stronger than fitting a quadratic function and the input data are provided in more interesting ways than an amplitude encoding. In this work, we focus on both classical and quantum ML models based on kernel functions $k(x_i, x_j)$. At a high level, a kernel function can be seen as a measure of similarity, if $k(x_i, x_j)$ is large when x_i and x_j are close. When considered for finite input data, a kernel function may be represented as a matrix $K_{ij} = k(x_i, x_j)$ and the conditions required for kernel methods are satisfied when the matrix representation is Hermitian and positive semi-definite.

A given kernel function corresponds to a nonlinear feature mapping $\phi(x)$ that maps x to a possibly infinite-dimensional feature space, such that $k(x_i, x_j) = \phi(x_i)^\dagger \phi(x_j)$. This is the basis of the so-called "kernel trick" where intricate and powerful maps $\phi(x_i)$ can be implemented through the evaluation of relatively simple kernel functions k. As a simple case, in the example above, using a kernel of $k(x_i, x_j) = |\langle x_i | x_j \rangle|^2$ corresponds to a feature map $\phi(x_i) = \sum_{kl} x_i^{k*} x_i^l |k\rangle \otimes |l\rangle$ which is capable of learning quadratic functions in the amplitudes. In kernel based ML algorithms, the trained model can always be written as $h(x) = \mathbf{w}^\dagger \phi(x)$ where $\mathbf{w}$ is a vector in the feature space defined by the kernel. For example, training a convolutional neural network with large hidden layers[19,35] is equivalent to using a corresponding neural tangent kernel k^{CNN}. The feature map ϕ^{CNN} for the kernel k^{CNN} is a nonlinear mapping that extracts all local properties of x[35]. In quantum mechanics, similarly a kernel function can be defined using the native geometry of the quantum state space $|x\rangle$. For example, we can define the kernel function as $\langle x_i | x_j \rangle$ or $|\langle x_i | x_j \rangle|^2$. Using the output from this kernel in a method like a classical support vector machine[16] defines the quantum kernel method.

A wide class of functions can be learned with a sufficiently large amount of data by using the right kernel function k. For example, in contrast to the perhaps more natural kernel, $\langle x_i | x_j \rangle$, the quantum kernel $k^Q(x_i, x_j) = |\langle x_i | x_j \rangle|^2 = \text{Tr}(\rho(x_i)\rho(x_j))$ can learn arbitrarily deep quantum neural network U_{QNN} that measures any observable O (shown in Supplementary Section 3), and the Gaussian kernel, $k^\gamma(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$ with hyperparameter γ, can learn any continuous function in a compact space[36], which includes learning any QNN. Nevertheless, the required amount of data N to achieve a small prediction error could be very large in the worst case. Although we will work with other kernels defined through a quantum space, due both to this expressive property and terminology of past work, we will refer to $k^Q(x_i, x_j) = \text{Tr}[\rho(x_i)\rho(x_j)]$ as the quantum kernel method throughout this work, which is also the definition given in[15].

Testing quantum advantage. We now construct our more general framework for assessing the potential for quantum prediction advantage in a machine learning task. Beginning from a general result, we build both intuition and practical tests based on the geometry of the learning spaces. This framework is summarized in Fig. 1.

Our foundation is a general prediction error bound for training classical/quantum ML models to predict some quantum model defined by $f(x) = \text{Tr}(O^U \rho(x))$ derived from concentration inequalities, where $O^U = U_{\text{QNN}}^\dagger O U_{\text{QNN}}$. Suppose we have obtained N training examples $\{(x_i, y_i = f(x_i))\}$. After training on this data, there exists an ML algorithm that outputs $h(x) = \mathbf{w}^\dagger \phi(x)$ using kernel $k(x_i, x_j) = K_{ij} = \phi(x_i)^\dagger \phi(x_j)$, which has a simplified prediction error bounded by

$$\mathbb{E}_{x \sim \mathcal{D}} |h(x) - f(x)| \leq c \sqrt{\frac{s_K(N)}{N}} \qquad (3)$$

for a constant $c > 0$ and N independent samples from the data

distribution $\mathcal{D}$. We note here that this and all subsequent bounds have a key dependence on the quantity of data N, reflecting the role of data to improve prediction performance. Due to a scaling freedom between $\alpha \phi(x)$ and $\mathbf{w}/\alpha$, we have assumed $\sum_{i=1}^N \phi(x_i)^\dagger \phi(x_i) = \text{Tr}(K) = N$. A derivation of this result is given in Supplementary Section 4.

Given this core prediction error bound, we now seek to understand its implications. The main quantity that determines the prediction error is

$$s_K(N) = \sum_{i=1}^N \sum_{j=1}^N (K^{-1})_{ij} \text{Tr}(O^U \rho(x_i)) \text{Tr}(O^U \rho(x_j)). \qquad (4)$$

The quantity $s_K(N)$ is equal to the model complexity of the trained function $h(x) = \mathbf{w}^\dagger \phi(x)$, where $s_K(N) = ||\mathbf{w}||^2 = \mathbf{w}^\dagger \mathbf{w}$ after training. A smaller value of $s_K(N)$ implies better generalization to new data x sampled from the distribution $\mathcal{D}$. Intuitively, $s_K(N)$ measures whether the closeness between x_i, x_j defined by the kernel function $k(x_i, x_j)$ matches well with the closeness of the observable expectation for the quantum states $\rho(x_i), \rho(x_j)$, recalling that a larger kernel value indicates two points are closer. The computation of $s_K(N)$ can be performed efficiently on a classical computer by inverting an $N \times N$ matrix K after obtaining the N values $\text{Tr}(O^U \rho(x_i))$ by performing order N experiments on a physical quantum device. The time complexity scales at most as order N^3. Due to the connection between $\mathbf{w}^\dagger \mathbf{w}$ and the model complexity, a regularization term $\mathbf{w}^\dagger \mathbf{w}$ is often added to the optimization problem during the training of $h(x) = \mathbf{w}^\dagger \phi(x)$, see e.g., refs. [16,37,38]. Regularization prevents $s_K(N)$ from becoming too large at the expense of not completely fitting the training data. A detailed discussion and proof under regularization is given in Supplementary Section 4 and 6.

The prediction error upper bound can often be shown to be asymptotically tight by proving a matching lower bound. As an example, when $k(x_i, x_j)$ is the quantum kernel $\text{Tr}(\rho(x_i)\rho(x_j))$, we can deduce that $s_K(N) \leq \text{Tr}(O^2)$ hence one would need a number of data N scaling as $\text{Tr}(O^2)$. In Supplementary Section 8, we give a matching lower bound showing that a scaling of $\text{Tr}(O^2)$ is unavoidable if we assume a large Hilbert space dimension. This lower bound holds for any learning algorithm and not only for quantum kernel methods. The lower bound proof uses mutual information analysis and could easily extend to other kernels. This proof strategy is also employed extensively in a follow-up work[39] to devise upper and lower bounds for classical and quantum ML in learning quantum models. Furthermore, not only are the bounds asymptotically tight, in numerical experiments given in Supplementary Section 13 we find that the prediction error bound also captures the performance of other classical ML models not based on kernels where the constant factors are observed to be quite modest.

Given some set of data, if $s_K(N)$ is found to be small relative to N after training for a classical ML model, this quantum model $f(x)$ can be predicted accurately even if $f(x)$ is hard to compute classically for any given x. In order to formally evaluate the potential for quantum prediction advantage generally, one must take $s_K(N)$ to be the minimal over efficient classical models. However, we will be more focused on minimally attainable values over a reasonable set of classical methods with tuned hyperparameters. This prescribes an effective method for evaluating potential quantum advantage in practice, and already rules out a considerable number of examples from the literature.

From the bound, we can see that the potential advantage for one ML algorithm defined by K^1 to predict better than another ML algorithm defined by K^2 depends on the largest possible separation between s_{K^1} and s_{K^2} for a dataset. The separation can be characterized by defining an asymmetric geometric difference
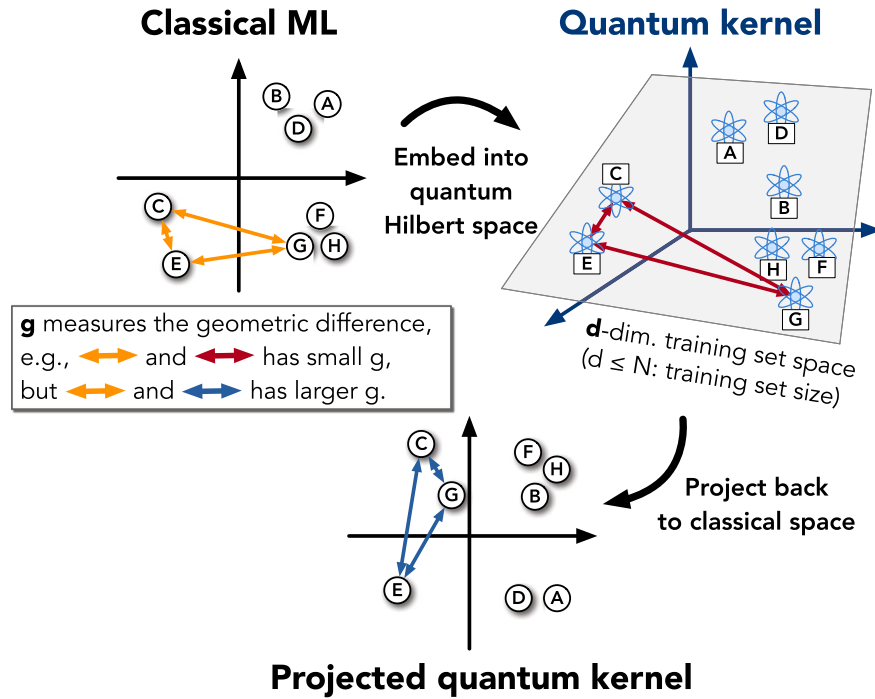
Fig. 2 Cartoon of the geometry (kernel function) defined by classical and quantum ML models. The letters A, B, ... represent data points $\{x_i\}$ in different spaces with arrows representing the similarity measure (kernel function) between data. The geometric difference g is a difference between similarity measures (arrows) in different ML models and d is an effective dimension of the dataset in the quantum Hilbert space.

that depends on the dataset, but is independent of the function values or labels. Hence evaluating this quantity is a good first step in understanding if there is a potential for quantum advantage, as shown in Fig. 1. This quantity is defined by

$$g_{12} = g(K^1 || K^2) = \sqrt{|| \sqrt{K^2}(K^1)^{-1}\sqrt{K^2} ||_\infty}, \quad (5)$$

where $||.||_\infty$ is the spectral norm of the resulting matrix and we assume $\mathrm{Tr}(K^1) = \mathrm{Tr}(K^2) = N$. One can show that $s_{K^1} \leq g_{12}^2 s_{K^2}$, which implies the prediction error bound $c\sqrt{s_{K^1}/N} \leq c g_{12}\sqrt{s_{K^2}/N}$. A detailed derivation is given in Supplementary Section C and an illustration of g_{12} can be found in Fig. 2. The geometric difference $g(K^1 || K^2)$ can be computed on a classical computer by performing a singular value decomposition of the $N \times N$ matrices K^1 and K^2. Standard numerical analysis packages[40] provide highly efficient computation of a singular value decomposition in time at most order N^3. Intuitively, if $K^1(x_i, x_j)$ is small/large when $K^2(x_i, x_j)$ is small/large, then the geometric difference g_{12} is a small value ~1, where g_{12} grows as the kernels deviate.

To see more explicitly how the geometric difference allows one to make statements about the possibility for one ML model to make different predictions from another, consider the geometric difference $g_{CQ} = g(K^C || K^Q)$ between a classical ML model with kernel $k^C(x_i, x_j)$ and a quantum ML model, e.g., with $k^Q(x_i, x_j) = \mathrm{Tr}(\rho(x_i)\rho(x_j))$. If g_{CQ} is small, because

$$s_C \leq g_{CQ}^2 s_Q, \quad (6)$$

the classical ML model will always have a similar or better model complexity $s_K(N)$ compared to the quantum ML model. This implies that the prediction performance for the classical ML will likely be competitive or better than the quantum ML model, and one is likely to prefer using the classical model. This is captured in the first step of our flowchart in Fig. 1.

In contrast, if g_{CQ} is large we show that there exists a dataset with $s_C = g_{CQ}^2 s_Q$ with the quantum model exhibiting superior

prediction performance. An efficient method to explicitly construct such a maximally divergent dataset is given in Supplementary Section 7 and a numerical demonstration of the stability of this separation is provided in the next section. While a formal statement about classical methods generally requires defining it overall efficient classical methods, in practice, we consider g_{CQ} to be the minimum geometric difference among a suite of optimized classical ML models. Our engineered approach minimizes this value as a hyperparameter search to find the best classical adversary, and shows remarkable robustness across classical methods including those without an associated kernel, such as random forests[41].

In the specific case of the quantum kernel method with $K^Q_{ij} = k^Q(x_i, x_j) = \mathrm{Tr}(\rho(x_i)\rho(x_j))$, we can gain additional insights into the model complexity s_K, and sometimes make conclusions about classically learnability for all possible U_{QNN} for the given encoding of the data. Let us define $\mathrm{vec}(X)$ for a Hermitian matrix X to be a vector containing the real and imaginary part of each entry in X. In this case, we find $s_Q = \mathrm{vec}(O^U)^T P_Q \mathrm{vec}(O^U)$, where P_Q is the projector onto the subspace formed by $\{\mathrm{vec}(\rho(x_1)), \ldots, \mathrm{vec}(\rho(x_N))\}$. We highlight

$$d = \dim(P_Q) = \mathrm{rank}(K^Q) \leq N, \quad (7)$$

which defines the effective dimension of the quantum state space spanned by the training data. An illustration of the dimension d can be found in Fig. 1. Because P_Q is a projector and has eigenvalues 0 or 1, $s_Q \leq \min(d, \mathrm{vec}(O^U)^T \mathrm{vec}(O^U)) = \min(d, \mathrm{Tr}(O^2))$ assuming $||O||_\infty \leq 1$. Hence in the case of the quantum kernel method, the prediction error bound may be written as

$$\mathbb{E}_{x \in \mathcal{D}} |h(x) - f(x)| \leq c\sqrt{\frac{\min(d, \mathrm{Tr}(O^2))}{N}}. \quad (8)$$

A detailed derivation is given in Supplementary Section A. We can also consider the approximate dimension d, where small eigenvalues in K^Q are truncated, by incurring a small training error. After

obtaining K^Q from a quantum device, the dimension d can be computed efficiently on a classical machine by performing a singular value decomposition on the $N \times N$ matrix K^Q. Estimation of $\mathrm{Tr}(O^2)$ can be performed by sampling random states $|\psi\rangle$ from a quantum 2-design, measuring O on $|\psi\rangle$, and performing statistical analysis on the measurement data[25]. This prediction error bound shows that a quantum kernel method can learn any U_{QNN} when the dimension of the training set space d or the squared Frobenius norm of observable $\mathrm{Tr}(O^2)$ is much smaller than the amount of data N. In Supplementary Section 8, we show that quantum kernel methods are optimal for learning quantum models with bounded $\mathrm{Tr}(O^2)$ as they saturate the fundamental lower bound. However, in practice, most observables, such as Pauli operators, will have exponentially large $\mathrm{Tr}(O^2)$, so the central quantity is the dimension d. Using the prediction error bound for the quantum kernel method, if both g_{CQ} and $\min(d, \mathrm{Tr}(O^2))$ are small, then a classical ML would also be able to learn any U_{QNN}. In such a case, one must conclude that the given encoding of the data is classically easy, and this cannot be affected by an arbitrarily deep U_{QNN}. This constitutes the bottom left part of our flowchart in Fig. 1.

Ultimately, to see a prediction advantage in a particular dataset with specific function values/labels, we need a large separation between s_C and s_Q. This happens when the inputs x_i, x_j considered close in a quantum ML model are actually close in the target function $f(x)$, but are far in classical ML. This is represented as the final test in Fig. 1 and the methodology here outlines how this result can be achieved in terms of its more essential components.

Projected quantum kernels. In addition to analyzing existing quantum models, the analysis approach introduced also provides suggestions for new quantum models with improved properties, which we now address here. For example, if we start with the original quantum kernel, when the effective dimension d is large, kernel $\mathrm{Tr}(\rho(x_i)\rho(x_j))$, which is based on a fidelity-type metric, will regard all data to be far from each other and the kernel matrix K^Q will be close to identity. This results in a small geometric difference g_{CQ} leading to classical ML models being competitive or outperforming the quantum kernel method. In Supplementary Section 9, we present a simple quantum model that requires an exponential amount of samples to learn using the quantum kernel $\mathrm{Tr}(\rho(x_i)\rho(x_j))$, but only needs a linear number of samples to learn using a classical ML model.

To circumvent this setback, we propose a family of projected quantum kernels as a solution. These kernels work by projecting the quantum states to an approximate classical representation, e.g., using reduced physical observables or classical shadows[25,27,42–44]. Even if the training set space has a large dimension $d \sim N$, the projection allows us to reduce to a low-dimensional classical space that can generalize better. Furthermore, by going through the exponentially large quantum Hilbert space, the projected quantum kernel can be challenging to evaluate without a quantum computer. In numerical experiments, we find that the classical projection increases rather than decreases the geometric difference with classical ML models. These constructions will be the foundation of our best performing quantum method later.

One of the simplest forms of projected quantum kernel is to measure the one-particle reduced density matrix (1-RDM) on all qubits for the encoded state, $\rho_k(x_i) = \mathrm{Tr}_{j \neq k}[\rho(x_i)]$, then define the kernel as

$$k^{\mathrm{PQ}}(x_i, x_j) = \exp\left(-\gamma \sum_k ||\rho_k(x_i) - \rho_k(x_j)||_F^2\right). \quad (9)$$

This kernel defines a feature map function in the 1-RDM space

that is capable of expressing arbitrary functions of powers of the 1-RDMs of the quantum state. From nonintuitive results in density functional theory, we know even one body densities can be sufficient for determining exact ground state[45] and time-dependent[46] properties of many-body systems under modest assumptions. In Supplementary Section 10, we provide examples of other projected quantum kernels. This includes an efficient method for computing a kernel function that contains all orders of RDMs using local randomized measurements and the formalism of classical shadows[25]. The classical shadow formalism allows efficient construction of RDMs from very few measurements. In Supplementary Section 11, we show that projected versions of quantum kernels lead to a simple and rigorous quantum speed-up in a recently proposed learning problem based on discrete logarithms[24].

Numerical studies. We now provide numerical evidence up to 30 qubits that supports our theory on the relation between the dimension d, the geometric difference g, and the prediction performance. Using the projected quantum kernel, the geometric difference g is much larger and we see the strongest empirical advantage of a scalable quantum model on quantum datasets to date. These are the largest combined simulation and analysis in digital quantum machine learning that we are aware of, and make use of the TensorFlow and TensorFlow-Quantum package[47], reaching a peak throughput of up to 1.1 quadrillion floating point operations per second (petaflop/s). Trends of ~300 teraflop/s for quantum simulation and 800 teraflop/s for classical analysis were observed up to the maximum experiment size with the overall floating point operations across all experiments totalling ~2 quintillion (exaflop).

In order to mimic a data distribution that pertains to real-world data, we conduct our experiments around the fashion-MNIST dataset[48], which is an image classification for distinguishing clothing items, and is more challenging than the original digit-based MNIST source[49]. We preprocess the data using principal component analysis[50] to transform each image into an n-dimensional vector. The same data are provided to the quantum and classical models, where in the classical case the data is the n-dimensional input vector, and the quantum case uses a given circuit to embed the n-dimensional vector into the space of n qubits. For quantum embeddings, we explore three options, E1 is a separable rotation circuit[32,51,52], E2 is an IQP-type embedding circuit[15], and E3 is a Hamiltonian evolution circuit, with explicit constructions in Supplementary Section 12.

For the classical ML task (C), the goal is to correctly identify the images as shirts or dresses from the original dataset. For the quantum ML tasks, we use the same fashion-MINST source data and embeddings as above, but take as function values the expectation value of a local observable that has been evolved under a quantum neural network resembling the Trotter evolution of 1D-Heisenberg model with random couplings. In these cases, the embedding is taken as part of the ground truth, so the resulting function will be different depending on the quantum embedding. For these ML tasks, we compare against the best performing model from a list of standard classical ML algorithms with properly tuned hyperparameters (see Supplementary Section 12 for details).

In Fig. 3, we give a comparison between the prediction performance of classical and quantum ML models. One can see that not only do classical ML models perform best on the original classical dataset, the prediction performance for the classical methods on the quantum datasets is also very competitive and can even outperform existing quantum ML models despite the quantum ML models having access to the training embedding
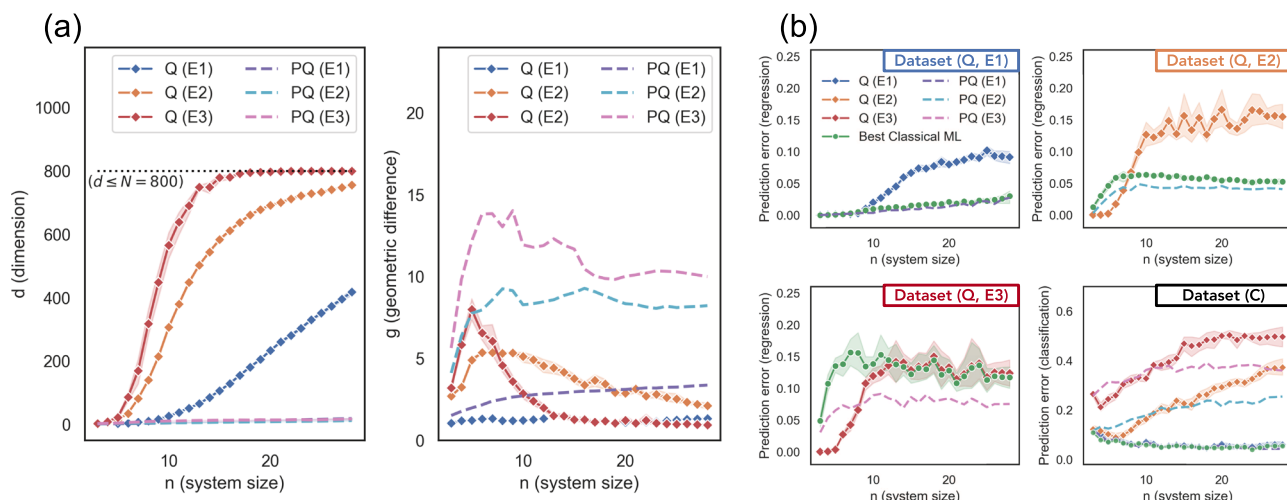
Fig. 3 Relation between dimension *d*, geometric difference *g*, and prediction performance. The shaded regions are the standard deviation over 10 independent runs and *n* is the number of qubits in the quantum encoding and dimension of the input for the classical encoding. **a** The approximate dimension *d* and the geometric difference *g* with classical ML models for quantum kernel (Q) and projected quantum kernel (PQ) under different embeddings and system sizes *n*. **b** Prediction error (lower is better) of the quantum kernel method (Q), projected quantum kernel method (PQ), and classical ML models on classical (C) and quantum (Q) datasets with number of data $N = 600$. As *d* grows too large, the geometric difference *g* for quantum kernel becomes small. We see that small geometric difference *g* always results in classical ML being competitive or outperforming the quantum ML model. When *g* is large, there is a potential for improvement over classical ML. For example, projected quantum kernel improves upon the best classical ML in Dataset (Q, E3).
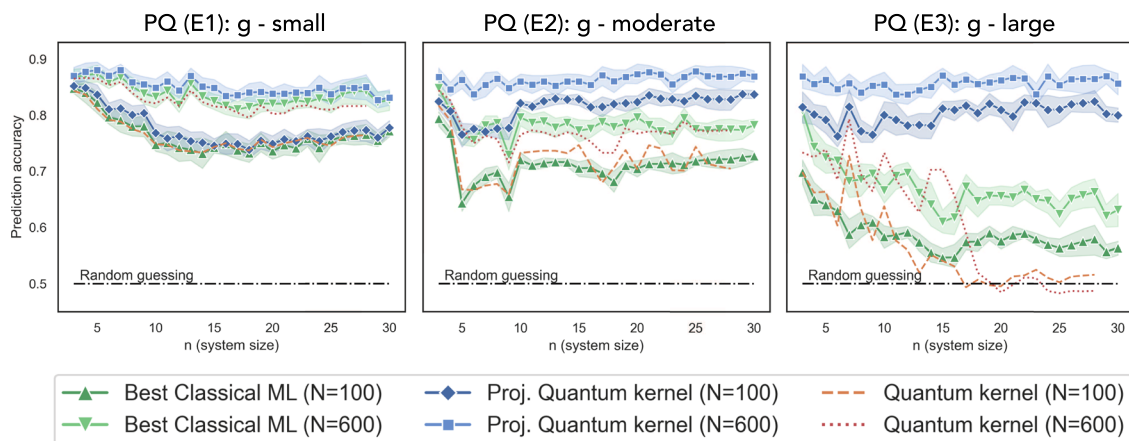

Fig. 4 Prediction accuracy (higher the better) on engineered datasets. A label function is engineered to match the geometric difference *g*(C‖PQ) between projected quantum kernel and classical approaches, demonstrating a significant gap between quantum and the best classical models up to 30 qubits when *g* is large. We consider the best performing classical ML models among Gaussian SVM, linear SVM, Adaboost, random forest, neural networks, and gradient boosting. We only report the accuracy of the quantum kernel method up to system size $n = 28$ due to the high simulation cost and the inferior performance.

while the classical methods do not. The performance of the classical ML model is especially strong on Dataset (Q, E1) and Dataset (Q, E2). This elevation of the classical performance is evidence of the power of data. Moreover, this intriguing behavior and the lack of quantum advantage may be explained by considering the effective dimension *d* and the geometric difference *g* following our theoretical constructions. From Fig. 3a, we can see that the dimension *d* of the original quantum state space grows rather quickly, and the geometric difference *g* becomes small as the dimension becomes too large ($d \propto N$) for the standard quantum kernel. The saturation of the dimension coincides with the decreasing and statistical fluctuations in performance seen in Fig. 4. Moreover, given poor ML performance a natural instinct is to throw more resources at the problem, e.g., more qubits, but as demonstrated here, doing

this for naïve quantum kernel methods is likely to lead to tiny inner products and even worse performance. In contrast, the projected quantum space has a low dimension even when *d* grows, and yields a higher geometric difference *g* for all embeddings and system sizes. Our methodology predicts that, when *g* is small, classical ML model will be competitive or outperform the quantum ML model. This is verified in Fig. 3b for both the original and projected quantum kernel, where a small geometric difference *g* leads to a very good performance of classical ML models and no large quantum advantage can be seen. Only when the geometric difference *g* is large (projected kernel method with embedding E3) can we see some mild advantage over the best classical method. This result holds disregarding any detail of the quantum evolution we are trying to learn, even for ones that are hard to simulate classically.

In order to push the limits of separation between quantum and classical approaches in a learning setting, we now consider a set of engineered datasets with function values designed to saturate the geometric inequality $s_C \leq g(K^C || K^{PQ})^2 s_{PQ}$ between classical ML models with associated kernels and the projected quantum kernel method. In particular, we design the dataset such that $s_{PQ} = 1$ and $s_C = g(K^C || K^{PQ})^2$. Recall from Eq. (3), this dataset will hence show the largest separation in the prediction error bound $\sqrt{s(N)/N}$. The engineered dataset is constructed via a simple eigenvalue problem with the exact procedure described in Supplementary Section 7 and the results are shown in Fig. 4. As the quantum nature of the encoding increases from E1 to E3, corresponding to increasing g, the performance of both the best classical methods and the original quantum kernel decline precipitously. The advantage of the projected quantum kernel closely follows the geometric difference g and reaches more than 20% for large sizes. Despite the optimization of g only being possible for classical methods with an associated kernel, the performance advantage remains stable across other common classical methods. Note that we also constructed engineered datasets saturating the geometric inequality between classical ML and the original quantum kernel, but the small geometric difference g presented no empirical advantage at large system size (see Supplementary Section 13).

In keeping with our arguments about the role of data, when we increase the number of training data N, all methods improve, and the advantage will gradually diminish. While this dataset is engineered, it shows the strongest empirical separation on the largest system size to date. We conjecture that this procedure could be used with a quantum computer to create challenging datasets that are easy to learn with a quantum device, hard to learn classically, while still being easy to verify classically given the correct labels. Moreover, the size of the margin implies that this separation may even persist under moderate amounts of noise in a quantum device.

Discussion

The use of quantum computing in machine learning remains an exciting prospect, but quantifying quantum advantage for such applications has some subtle issues that one must approach carefully. Here, we constructed a foundation for understanding opportunities for quantum advantage in a learning setting. We showed quantitatively how classical ML algorithms with data can become computationally more powerful, and a prediction advantage for quantum models is not guaranteed even if the data come from a quantum process that is challenging to independently simulate. Motivated by these tests, we introduced projected quantum kernels. On engineered datasets, projected quantum kernels outperform all tested classical models in prediction error. To the authors' knowledge, this is the first empirical demonstration of such a large separation between quantum and classical ML models.

This work suggests a simple guidebook for generating ML problems which give a large separation between quantum and classical models, even at a modest number of qubits. The size of this separation and trend up to 30 qubits suggests the existence of learning tasks that may be easy to verify, but hard to model classically, requiring just a modest number of qubits and allowing for device noise. Claims of true advantage in a quantum machine learning setting require not only benchmarking classical machine learning models, but also classical approximations of quantum models. Additional work will be needed to identify embeddings that satisfy the sometimes conflicting requirements of being hard to approximate classically and exhibiting meaningful signal on

local observables for very large numbers of qubits. Further research will be required to find use cases on datasets closer to practical interest and evaluate potential claims of advantage, but we believe the tools developed in this work will help to pave the way for this exciting frontier.

References

1. Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *IEEE Intell. Syst.* **24**, 8 (2009).
2. Grover, L. K. A fast quantum mechanical algorithm for database search. in *Proc. twenty-eighth annual ACM symposium on Theory of computing* (1996).
3. Durr, C. & Hoyer, P. A quantum algorithm for finding the minimum. https://arxiv.org/abs/quant-ph/9607014 (1996).
4. Farhi, E. et al. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science* **292**, 472 (2001).
5. Neven, H., Denchev, V. S., Rose, G. & Macready, W. G. *Training a large scale classifier with the quantum adiabatic algorithm.* https://arxiv.org/abs/0912.0779 (2009).
6. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **113**, 130503 (2014).
7. Leifer, M. S. & Poulin, D. Quantum graphical models and belief propagation. *Ann. Phys.* **323**, 1899 (2008).
8. Aaronson, S. & Ambainis, A. *The need for structure in quantum speedups.* https://arxiv.org/abs/0911.0996 (2009).
9. McClean, J. R. et al. *Low depth mechanisms for quantum optimization.* https://arxiv.org/abs/2008.08615 (2020).
10. Boixo, S. et al. Characterizing quantum supremacy in near-term devices. *Nat. Phys.* **14**, 595 (2018).
11. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505 (2019).
12. Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
13. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *N. J. Phys.* **18**, 023023 (2016).
14. Farhi, E. & Neven, H. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002* https://arxiv.org/abs/1802.06002 (2018).
15. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209 (2019).
16. Cortes, C. & Vapnik, V. Support-vector networks. *Mach. Learn.* **20**, 273 (1995).
17. Schölkopf, B. et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* https://mitpress.mit.edu/books/learning-kernels (2002).
18. Mohri, M., Rostamizadeh, A. & Talwalkar, A. *Foundations of machine learning.* https://mitpress.mit.edu/books/foundations-machine-learning-second-edition (2018).
19. Jacot, A., Gabriel, F. & Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *NIPS'18: Proceedings of the 32nd International Conference on Neural Information Processing Systems* https://dl.acm.org/doi/abs/10.5555/3327757.3327948 pp. 8580–8589 (2018).
20. Novak, R. et al. Neural tangents: Fast and easy infinite neural networks in python. *arXiv preprint arXiv:1912.02803* https://openreview.net/pdf?id=SklD9yrFPS (2019).
21. Arora, S. et al. On exact computation with an infinitely wide neural net. in *Advances in Neural Information Processing Systems* (2019).
22. Blank, C., Park, D. K., Rhee, J.-K. K. & Petruccione, F. Quantum classifier with tailored quantum kernel. *npj Quantum Inf.* **6**, 1 (2020).
23. Bartkiewicz, K. Experimental kernel-based quantum machine learning in finite feature space. *Sci. Rep.* **10**, 1 (2020).

24. Liu, Y., Arunachalam, S. & Temme, K. A rigorous and robust quantum speed-up in supervised machine learning. *arXiv preprint arXiv:2010.02174* https://arxiv.org/abs/2010.02174 (2020).

25. Huang, H.-Y., Kueng, R. & Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.* https://doi.org/10.1038/s41567-020-0932-7 (2020).

26. Cotler, J. & Wilczek, F. Quantum overlapping tomography. *Phys. Rev. Lett.* **124**, 100401 (2020).

27. Paini, M. and Kalev, A. An approximate description of quantum states. *arXiv preprint arXiv:1910.10543* https://arxiv.org/abs/1910.10543 (2019).

28. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J. & Killoran, N. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622* https://arxiv.org/abs/2008.08605 (2020).

29. Schuld, M., Sweke, R. & Meyer, J. J. The effect of data encoding on the expressive power of variational quantum machine learning models. *Phys. Rev. A* **103**, 032430 (2021).

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

31. Grant, E., Wossnig, L., Ostaszewski, M. & Benedetti, M. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum* **3**, 214 (2019).

32. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**, 032308 (2020b).

33. LaRose, R. & Coyle, B. Robust data encodings for quantum classifiers. *Phys. Rev. A* **102**, 032420 (2020).

34. Harrow, A. W. & Montanaro, A. Quantum computational supremacy. *Nature* **549**, 203 (2017).

35. Li, Z. et al. Enhanced convolutional neural tangent kernels. *arXiv preprint arXiv:1911.00809* https://arxiv.org/abs/1911.00809 (2019).

36. Micchelli, C. A., Xu, Y. & Zhang, H. Universal kernels. *J. Mach. Learn. Res.* **7**, 2651 (2006).

37. Krogh, A. & Hertz, J. A. A simple weight decay can improve generalization. *Adv. Neural Inf. Process. Syst.* 950–957 (1992).

38. Suykens, J. A. & Vandewalle, J. Least squares support vector machine classifiers. *Neural Process. Lett.* **9**, 293 (1999).

39. Huang, H.-Y., Kueng, R. & Preskill, J. Information-theoretic bounds on quantum advantage in machine learning. *arXiv preprint arXiv:2101.02464* https://arxiv.org/abs/2101.02464 (2021).

40. Anderson, E. et al. *LAPACK Users' Guide*, 3rd edn. (Society for Industrial and Applied Mathematics, 1999).

41. Breiman, L. Random forests. *Mach. Learn.* **45**, 5 (2001).

42. Gosset, D. & Smolin, J. A compressed classical description of quantum states. *arXiv preprint arXiv:1801.05721* https://arxiv.org/abs/1801.05721 (2018).

43. Aaronson, S. Shadow tomography of quantum states. *SIAM J. Comput.* https://dl.acm.org/doi/abs/10.1145/3188745.3188802 (2020).

44. Aaronson, S. and Rothblum, G. N. Gentle measurement of quantum states and differential privacy. in *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing* (2019).

45. Hohenberg, P. & Kohn, W. Inhomogeneous electron gas. *Phys. Rev.* **136**, B864 (1964).

46. Runge, E. & Gross, E. K. Density-functional theory for time-dependent systems. *Phys. Rev. Lett.* **52**, 997 (1984).

47. Broughton, M. et al. Tensorflow quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989* https://arxiv.org/abs/2003.02989 (2020).

48. Xiao, H., Rasul, K. & Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747* https://arxiv.org/abs/1708.07747 (2017).

49. LeCun, Y., Cortes, C. & Burges, C. *Mnist handwritten digit database*. http://yann.lecun.com/exdb/mnist (2010).

50. Jolliffe, I. T. in *Principal component analysis* 129–155 (Springer, 1986).

51. Schuld, M. & Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).

52. Skolik, A., McClean, J. R., Mohseni, M., van der Smagt, P. & Leib, M. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence* **3**, 5 (2021).

Acknowledgements

Author contributions

H.H. and J.M. developed the theoretical aspects of this work. H.H. and M.B. conducted the numerical experiments and wrote the open source code. H.H., M.M., R.B., S.B., H.N., and J.M. contributed to technical discussions and writing of the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-021-22539-9.

Correspondence and requests for materials should be addressed to J.R.M.

Peer review information *Nature Communications* thanks Nana Liu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.