The Born supremacy: quantum advantage and training of an Ising Born machine
Brian Coyle, Daniel Mills, Vincent Danos, Elham Kashefi

To cite this version:
Brian Coyle, Daniel Mills, Vincent Danos, Elham Kashefi. The Born supremacy: quantum advantage and training of an Ising Born machine. npj Quantum Information, 2020, 6 (1), pp.60. 10.1038/s41534-020-00288-9. hal-03096252

HAL Id: hal-03096252
https://hal.science/hal-03096252
Submitted on 4 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
The Born supremacy: quantum advantage and training of an Ising Born machine

Brian Coyle, Daniel Mills, Vincent Danos and Elham Kashefi

The search for an application of near-term quantum devices is widespread. Quantum machine learning is touted as a potential utilisation of such devices, particularly those out of reach of the simulation capabilities of classical computers. In this work, we study such an application in generative modelling, focusing on a class of quantum circuits known as Born machines. Specifically, we define a subset of this class based on Ising Hamiltonians and show that the circuits encountered during gradient-based training cannot be efficiently sampled from classically up to multiplicative error in the worst case. Our gradient-based training methods use cost functions known as the Sinkhorn divergence and the Stein discrepancy, which have not previously been used in the gradient-based training of quantum circuits, and we also introduce quantum kernels to generative modelling. We show that these methods outperform the previous standard method, which used maximum mean discrepancy (MMD) as a cost function, and achieve this with minimal overhead. Finally, we discuss the ability of the model to learn hard distributions and provide formal definitions for ‘quantum learning supremacy’. We also exemplify the work of this paper by using generative modelling to perform quantum circuit compilation.

npj Quantum Information (2020) 6:60; https://doi.org/10.1038/s41534-020-00288-9

INTRODUCTION

As quantum devices with ~80–200 qubits, but without fault tolerance, begin to be built, we near the dawn of the noisy intermediate scale quantum (NISQ) technology era. Because of the low number of qubits, the limited connectivity between them, and the low circuit depth permitted by low decoherence times, these devices cannot perform many of the most famous algorithms thought to demonstrate exponential speedups over classical algorithms.

In spite of this, NISQ devices could provide efficient solutions to other problems that cannot be solved in polynomial time by classical means. Showing this to be true is referred to as a demonstration of quantum computational supremacy, with the first such experimental realisation occurring recently.

Proposals for demonstrations of quantum computational supremacy on NISQ technology typically involve sampling from the output distribution of random quantum circuits. While a realisation of such an advantage is of great theoretical importance, generating random samples is not obviously independently interesting. We incorporate this sampling into a useful application, keeping the provable quantum advantage, but in a context with more practical applicability.

Specifically, we explore generative modelling in quantum machine learning (QML), which is the task of generalising from a finite set of samples, \( \{ y \} \), drawn from a data set. By learning the underlying probability distribution from which these samples are drawn, \( \hat{p}(y) \), a model should be able to generate new samples from the said distribution.

Generative models range from simple naive Bayes models to complicated neural networks, like generative adversarial networks (GANs). The intrinsic randomness inherent in quantum mechanics allows for the definition of a new class of generative models that are without a classical analogue. Known as Born machines, they have the ability to produce statistics according to Born’s measurement rule. Specifically, for a state \( | \psi \rangle \), a measurement produces a sample \( x \sim \rho(x) = | \langle x | \psi \rangle |^2 \). There are several variants, including Bayesian approaches, adversarial training methods, and adaptations to continuous distributions.

Quantum circuit Born machines (QCBM) are a subclass of parameterised quantum circuits (PQCs) and are widely applicable (see ref. for a review). PQCs consist of a quantum circuit which carries parameters that are updated during a training process (typically a classical optimisation routine). The circuit is kept as shallow as possible so as to be suitable for NISQ devices.

We ask in this work whether it is possible to have a machine learning application for a PQC, which comes with a provable superior performance over all classical alternatives on near term devices? Such provable guarantees are even more relevant given recent work in QML algorithm ‘dequantisation’.

We take the first steps in answering this question in several ways. We define a subclass of QCBM that we call Ising Born machines (QICBM). We improve the training of the model over previous methods, which use the maximum mean discrepancy (MMD) with a classical kernel, by introducing quantum kernels into the MMD, as well as by using entirely new cost functions: the Stein discrepancy (SD) and the Sinkhorn divergence (SHD). To do so, we derive their corresponding gradients in the quantum setting.

We show that these novel methods outperform the MMD with classical kernel by achieving a closer fit to the data as measured by the total variation (TV) distance. We derive forms of the SHD, Stein discrepancy, and Sinkhorn divergences that can either be efficient to compute or result in an upper bound on TV. We observe numerically that the SD provides an upper bound to TV. Next, we show that sampling from this model cannot be simulated efficiently by any classical randomised algorithm, up to multiplicative error in the worst case, subject to common assumptions in complexity theory (namely the non-collapse...
of the polynomial hierarchy). Furthermore, this holds for many circuit families encountered during training.

We define a framework in which a provable advantage could be demonstrated, which we refer to as quantum learning supremacy (QLS), and based on distribution learning theory\(^\text{38}\). Based on our classical sampling hardness results, we conjecture that the QCIBM may be a good candidate for a quantum model which could demonstrate this notion of learning supremacy; however, we leave the further investigation of QLS and its potential to be achieved by such models to future work. Finally, we provide a novel utilisation of such generative models in quantum circuit compilation.

## RESULTS

The main results of this work are new efficient gradient-based training methods and results on the hardness of simulating the model we introduce using classical computers. First, we define the model used and discuss its connection to previously studied quantum circuit families. We then discuss the efficient training of the model, first recalling a previously known gradient-based training method, which uses the MMD cost function, and then moving onto our new training methods, which use the SD and the SHD. We then discuss the SHD complexity in detail, and further argue, using its connection to the TV distance, why it should be used. We then prove the hardness results mentioned above, namely that many circuits encountered during gradient-based training are hard to classically simulate, before finally discussing the potential use of quantum generative models in learning distributions that are intractable to classical models. In addition, we provide a framework to study these advantages.

### Ising Born machine

Here we define the model we use for distribution learning. A generic quantum circuit Born machine consists of a parameterised quantum circuit, which produces samples by measuring the resulting quantum state, and a classical optimisation loop used to learn a data distribution. The circuits we study have the following structure:

\[
\begin{align*}
|0\rangle & \rightarrow H |0\rangle \\
|0\rangle & \rightarrow U_\gamma (|0\rangle) \\
\cdots & \\
|0\rangle & \rightarrow U_\gamma (\Gamma, \Delta, \Sigma) |0\rangle \\
& \cdots
\end{align*}
\]

where \(X_i\in\{0, 1\}\); the unitaries are defined by Eqs. (2) and (3); \(S_j\) indicates the subset of qubits on which each operator, \(j\), is applied; and a boldface parameter indicates a set of parameters, \(\alpha = (\alpha_j)\).

\[
U_\gamma (\alpha) := \prod_j U_\gamma (\alpha_j, S_j) = \prod_j \exp \left( i \theta_j \otimes Z_k \right)
\]

\[
U_\gamma (\Gamma, \Delta, \Sigma) := \exp \left( \sum_{k=1}^n \Gamma_{k} X_k + \Delta_{k} Y_k + \Sigma_{k} Z_k \right)
\]

The operators, \(X_k\), \(Y_k\), and \(Z_k\), are the standard Pauli operators acting on qubit \(k\). Restricting to the case \(|S_j| \leq 2\) (since only single and two-qubit gates are required for universal quantum computation), the term in the exponential of Eq. (2) becomes exactly an Ising Hamiltonian:

\[
\mathcal{H} := \sum_{i<j} J_{ij} Z_i Z_j + \sum_{k=1}^n b_k Z_k,
\]

where we are dividing the diagonal unitary parameters, \(\alpha = U_\gamma ^{b_j}\), into local terms that act only on qubit \(k\), \(b_j\), and coupling terms between two qubits \(i\) and \(j\), \(J_{ij}\). We call the model a QCIBM.

A measurement on all qubits in the computational basis results in sample vectors, \(x \in \chi^n\), where \(\chi = \{0, 1\}\). These samples are drawn from the distribution, \(p_\theta (x)\), parameterised by the set of angles, \(\theta = (\alpha, \Gamma, \Delta, \Sigma)\):

\[
p_\theta (x) := \left| \langle x | U_\gamma (\Gamma, \Delta, \Sigma) U_\gamma (\alpha) | + \rangle \right|^2.
\]

We denote the above model and parameters by QCIBM(\(\theta\)) := QCIBM(\(\alpha, \Gamma, \Delta, \Sigma\)). We choose this structure in order to easily recover two well-known circuit classes, namely instantaneous quantum polynomial time\(^{26}\) (IQP) circuits, and the shallowest depth (\(p = 1\)) version of the quantum approximate optimisation algorithm\(^{27}\) (QAOA).

IQP circuits are named to reflect the commuting nature of elements in the produce defining the unitary \(U_\gamma\), while QAOA\(^{27}\) was originally developed as an approximate version of the quantum adiabatic algorithm\(^{28}\). Both of these classes of circuits are known to be routes to demonstrate quantum supremacy\(^{4,6,25}\), and we extend this property here by using the results of ref.\(^{39}\). These classes can be recovered by setting the parameters of a QCIBM as follows:

\[
\text{IQP}(\{J_{ij}, b_j\}) = \text{QCIBM}(\{J_{ij}, b_j\}, \Gamma = \left\{ \frac{\pi}{2 \sqrt{2}} \right\}, \Sigma = \left\{ \frac{\pi}{2 \sqrt{2}} \right\})
\]

\[
\text{QAOA}_{\gamma^{-1}}(\{J_{ij}, b_j\}, \Gamma) = \text{QCIBM}(\{J_{ij}, b_j\}, \Gamma = -\Gamma, 0, 0).
\]

We denote, for example \(\left\{ \frac{\pi}{2 \sqrt{2}} \right\}\), to be all parameters of the \(n\) single qubit gates set to the same value, \(\pi/2\sqrt{2}\). We choose the final gate before the computational basis measurement to be in the form of Eq. (3), rather than the more common Euler decomposition of a single qubit gate decomposition found in the literature\(^{8,16}\). This is chosen to make the classical simulation hardness results more apparent in our proofs.

To recover IQP circuits, we simply need to generate the final layer of Hadamard gates (up to a global phase) and do so by setting \(U_\gamma\) in Eq. (3) as follows:

\[
U_\gamma ^{\text{IQP}} \left( \left\{ \frac{\pi}{2 \sqrt{2}} \right\}, 0, \left\{ \frac{\pi}{2 \sqrt{2}} \right\} \right) = \frac{1}{\sqrt{n}} \otimes e^{i \pi \xi \langle X_0 | Z_1 \rangle} = i H_0^{\text{IQP}}.
\]

To recreate depth 1 QAOA circuits, we need to set the Pauli \(Z\) and \(Y\) parameters, \(\Delta, \Sigma = 0\), since the final gates should be a product of Pauli-\(X\) rotations with parameters, \(-\Gamma\).

### Training the Ising Born machine

Here we introduce the alternative training methods that we use for our purposes and that would be applicable to any generative model. The training procedure is a hybrid of classical and quantum computation, with the only quantum component being the model itself. The remainder of the computation is classical, bringing our scheme into the realm of what is possible for NISQ devices. The procedure can be seen in Fig. 1.

The optimisation procedures we implement are stochastic gradient descents. Parameters, \(\theta_\alpha\), are updated at each epoch of training, \(d\), according to the rule \(\theta_\alpha \leftarrow \theta_\alpha - \eta \delta_\alpha \hat{L}_\alpha\).

The parameter \(\eta\) is the learning rate and controls the speed of the descent. The initial proposals to train QCBMs were gradient-free\(^{15,31}\), but gradient-based methods have also been proposed\(^{14,16,32}\). In this work, we advocate for increasing the classical computational power required in training to achieve better performance, rather than increasing the quantum resources, for example by adding extra ancillae\(^{16}\) or adding costly and potentially unstable (quantum) adversaries\(^{17,33,34}\).

For gradient-based methods, a cost function or metric is required, \(\hat{L}_\alpha(p_\theta (\mathbf{x}), \mathbb{P}(\mathbf{y}))\) to compare the Born Machine distribution, \(p_\theta (\mathbf{x})\), and the data distribution, \(\mathbb{P}(\mathbf{y})\). Good cost functions will...
Recent works\textsuperscript{37,38} on the near term advantage of using quantum computers in QML have explored quantum kernels, which can be evaluated on a quantum computer. To gain such an advantage, these kernels should be difficult to compute on a classical device. In particular, we will adopt the following kernel\textsuperscript{37} in which the samples are encoded in a quantum state, $|\phi(x)|$, via a feature map, $\phi(x) \rightarrow |\phi(x)|$. The kernel is the inner product between vectors:

$$k_{Q}(x,y) := |\langle \phi(x)|\phi(y) \rangle|^{2},$$

which can be calculated by measuring, in the computational basis, the state which results from running the circuit given by $U_{\phi(y)}$, followed by that of $U_{\phi(x)}$. This is seen in Fig. 1. The kernel, Eq. (14), is the observed probability of measuring the all-zero outcome, $0^{n}$. If this outcome is not observed after polynomially many measurements, the value of the kernel for this particular pair of samples $(x,y)$ is set to zero. Intuitively, this means the feature map has mapped the original points to points with at most exponentially small overlap in the Hilbert space and therefore will not contribute to the MMD.

It is also necessary to derive an expression for the gradient of the cost function. For the MMD, the gradient with respect to the $k$th parameter\textsuperscript{14}, carried by the $k$th unitary gate, $U_{k}(\theta_{k})$, is given by:

$$\frac{\partial C_{\text{MMD}}}{\partial \theta_{k}} = 2 \Im \langle (k(a,x)) - 2 \Im \langle (k(a,y)) + 2 \Re \langle (k(b,y)) \rangle_{y},$$

where $p_{k}^{\pm}$ are output distributions generated by running the following auxiliary circuits\textsuperscript{39,40} for each unitary gate, $U_{k}(\theta_{k})$:

$$|0\rangle^{\otimes n} \xrightarrow{H^{\otimes n}} U_{1:k-1} \xrightarrow{U_{k}(\theta_{k}^{\pm})} U_{k+1:l} \xrightarrow{U_{l}}$$

where $\theta_{k}^{\pm} := \theta_{k} \pm \pi/2$, and $U_{l} := U_{l+1} \ldots U_{m-1} U_{m}$ are the unitary gates in the Born machine. This gradient occurs because the form of the unitary gates in our case are exponential Pauli operators $U_{k}(\theta_{k}) = \exp(i\theta_{k} \Sigma_{k})$, with $\Sigma_{k}^{2} \equiv I$. With the unitaries in this form, the gradient of the probabilities outputted from the parameterised
state, with respect to a parameter \( \theta \), is given by\textsuperscript{14,40}:
\[
\frac{\partial p_\theta(z)}{\partial \theta} = p_{\theta_0}(z) - p_{\theta_0}(z).
\]

There is a slight difference between Eq. (17) and that of ref.\textsuperscript{14}, due to a different parameterisation of the unitaries above.

The gradients of the cost functions which we introduce next will also require the parameter-shifted circuits in Eq. (16). For more details on kernel methods and the MMD, see Supplementary Material Section II.

SD training
So far, we have only proposed a change of kernel in the MMD method of training QCIBMs. We now consider changing the cost function altogether. We endeavour to find costs which are efficient for computing for quantum models, yet stronger than MMD.

The first cost we propose is called the SD. SD has become popular for goodness-of-fit tests\textsuperscript{1}, i.e. testing whether samples come from a particular distribution or not, as opposed to the MMD, which is typically used for kernel two-sample tests\textsuperscript{36}. This discrepancy is based on Stein’s identity\textsuperscript{43}, which is a way to bound distance metrics between probability distributions including, for example, the other integral probability metrics (IPM) we utilise in this work. For details on IPMs, see Supplementary Material Section I.

We use the discrete version of the SD\textsuperscript{43} since, in its original form\textsuperscript{1}, it only caters for the case where the distributions are supported over a continuous space. The discretisation is necessary since the QCIBM outputs binary strings and so the standard gradient w.r.t. a sample, \( x, \nabla_w \), is undefined. As such, we need to use a discrete ‘shift’ operator, \( \Delta_x \), instead, which is an operator defined by \( \Delta_x f(x) := f(x) - f(x+1) \) for a function \( f \), where \( \neg \) flips the \( i \)th element of the binary vector \( x \).

Fortunately, the discretisation procedure is relatively straightforward (the necessary definitions and proofs can be found in Supplementary Material Section III). The discrepancy is derived\textsuperscript{41,44} from the (discrete) Stein identity\textsuperscript{43}, given by:
\[
E_{x \sim \pi}(A_x \phi(x)) = E_{x \sim \pi}(s_x \phi(x) - \Delta_x \phi(x)) = 0.
\]

\( A_x \phi(x) := s_x \phi(x) - \Delta_x \phi(x), \)

where \( E \) denotes the expectation value over the distribution, \( \pi \).

This holds for any function \( \phi : \lambda^n \rightarrow \mathbb{C} \), and probability mass function \( \pi \) on \( \lambda^n \). The function \( s_x \phi(x) = \Delta^x \log \pi(x) \) is the Stein score function of the distribution \( \pi \), and \( A_x \) is a so-called Stein operator of \( \pi \). Now, the SD cost function can be written in a kernelised form\textsuperscript{41,45}, similarly to the MMD:
\[
\mathcal{L}_{SD}(p_\theta, \pi) := E_{x \sim \pi} \left[ k_\theta(x, y) - k_\theta(x, y) \right],
\]

\[
k_\theta(x, y) := s_x \phi(x)^\top k(x, y) s_y \phi(y) - s_x \phi(x)^\top \Delta_x \phi(x) - \Delta_x \phi(x)^\top s_y \phi(y)
- \Delta_x \phi(x)^\top \Delta_y \phi(y) + \text{tr}(\Delta_x \phi(x)^\top \Delta_y \phi(y)),
\]

where \( k_\theta \) is the Stein kernel and \( k \) is a usual positive semi-definite kernel. \( \Delta^x \) is a conjugate version of the operator \( \Delta_x \) but for our purposes, the behaviour of both \( \Delta^x \) and \( \Delta_x \) are identical. For completeness, we define it in generality in Supplementary Material Section III.

Just as above, the gradient (derived in an identical fashion to the MMD gradient Eq. (15) as is detailed in Supplementary Material Section III) of \( \mathcal{L}_{SD} \) with respect to the parameter, \( \theta_k \), is given by:
\[
\frac{\partial \mathcal{L}_{SD}}{\partial \theta_k} = E_{x \sim p_\theta}[k_\theta(x, y)] - E_{x \sim \pi}[k_\theta(x, y)] - E_{x \sim \pi}[k_\theta(x, y)] - E_{x \sim \pi}[k_\theta(x, y)].
\]

\[
\frac{\partial \mathcal{L}_{SD}}{\partial \theta_k} = E_{x \sim \pi} [\phi(x)] - E_{x \sim \pi} [\phi(x)].
\]

We show that almost every term in Eqs. (20) and (22) can be computed efficiently, even when the quantum kernel \( k_\theta \) from Eq. (13) is used in Eq. (21), that is, with the exception of the score function \( s_\pi \) with respect to the data distribution. The score contains an explicit dependence on the data distribution, \( \pi \). If we are given oracle access to the probabilities, \( \pi(y) \), then there is no issue and SD will be computable. Unfortunately, in any practical application this will not be the case.

To deal with such a scenario, we give two approaches to approximate the score via samples from \( \pi \). The first of these we call the ‘identity’ method since it inverts Stein’s identity\textsuperscript{43} from Eq. (18). We refer to the second as the ‘Spectral’ method since it uses a spectral decomposition\textsuperscript{49} of a kernel to approximate the score. The latter approach uses the Nyström method\textsuperscript{37}, which is a technique used to approximately solve integral equations. We will only use the Spectral method in training the QCIBM in the numerical results in Fig. 3, since the Identity method does not give an immediate out-of-sample method to compute the score. Details of these methods can be found in Supplementary Material Section III.

Notice that, even with the difficulty in computing the score, the SD is still more suitable for training these models than the KL divergence as the latter requires computing the circuit probabilities, \( p_\theta(x) \), which is in general intractable, and so could not be computed for any data set.

SHD training
The second cost function we consider is the so-called SHD. This is a relatively new method to compare probability distributions\textsuperscript{48—49}, defined by the following:
\[
\mathcal{L}^n_{SHD}(p_\theta, \pi) := \mathcal{L}^n_{OT}(p_\theta, \pi) - \frac{1}{2} \mathcal{L}^n_{OT}(p_\theta, p_\theta) - \frac{1}{2} \mathcal{L}^n_{OT}(\pi, \pi),
\]

\[
\mathcal{L}^n_{OT}(p_\theta, \pi) := \min_{U(\lambda^n \times \lambda^n)} \sum_{x \sim \pi, y \sim \pi} C(x, y) U(x, y) + \epsilon KL(U|p_\theta \otimes \pi)
\]

where \( \epsilon \geq 0 \) is a regularisation parameter, \( C(x, y) \) is a Lipschitz ‘cost’ function, and \( U(\pi, \pi) \) is the set of all couplings between \( p_\theta \) and \( \pi \), i.e. the set of all joint distributions, whose marginals with respect to \( x \) and \( y \) are \( p_\theta(x) \) and \( \pi(y) \), respectively. The above cost function, \( \mathcal{L}^n_{SHD} \), is particularly favourable as a candidate because of its relationship to the theory of optimal transport\textsuperscript{48} (OT), a method to compare probability distributions. It has become a major tool used to train models in the classical domain, for example with GANs\textsuperscript{22} through a restriction of OT called the Wasserstein metric, which is derived from OT, when the cost \( c(x, y) \) is chosen to be a metric on the space of \( \lambda^n \).

We would like to use OT itself to train generative models, due to its metric properties. Unfortunately, OT has high computational cost and exponential sample complexity\textsuperscript{53}. For this reason, the SHD was proposed in refs\textsuperscript{48—50} to interpolate between OT and the MMD as a function of the regularisation parameter \( \epsilon \) in Eq. (24). In particular, for the two extreme values of \( \epsilon \), we recover\textsuperscript{55} both unregularised OT and the MMD:
\[
\epsilon \rightarrow 0 : \mathcal{L}^n_{SHD}(p_\theta, \pi) \rightarrow \mathcal{L}^n_{OT}(p_\theta, \pi) \cdot \epsilon \rightarrow \infty : \mathcal{L}^n_{SHD}(p_\theta, \pi) \rightarrow \mathcal{L}^n_{MMD}(p_\theta, \pi) : k(x, y) = -c(x, y).
\]

As before, we need a gradient of the \( \mathcal{L}^n_{SHD} \) with respect to the parameters, which is given by:
\[
\frac{\partial \mathcal{L}^n_{SHD}(p_\theta, \pi)}{\partial \theta_k} = E_{x \sim p_\theta} [\phi(x)] - E_{x \sim \pi} [\phi(x)].
\]
where \( g(x) \) is a function that depends on the optimal solutions found to the regularised OT problem in Eq. (24). See Supplementary Material Section IV for more details on the SHD and its gradient.

Sinkhorn complexity

The sample complexity of the SHD is of great interest to us as we claim that the TV and the KL are not suitable to be directly used as cost functions. This is due to the difficulty of computing the outcome probabilities of quantum circuits efficiently. We now analyse why the MMD is a weak cost function and why the SHD should be used as an alternative. This will depend critically on the regularisation parameter \( \epsilon \), which allows a smooth interpolation between the OT metric and the MMD.

First, we address the computability of \( L_{\text{SHD}}^c \) and we find, due to the results of ref. 54, a somewhat ‘optimal’ value for \( \epsilon \), for which the sample complexity of \( L_{\text{SHD}}^c \) becomes efficient. Specifically, the mean error between \( L_{\text{SHD}}^c \) and its approximation \( L_{\text{SHD}}^c \) for \( n \) qubits, computed using \( M \) samples, scales as:

\[
E[|L_{\text{SHD}}^c - L_{\text{SHD}}^c|] = O\left(\frac{1}{\sqrt{M}} \left(1 + e^{\frac{c^2}{2\epsilon}}\right)\left(1 + \frac{1}{e^{\epsilon/2}}\right)\right).
\]

We show in Supplementary Material Section IV.1 that by choosing \( \epsilon = O(n^2) \), we get:

\[
E[|L_{\text{SHD}}^{O(n^2)} - L_{\text{SHD}}^{O(n^2)}|] = O\left(\frac{1}{\sqrt{M}}\right),
\]

which is the same sample complexity as the MMD but exponentially better than that of unregularised OT, which scales as \( O(1/M^{1/n}) \). A similar result can be derived using a concentration bound54, such that, with probability \( 1 - \delta \),

\[
|L_{\text{SHD}}^{O(n^2)} - L_{\text{SHD}}^{O(n^2)}| = O\left(\frac{n}{\sqrt{M}} \log(1/\delta)^{1/2}\right).
\]

where we have chosen the same scaling for \( \epsilon \) as in Eq. (29).

Therefore, we can choose an optimal theoretical value for the regularisation, such that \( L_{\text{SHD}}^c \) is sufficiently far from OT to be efficiently computable but perhaps still retains some of its favourable properties. It is likely in practice, however, that a much lower value of \( \epsilon \) could be chosen without a blow up in sample complexity54,55. See Supplementary Material Section IV for derivations of the above results.

Second, we can relate the \( L_{\text{SHD}}^c \) to unregularised OT and TV via a sequence of inequalities. We have mentioned that the MMD is weak, meaning it provides a lower bound on TV in the following sequence of inequalities. We have mentioned that the MMD is hence stronger than the MMD but

\[
\text{TV}(\rho_0, \rho_1) \leq OT_0^c(\rho_0, \rho_1) \leq n \cdot \text{TV}(\rho_0, \rho_1).
\]

Finally, we can examine the relationship induced by the regularisation parameter through the following inequality: Theorem 1 in ref. 54:

\[
0 \leq OT^c(\rho_0, \rho_1) - OT^c(\rho_0, \rho_1) \leq 2\log\left(\frac{e^2LD}{nC}\right) \approx \sqrt{n} \log(1/e).
\]

where the size of the sample space is bounded by \( D \), as measured by the metric, and \( L \) is the Lipschitz constant of the cost \( c \). As detailed in Supplementary Material Section IV.1, we can choose \( D = n \) and \( L = n \):

\[
0 \leq OT^c(\rho_0, \rho_1) - OT^c(\rho_0, \rho_1) \leq 2\log\left(\frac{e^2n}{\epsilon}\right).
\]

The log term will be positive as long as \( \epsilon \leq ne^2 \), in which case regularised OT will give an upper bound for the Wasserstein metric, and hence the TV through Eq. (34), so we arrive at:

\[
\text{TV}(\rho_0, \rho_1) \leq OT^c(\rho_0, \rho_1) \leq OT^c(\rho_0, \rho_1) \leq \text{TV}(\rho_0, \rho_1).
\]

Unfortunately, comparing this with Eqs. (29) and (30), we can see that, with this scaling of \( \epsilon \), the sample complexity would pick up an exponential dependence on the dimension, \( n \), so it would not be efficiently computable. We comment further on this point later.

Numerical performance

In Figs 2–4, we illustrate the superior performance of our alternative training methods, as measured by the TV distance. A lower TV indicates that the model is able to learn parameters which fit the true data more closely. TV was chosen as an objective benchmark for several reasons. First, it is typically the notion of distance that is required by quantum supremacy experiments where one wants to prove hardness of classical simulation. Second, we use it in the definitions of QLS. Finally, it is one of the strongest notions of convergence in probability one can ask for, so it follows that a training procedure that can more effectively minimise TV, in an efficient way, should be better for generative modelling.

We train the model on Rigetti’s Forest platform57 using both a simulator and the real quantum hardware, the Aspen QPU. Figure 2 illustrates the training of the model using the Gaussian (Eq. (12)) vs. the quantum kernel (Eq. (13)) for 4 qubits, and we see that the Gaussian kernel, as measured by TV in Fig. 3. Furthermore, we observed that the gap (highlighted in the inset in Fig. 3a) which separates the SHD and SD (red and blue lines) from the MMD (green, yellow, and cyan lines) grows as the number of qubits grows. Unfortunately, the Spectral method to approximate the
Stein score does not outperform the MMD, despite training successfully. The discrepancy between the true and approximate versions of the Stein score is likely due to the low number of samples used to approximate the score, with the number of samples limited by the computational inefficiency. We leave tuning the hyperparameters of the model in order to get better performance to future work.

This behaviour is shown to persist on the QPU, Fig. 4, where we show training of the model with both the MMD and SHD relative to TV, (Fig. 4a), the learned probabilities of both methods on, Fig. 3. MMD (green, yellow) vs. Sinkhorn (blue, cyan) for 4 qubits comparing performance on the real QPU (Aspen 4Q-A) vs. simulated behaviour on QVM (Aspen 4Q-A-qvm) using 500 samples and a batch size of 250, learning target data (black), and an initial learning rate for Adam as $\eta_{\text{init}} = 0.01$. a TV difference achieved with both kernel methods during training. No observable or obvious advantage is seen in using the quantum kernel over the Gaussian one; in contrast, the Gaussian kernel seems to perform better on average. b Final learned probabilities with $\eta_{\text{init}} = 0.01$ using the Adam optimiser. c MMD computed using 400 samples as training points and 100 as test points (seen as the thin lines without markers), independent of the training data.
off, the QPU (Fig. 4b), and the behaviour of the cost functions associated with both methods (Fig. 4c, d). This reinforces our theoretical argument that the SHD is able to better minimise TV to achieve superior results.

Given the performance noted above, we would recommend the SHD as the primary candidate for future training of these models, due to its simplicity and competitive performance. One should also note that we do not attempt to learn these data distributions exactly since we use a shallow fixed circuit structure for training (i.e., a QAOA circuit), which we do not alter. Better fits to the data could likely be achieved with deeper circuits with more parameters.

For extra numerical result demonstrating the performance of the learning algorithms, see Supplementary Material Section V, including a comparison between the quantum and Gaussian kernels for two qubits, similar to Fig. 2; the behaviour of the corresponding cost functions themselves associated with Fig. 3; the performance of the model for 4 qubits, similar to Fig. 3; and the results using a 3 qubit device, the Aspen–4–3Q–A. In all cases, the performance was qualitatively similar to that reported in the main text.

Hardness and quantum advantage

It is crucially important, not just for our purposes but for the design of QML algorithms in general, that the algorithm itself is providing some advantage over any classical one for the same task. This is the case for so-called coherent algorithms, like the HHL linear equation solver, which is BQP-complete, and therefore unlikely to be fully dequantised. However, such a proven quantum supremacy, by providing a quantum circuit close to , then the resulting circuit has the same form. This holds also for the gradient-shifted circuits in Eq. (16) since these correspond to circuits whose parameters are updated as follows: .

We now provide definitions to meet the requirements of point 2, adapting definitions from distribution learning theory for this purpose. Specifically, we say that a generative QML algorithm, with a small abuse of notation) has demonstrated QLS if there exists a class of probability distributions over (bit vectors of length ), for which there exists a metric and a fixed such that is (d, e, BQP)-learnable via but not (d, e, BPP)-learnable (i.e. learnable by a purely classical algorithm). The task of the learning algorithm is, given a target distribution output, with high probability, a Generator, such that ‘s close to with respect to the metric . For the precise definitions of learnability we employ, see Supplementary Material Section VII.

This framework is very similar to that of , and inspired by, probably approximately correct (PAC) learning, which has been well studied in the quantum case, but it applies more closely to the task of generative modelling. It is known that, in certain cases, the use of quantum computers can be beneficial to PAC learning but not generically. Based on this, it is possible that there exist some classes of distributions that cannot be efficiently learned by classical computers (BPP algorithms) but that could be learned by quantum devices (BPO algorithms). The motivation for this is exactly rooted in the question of quantum supremacy and illustrated crudely in Fig. 5b.

An initial attempt at QLS is as follows. As mentioned above, if random IQP circuits could be classically simulated to within a TV error of in the worst case (with high probability over the choice of circuit), this would imply unlikely consequences for complexity theory. Now, if a generative quantum model was able to achieve a closeness in TV less than this constant value, perhaps by minimising one of the upper bounds in Eq. (38), then we could claim that this model had achieved something classically intractable. For example, if we make the following assumptions,

1. QGIBM could achieve a TV < to a target IQP distribution.
2. A classical probabilistic algorithm, could output a distribution in polynomial time which was close in TV to the QCIBM, i.e. it could simulate it efficiently.

We can address point 1 informally (see Supplementary Material Section VI for the formal statements and proof) in three steps:

- If the parameters of the model are initialised randomly in and final measurement angles are chosen such that , then the resulting QCIBM circuit class will be hard to simulate up to an additive error of in TV distance, subject to a conjecture relating to the hardness of computing the Ising partition function.
- If certain configurations of the parameters are chosen to be either of the form, 1 = 1/384 in the worst case.
- The circuits produced at each epoch as a result of the gradient updates will each result in a hard circuit class as long as the gradient updates are not chosen carelessly. In each epoch, if the update step is constrained in a way that the new value of the parameter does not become rational, then the updated circuits will also belong to a class that is hard to simulate (a similar result can be shown for the case where the parameters are updated to keep within the form of).

We now provide definitions to meet the requirements of point 2, adapting definitions from distribution learning theory for this purpose. Specifically, we say that a generative QML algorithm, has demonstrated QLS if there exists a class of probability distributions over (bit vectors of length), for which there exists a metric and a fixed such that is (d, e, BQP)-learnable via but not (d, e, BPP)-learnable (i.e. learnable by a purely classical algorithm). The task of the learning algorithm is, given a target distribution to output, with high probability, a Generator, such that ‘s close to with respect to the metric . For the precise definitions of learnability we employ, see Supplementary Material Section VII.

This framework is very similar to that of, and inspired by, probably approximately correct (PAC) learning, which has been well studied in the quantum case, but it applies more closely to the task of generative modelling. It is known that, in certain cases, the use of quantum computers can be beneficial to PAC learning but not generically. Based on this, it is possible that there exist some classes of distributions that cannot be efficiently learned by classical computers (BPP algorithms) but that could be learned by quantum devices (BPO algorithms). The motivation for this is exactly rooted in the question of quantum supremacy and illustrated crudely in Fig. 5b.

An initial attempt at QLS is as follows. As mentioned above, if random IQP circuits could be classically simulated to within a TV error of in the worst case (with high probability over the choice of circuit), this would imply unlikely consequences for complexity theory. Now, if a generative quantum model was able to achieve a closeness in TV less than this constant value, perhaps by minimising one of the upper bounds in Eq. (38), then we could claim that this model had achieved something classically intractable. For example, if we make the following assumptions,

1. QGIBM could achieve a TV < to a target IQP distribution.
2. A classical probabilistic algorithm, could output a distribution in polynomial time which was close in TV to the QCIBM, i.e. it could simulate it efficiently.

We now provide definitions to meet the requirements of point 2, adapting definitions from distribution learning theory for this purpose. Specifically, we say that a generative QML algorithm, has demonstrated QLS if there exists a class of probability distributions over (bit vectors of length), for which there exists a metric and a fixed such that is (d, e, BQP)-learnable via but not (d, e, BPP)-learnable (i.e. learnable by a purely classical algorithm). The task of the learning algorithm is, given a target distribution to output, with high probability, a Generator, such that ‘s close to with respect to the metric . For the precise definitions of learnability we employ, see Supplementary Material Section VII.

This framework is very similar to that of, and inspired by, probably approximately correct (PAC) learning, which has been well studied in the quantum case, but it applies more closely to the task of generative modelling. It is known that, in certain cases, the use of quantum computers can be beneficial to PAC learning but not generically. Based on this, it is possible that there exist some classes of distributions that cannot be efficiently learned by classical computers (BPP algorithms) but that could be learned by quantum devices (BPO algorithms). The motivation for this is exactly rooted in the question of quantum supremacy and illustrated crudely in Fig. 5b.

An initial attempt at QLS is as follows. As mentioned above, if random IQP circuits could be classically simulated to within a TV error of in the worst case (with high probability over the choice of circuit), this would imply unlikely consequences for complexity theory. Now, if a generative quantum model was able to achieve a closeness in TV less than this constant value, perhaps by minimising one of the upper bounds in Eq. (38), then we could claim that this model had achieved something classically intractable. For example, if we make the following assumptions,

1. QGIBM could achieve a TV < to a target IQP distribution.
2. A classical probabilistic algorithm, could output a distribution in polynomial time which was close in TV to the QCIBM, i.e. it could simulate it efficiently.
Fig. 5  Quantum learning supremacy in distribution learning. a Illustration of a learning procedure using a Generator. The algorithm \( \mathcal{A} \) is given access to \( \text{GEN}_D \), which provides samples, \( x \sim D \), and must output a Generator for a distribution that is close to the original. We allow the target generator to be classical, hence it may take as input a string of random bits of size polynomial in \( n, \log n \), if not able to generate its own randomness. b Crude illustration of quantum learning supremacy. No classical algorithm, \( C \), should be able to achieve the required closeness in total variation to the target distribution, but the QCIBM (or similar) should be able to, for some class of target distributions. There should be some path in the parameter space of the QCIBM, \( \theta \), which achieves this.

where the third line follows from the triangle inequality. Therefore, \( C \) could simulate an IQP distribution also, and we arrive at a contradiction.

The major open question left by this work is whether QLS is possible at all; can a quantum model outperform all classical ones in generative learning? This idea motivated our search for metrics in generative learning? This idea motivated our search for metrics possible at all; can a quantum model outperform all classical ones.

Quantum compiling

As a concrete application of such a model outside the scope of classical generative modelling, we can use the QCIBM training to perform a type of ‘weak’ quantum circuit compilation. There are potentially other areas that could be studied using these tools or by applying techniques in generative modelling to other quantum information processing tasks, but this is beyond the scope of this work.

The major objective in this area is to compile a given target unitary, \( U \), into one that consists exclusively of operations available to the native hardware of the quantum computer in question. For example, in the case of Rigetti’s Aspen QPU, the native gates are \( R_x(\pm \pi/2), R_y(\theta), CZ \) \(^{67,68} \), and any unitary which a user wishes to implement must be compiled into a unitary \( V \) that contains only these ingredients.

Potential solutions to this problem \(^{69,70} \) involve approximating the target unitary by assuming that \( V \) is a parametric circuit built from the native gates, which can be trained by some optimisation strategy. We adopt a similar view here, but we do not require any extra quantum resources to perform the compilation. With this limitation, we make a trade-off in that we are not guaranteed to apply the same target unitary, only that the output distribution will be close to that produced by the target. Clearly this is a much weaker constraint than the task of direct compilation, since many unitaries may give rise to the same distribution, but it is much closer to the capabilities of near term devices. To illustrate this application, we train an QCIBM to learn the output distribution of a random IQP circuit when restricted to a QAOA architecture itself using \( \text{SHD} \) as a cost function. The process is illustrated in Eq. (43), where we try to determine suitable QAOA parameters, \( \{ f_0, b_0 \} \), which reproduce the distribution observed from a set of random IQP parameters, \( \{ f_k, b_k \} \).

\[
\text{QCIBM} \left( \{ f_0, b_0 \}, \{ \Gamma_k = \frac{\pi}{4}, 0 \} \right) \xrightarrow{\text{Compile}} \text{QCIBM} \left( \{ f_k, b_k \}, \{ \Gamma_k = \frac{\pi}{4}, \Sigma_k = \frac{\pi}{4} \} \right). \tag{43}
\]

The measurement unitary at the end of the circuit makes this process non-trivial, since this will give rise to significantly different distributions, even given the same parameters in \( U \). We illustrate this in Fig. 6 using the Rigetti 2q-qvm and for three qubits in Supplementary Material Section V. We find that, even though the learned parameter values are different from the target, the resulting distributions are quite similar, as expected.

DISCUSSION

Providing provable guarantees of the superior performance of near term quantum computers relative to any classical device for some particular non-trivial application is an important milestone of the field. We have shown one potential route towards this goal...
by combining complexity–theoretic arguments$^{4,7,9}$ with an application in generative machine learning$^{14–16,62}$, and improved training methods of generative models. Specifically, we introduced the Ising Born machine, a restricted form of a quantum circuit Born machine. These models utilise the Born rule of quantum mechanics to train a parameterised quantum circuit as a generative machine learning model, in a hybrid manner.

We proved that the model cannot be simulated efficiently by any classical algorithm up to a multiplicative error in the output probabilities, which holds for many circuit families that may be encountered during gradient-based training. As such, this type of model is a good candidate for a provable quantum advantage in QML using NISQ devices. To formalise this intuition, we defined a notion of QLS to rigorously define what such an advantage would look like, in the context of machine learning.

We adapted novel training methods for generative modelling in two ways. First, by introducing quantum kernels to be evaluated on the quantum hardware and, second, by proposing and adapting new cost functions. In the case of the SD, we discussed its sample complexity and used this to define a somewhat optimal cost function through a judicious choice of the regularisation parameter. It is possible to choose this parameter such that the cost is efficiently computable even as the number of qubits grows. We showed numerically that these methods have the ability to outperform previous methods in the random data set we used as a test case.

Finally, we demonstrated an application of the model as a heuristic compiler to compile one quantum circuit into another via classical optimisation techniques, which has the advantage of requiring minimal quantum overhead. These techniques could potentially be adapted into methods to benchmark and verify near term quantum devices.

The major question that this work raises is whether or not a provable notion of quantum learning could be achievable for a particular data set, thereby solidifying a use case for quantum computers in the near term with provable advantage. The best prospect for this is the quantum supremacy distributions we know of (for example, IQP), but they are not efficiently testable$^{63}$. Owing to this, they are also likely to not be efficiently learnable either, given the close relationship between distribution testing and learning$^{84}$. Furthermore, we can see from the exponential scaling required in Eq. (38) for regularised OT to upper bound TV that other techniques are necessary to achieve QLS, since the methods we present here are not suited to this particular task, despite achieving our goal of being stronger than the MMD for generative modelling. However, this assumes that we have access to classical samples from the distribution, and the possibility of gaining an advantage using quantum samples$^{38,59}$ is unexplored in the context of distribution learning.

**METHODS**

In this section, we detail the methods used to train the QCIBM to reproduce a given probability distribution. The target distribution is the one given by Eq. (44), which is used in both refs$^{65,66}$ to train versions of the quantum Boltzmann machine:

$$p(n) := \frac{1}{Z} \prod_{k=1}^{n} e^{-\beta_k s_k(n)} \frac{1-p}{1-\beta_k s_k(n)} \eta_{\text{bin}}(n),$$

(44)

to generate this data, $7$ binary strings of length $n$, written $s_k$ and called ‘modes’, are chosen randomly. A sample $y$ is then produced with a probability that depends on its Hamming distance $d_h(s_k, y)$ to each mode. In all of the above, the Adam$^{77}$ optimiser was applied, using the suggested hyperparameters, i.e. $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-4}$, and initial learning rate, $\eta_{\text{bin}}$. This was chosen since it was found to be more robust to sampling noise$^{14}$.

In all of the numerical results, we used a QAOA structure as the underlying circuit in the QCIBM. Specifically, the parameters in $U_i$ were chosen such that $Y_k$, $\Gamma_k = \pi/4$, $\Delta_k = 0$, and $\Sigma_k = 0$. The Ising parameters $U_i$, $b_i$ were initialised randomly.

For the SD, we used $3$ Nyström eigenvectors to approximate the Spectral score in Fig. 3 for $3$ qubits, and $6$ eigenvectors for $4$ qubits. In all cases when using the MMD with a Gaussian kernel, we chose the bandwidth parameters, $\sigma = [0.25, 10, 1000]$. Note that this article was previously published as a preprint$^{78}$.

**DATA AVAILABILITY**

Data and simulations presented in this work are available from the corresponding author upon request. Code used in this work is available from the corresponding author upon request or on Github$^{79}$.

Received: 29 August 2019; Accepted: 12 May 2020; Published online: 08 July 2020

**REFERENCES**

1. Preskill, J. Quantum computing in the NISQ era and beyond. Quantum 2, 79 (2018).
2. Shor, P. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26, 1484–1509 (1997).
3. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. Phys. Rev. Lett. 103, 150502 (2009).
4. Brenmer, M. J., Joza, R. & Shepherd, D. J. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. Proc. R. Soc. Lond. A 467, 459–472 (2011).
5. Gao, X., Wang, S.-T. & Duan, L.-M. Quantum supremacy for simulating a translation-invariant Ising spin model. Phys. Rev. Lett. 118, 040502 (2017).
6. Brenmer, M. J., Montanaro, A. & Shepherd, D. J. Average-case complexity versus approximate simulation of commuting quantum computations. Phys. Rev. Lett. 117, 080501 (2016).
7. Aaronson, S. & Arkhipov, A. The computational complexity of linear optics. Theory Comput. 9, 143–252 (2013).
8. Farhi, E. & Harrow, A. W. Quantum supremacy through the quantum approximate optimization algorithm. Preprint at http://arxiv.org/abs/1602.07674 (2016).
9. Boixo, S. et al. Characterizing quantum supremacy in near-term devices. Nat. Phys. 14, 595–600 (2018).
10. Arute, F. et al. Quantum supremacy using a programmable superconducting processor. Nature 574, 505–510 (2019).
11. Maron, M. E. Automatic indexing: an experimental inquiry. J. ACM 8, 404–417 (1961).
12. Goodfellow, I. J. et al. Generative Adversarial Nets. In Advances in Neural Information Processing Systems 27 (eds Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D. & Weinberger, K. Q.), pp. 2672–2680 (Curran Associates, Inc., 2014).
13. Cheng, S., Chen, J. & Wang, L. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. Entropy 20, 583 (2018).
14. Liu, J.-G. & Wang, L. Differentiable learning of quantum circuit Born machines. Phys. Rev. A 98, 062324 (2018).
15. Benedetti, M. et al. A generative modeling approach for benchmarking and training shallow quantum circuits. npj Quantum Inf. 5, 1–9 (2019).
16. Du, Y., Hsieh, M.-H., Liu, T. & Tao, D. The expressive power of parameterized quantum circuits. Preprint at http://arxiv.org/abs/1810.11922 (2018).
17. Zeng, J., Wu, Y., Liu, J.-G., Wang, L. & Hu, J. Learning and inference on generative adversarial quantum circuits. Phys. Rev. A 99, 052306 (2018).
18. Romero, J. & Aspuru-Guzik, A. Variational quantum generators: generative adversarial quantum machine learning for continuous distributions. Preprint at http://arxiv.org/abs/1901.00848 (2019).
19. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. Quantum Sci. Technol. 4, 043001 (2019).
20. Tang, E. Quantum-inspired classical algorithms for principal component analysis and supervised clustering. Preprint at http://arxiv.org/abs/1811.00411 (2018).
21. Tang, E. A quantum-inspired classical algorithm for recommendation systems. In Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, 217–228 (2019).
22. Andoni, A., Krauthgamer, R. & Pogrow, Y. On solving linear systems in sublinear time. Preprint at http://arxiv.org/abs/1809.02995 (2018).
23. Chia, N.-H., Lin, H.-H. & Wang, C. Quantum-inspired sublinear classical algorithms for solving low-rank linear systems. Preprint at http://arxiv.org/abs/1811.04852 (2018).
24. Gilýen, A., Lloyd, S. & Tang, E. Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension. Preprint at http://arxiv.org/abs/1811.04909 (2018).
25. Kearns, M. et al. On the learnability of discrete distributions. In Proc. Twenty-sixth Annual ACM Symposium on Theory of Computing 273–282 (ACM, New York, NY, 1994).

26. Shepherd, D. & Bremner, M. J. Temporally unstructured quantum computation. Proc. R. Soc. A. https://doi.org/10.1098/rspa.2008.0443 (2009).

27. Farhi, E., Goldstone, J. & Gutmann, S. A Quantum approximate optimization algorithm. Preprint at https://arxiv.org/abs/1411.4028 (2014).

28. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. Preprint at http://arxiv.org/abs/quant-ph/0001106 (2000).

29. Bremner, M. J., Montanaro, A. & Shepherd, D. Achieving quantum supremacy with sparse and noisy commuting quantum computations. Quantum 1, 8 (2017).

30. Fujii, K. & Morimae, T. Commuting quantum circuits and complexity of ising partition functions. New J. Phys. 19, 033003 (2017).

31. Leyton-Ortega, V., Perdomo-Ortiz, A. & Perdomo, O. Robust implementation of generative modeling with parameterized quantum circuits. Preprint at http://arxiv.org/abs/1901.08047 (2019).

32. Hamilton, K. E., Dumitrescu, E. F. & Poozer, R. C. Generative model benchmarks for superconducting qubits. Phys. Rev. A 99, 062323 (2019).

33. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. Phys. Rev. Lett. 121, 040502 (2018).

34. Dallaire-Demers, P.-L. & Killoran, N. Quantum generative adversarial networks. Phys. Rev. A 98, 012324 (2018).

35. Borgwardt, K. M. et al. Integrating structured biological data by kernel maximum entropy. Bioinformatics 22, e49–e57 (2006).

36. Gretton, A., Borgwardt, K. M., Rasch, M., Schölkopf, B. & Smola, A. J. A kernel method for two-sample-problems. In Advances in Neural Information Processing Systems 19 (eds. Schölkopf, B., Platt, J. C. & Hoffman, T.) 513–520 (MIT Press, 2007).

37. Hlavíček, V. et al. Supervised learning with quantum-enhanced feature spaces. Nature 567, 209–212 (2019).

38. Schuld, M. & Petruccione, F. Supervised Learning with Quantum Computers. Quantum Science and Technology (Springer International Publishing, 2018).

39. Mittarai, K., Negoro, M., Kitagawa, M. & Fuji, K. Quantum circuit learning. Phys. Rev. A 98, 032309 (2018).

40. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. Phys. Rev. A 99, 023321 (2019).

41. Liu, Q., Lee, J. D. & Jordan, M. A Kernelized Stein Discrepancy for Goodness-of-fit Tests. In Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48, 276–284 (PMLR, New York, NY, 2016).

42. Stein, C. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In Proc. Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory. 583–602 (University of California Press, Berkeley, CA, 1972).

43. Dudley, R. M. The speed of mean Glivenko-Cantelli convergence. Ann. Math. Stat. 40, 40–50 (1969).

44. Genevay, A., Chizat, L., Bach, F., Cuturi, M. & Peyré, G. Sample complexity of Sinkhorn divergences. In Proceedings of Machine Learning Research 89, 1574–1583 (eds Chaudhuri, K. & Sugiyama, M.). (PMLR, 2019).

45. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Quantum generative adversarial networks using low-depth circuits. Preprint at http://arxiv.org/abs/1901.08047 (2019).

46. Gibbs, A. L. & Su, F. E. On choosing and bounding probability metrics. Int. Stat. Rev. 79, 419–433 (2002).

47. Smith, R. S., Curtis, M. J. & Zeng, W. J. A practical quantum instruction set architecture. Preprint at https://arxiv.org/abs/1608.03355 (2016).

48. Arjovsky, M., Chintala, S. & Bottou, L. Wasserstein GAN. In Proceedings of the 34th International Conference on Machine Learning, Proceedings of Machine Learning Research 70, 214–223 (eds Doina, P. & Yee, W. T.), International Convention Centre, Sydney, Australia, (PMLR, 2017).

49. Dudley, R. M. The speed of mean Glivenko-Cantelli convergence. Ann. Math. Stat. 40, 40–50 (1969).

50. Genevay, A., Chizat, L., Bach, F., Cuturi, M. & Peyré, G. Sample complexity of Sinkhorn divergences. In Proceedings of Machine Learning Research 89, 1574–1583 (eds Chaudhuri, K. & Sugiyama, M.). (PMLR, 2019).

51. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Quantum generative adversarial networks using low-depth circuits. Preprint at http://arxiv.org/abs/1901.08047 (2019).

52. Gibbs, A. L. & Su, F. E. On choosing and bounding probability metrics. Int. Stat. Rev. 79, 419–433 (2002).

53. Dudley, R. M. The speed of mean Glivenko-Cantelli convergence. Ann. Math. Stat. 40, 40–50 (1969).

54. Genevay, A., Chizat, L., Bach, F., Cuturi, M. & Peyré, G. Sample complexity of Sinkhorn divergences. In Proceedings of Machine Learning Research 89, 1574–1583 (eds Chaudhuri, K. & Sugiyama, M.). (PMLR, 2019).

55. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Quantum generative adversarial networks using low-depth circuits. Preprint at http://arxiv.org/abs/1901.08047 (2019).

56. Hangleiter, D., Kliesch, M., Eisert, J. & Gogolin, C. Sample complexity of device-independent certified “quantum supremacy”. Phys. Rev. Lett. 122, 210502 (2019).

57. Goldreich, O., Goldwasser, S. & Ron, D. Property testing and its connection to learning and approximation. J. ACM 45, 653–750 (1998).

58. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B. & Melko, R. Quantum Boltzmann machines. Phys. Rev. X 8, 021050 (2018).

59. Verdon, G., Broughton, M. & Biamonte, J. A quantum algorithm to train neural networks using low-depth circuits. Preprint at http://arxiv.org/abs/1712.05040 (2017).

60. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. In 3rd International Conference on Learning Representations, (ICLR) 2015, (eds Yoshua, B. & Yann, L.) (San Diego, CA, USA, 2015).

61. Coyle, B., Mills, D., Danos, V. & Kashefi, E. The Born supremacy: quantum advantage and training of an Ising Born machine. Preprint at http://arxiv.org/abs/1904.02214 (2019).

62. Coyle, B. IsingBornMachine. https://zenodo.org/record/3779865#.XqvfknVKhrk (2020).

ACKNOWLEDGMENTS
B.C. thanks Andru Gheorghiu for useful discussions and title suggestion. We also thank Jean Feydy, Patric Fulop, Vojtech Havlicek, and Jiasen Yang for clarifying pointers. We thank Atul Mantni for comments on the manuscript. This work was supported by the Engineering and Physical Sciences Research Council (grants EP/L015033/X, EP/N003829/1), EPSRC Centre for Doctoral Training in Pervasive Parallelism at the University of Edinburgh, and School of Informatics and Entapping Machines (grant FA9550-17-1-0055). We also thank Rigetti Computing for the use of their quantum compute resources, and views expressed in this paper are those of the authors and do not necessarily reflect the views or policies of Rigetti Computing.

AUTHOR CONTRIBUTIONS
B.C. devised the theoretical aspects of the work and wrote the code for the numerical results with help from D.M.; D.M. contributed to the learning supremacy definitions. E.K. and V.D. supervised the work. All authors contributed to the manuscript writing.

COMPETING INTERESTS
The authors declare no competing interests.

ADDITIONAL INFORMATION
Supplementary information is available for this paper at https://doi.org/10.1038/s41534-020-00288-9.

Correspondence and requests for materials should be addressed to B.C.

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