

ARTICLE OPEN

A generative modeling approach for benchmarking and training shallow quantum circuits

Marcello Benedetti^{1,2}, Delfina Garcia-Pintos³, Oscar Perdomo^{3,4,5}, Vicente Leyton-Ortega^{3,4}, Yunseong Nam⁶ and Alejandro Perdomo-Ortiz^{1,3,4,7,8}

Hybrid quantum-classical algorithms provide ways to use noisy intermediate-scale quantum computers for practical applications. Expanding the portfolio of such techniques, we propose a quantum circuit learning algorithm that can be used to assist the characterization of quantum devices and to train shallow circuits for generative tasks. The procedure leverages quantum hardware capabilities to its fullest extent by using native gates and their qubit connectivity. We demonstrate that our approach can learn an optimal preparation of the Greenberger-Horne-Zeilinger states, also known as “cat states”. We further demonstrate that our approach can efficiently prepare approximate representations of coherent thermal states, wave functions that encode Boltzmann probabilities in their amplitudes. Finally, complementing proposals to characterize the power or usefulness of near-term quantum devices, such as IBM’s quantum volume, we provide a new hardware-independent metric called the qBAS score. It is based on the performance yield in a specific sampling task on one of the canonical machine learning data sets known as Bars and Stripes. We show how entanglement is a key ingredient in encoding the patterns of this data set; an ideal benchmark for testing hardware starting at four qubits and up. We provide experimental results and evaluation of this metric to probe the trade off between several architectural circuit designs and circuit depths on an ion-trap quantum computer.

npj Quantum Information (2019)5:45; <https://doi.org/10.1038/s41534-019-0157-8>

INTRODUCTION

What is a good metric for the computational power of noisy intermediate-scale quantum¹ (NISQ) devices? Can machine learning (ML) provide ways to benchmark the power and usefulness of NISQ devices? How can we capture the performance scaling of these devices as a function of circuit depth, gate fidelity, and qubit connectivity? In this work, we design a hybrid quantum-classical framework called data-driven quantum circuit learning (DDQCL) and address these questions through simulations and experiments.

Hybrid quantum-classical algorithms, such as the variational quantum eigensolver^{2,3} (VQE) and the quantum approximate optimization algorithm^{4,5} (QAOA), provide ways to use NISQ computers for practical applications. For example, VQE is often used in quantum chemistry when searching the ground state of the electronic Hamiltonian of a molecule.^{6–8} QAOA is used in combinatorial optimization to find approximate solutions of a classical Ising model⁹ and it has been demonstrated on a 19-qubit device for the task of clustering synthetic data.¹⁰ Other successful hybrid approaches based on genetic algorithms were proposed for approximating quantum adders and training quantum autoencoders.^{11–13} In all these examples, there is a clear-cut objective function describing the cost associated with each candidate solution. The task is then to optimize it exploiting both quantum and classical resources.

Although optimization tasks offer a great niche of applications, probabilistic tasks involving sampling have most of the potential to prove quantum advantage in the near-term.^{14–16} For example,

learning probabilistic generative models is in many cases an intractable task, and quantum-assisted algorithms have been proposed for both gate-based^{17,18} and quantum annealing devices.^{19–23} Differently from optimization, the learning of a generative model is not always described by a clear-cut objective function. All we are given as input is a data set, and there are several cost functions that could be used as a guide, each one with their own advantages, disadvantages, and assumptions.

Here we present a hybrid quantum-classical approach for generative modeling on gate-based NISQ devices which heavily relies on sampling. We use the 2^N amplitudes of the wave function obtained from a N -qubit quantum circuit to construct and capture the correlations observed in a data set. As Born’s rule determines the output probabilities, this model belongs to the class of models called Born machines.²⁴ Previous implementations of Born machines^{25–28} often relied on the construction of tensor networks and their efficient manipulation through graphical-processing units. Our work differs in that Born’s rule is naturally implemented by a quantum circuit executed on a NISQ hardware. Following the notation of subsequent work,²⁹ we refer to our generative model as a *quantum circuit Born machine*.

By employing the quantum circuit itself as a model for the data set, we also differentiate from quantum algorithms that target specific probability distributions. For example, in ref.³⁰ the authors developed a hybrid algorithm to approximately sample from a Boltzmann distribution. The samples are used to update a classical generative model, which requires running the hybrid algorithm for

¹Department of Computer Science, University College London, London WC1E 6BT, UK; ²Cambridge Quantum Computing Limited, Cambridge CB2 1UB, UK; ³Qubitera, LLC., Mountain View, CA 94041, USA; ⁴Rigetti Computing, 2919 Seventh Street, Berkeley, CA 94710-2704, USA; ⁵Department of Mathematics, Central Connecticut State University, New Britain, CT 06050, USA; ⁶IonQ, Inc., College Park, MD 20740, USA; ⁷Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, Mountain View, CA 94035, USA and ⁸Zapata Computing Inc., 439 University Avenue, Office 535, Toronto, ON M5G 1Y8, Canada
Correspondence: Alejandro Perdomo-Ortiz (alejandror@zapatacomputing.com)

Received: 21 November 2018 Accepted: 1 May 2019

Published online: 27 May 2019

each training iteration. In contrast, our work does not assume a Boltzmann distribution and therefore does not require a specific sampling beyond Born's rule.

Our work is also in contrast with other quantum-assisted ML work (see e.g., refs 31–34) requiring fault-tolerant quantum computers, which are not expected to be readily available in the near-term.³⁵ Instead, our circuits are carefully designed to exploit the full power of the underlying NISQ hardware without the need for a compilation step.

On the benchmarking of NISQ devices, quantum volume^{9,36} has been proposed as an architecture-neutral metric. It is based on the task of approximating a specific class of random circuits and estimating the associated effective error rate. This is very general and it is indeed useful for estimating the computational power of a quantum computer. In this paper, we propose the *qBAS score*, a complementary metric designed for benchmarking hybrid quantum-classical systems. The score is based on the generative modeling performance on a canonical synthetic data set which is easy to generate, visualize, and validate for sizes up to hundreds of qubits. Yet, implementing a shallow circuit that can uniformly sample such data is hard; we will show that some candidate solutions require large amount of entanglement. Hence, any miscalibration or environmental noise will affect this single performance number, enabling comparison between different devices or across different generations of the same device. Moreover, the score depends on the classical resources, hyper-parameters, and various design choices, making it a good choice for the assessment of the hybrid system as a whole.

Our design choices are based on the setup of existing ion trap architectures. We choose the ion trap because of its full connectivity among qubits³⁷ which allows us to study several circuit layouts on the same device. Our experiments are carried out on an ion trap quantum computer hosted at the University of Maryland.³⁸

RESULTS

The learning pipeline

In this Section, we present a hybrid quantum-classical algorithm for the unsupervised machine learning task of approximating an unknown probability distribution from data. This task is also known as generative modeling. First, we describe the data and the model, then we present a training method.

The data set $\mathcal{D} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(D)})$ is a collection of D independent and identically distributed random vectors. The underlying probabilities are unknown and the target is to create a model for such distribution. For simplicity, we restrict our attention to N -dimensional binary vectors $\mathbf{x}^{(d)} \in \{-1, +1\}^N$, e.g., black and white images. This gives us an intuitive one-to-one mapping between observation vectors and the computational basis of an N -qubit quantum system, that is $\mathbf{x} \leftrightarrow |\mathbf{x}\rangle = |x_1 x_2 \dots x_N\rangle$. Note that standard binary encodings can be used to implement integer, categorical, and approximate continuous variables.

Provided with the data set \mathcal{D} , our goal is now to obtain a good approximation to the target probability distribution $P_{\mathcal{D}}$. A quantum circuit model with fixed depth and gate layout, parametrized by a vector θ , prepares a wave function $|\psi(\theta)\rangle$ from which probabilities are obtained according to Born's rule $P_{\theta}(\mathbf{x}) = |\langle \mathbf{x} | \psi(\theta) \rangle|^2$. Following a standard approach from generative machine learning,³⁹ we can minimize the Kullback–Leibler (KL) divergence⁴⁰ $D_{\text{KL}}[P_{\mathcal{D}}|P_{\theta}]$ from the circuit probability distribution in the computational basis P_{θ} to the target probability distribution $P_{\mathcal{D}}$. Minimization of this quantity is directly related to the minimization of a well known cost function: the negative log-likelihood $\mathcal{C}(\theta) = -\frac{1}{D} \sum_{d=1}^D \ln(P_{\theta}(\mathbf{x}^{(d)}))$. However, there is a caveat; as probabilities are estimated from frequencies of a finite number of measurements, low-amplitude states could lead to

incorrect assignments. For example, an estimate $P_{\theta}(\mathbf{x}^{(d)}) = 0$ for some $\mathbf{x}^{(d)}$ in the data set would lead to infinite cost. To avoid singularities in the cost function, we use a simple variant

$$C_{\text{nil}}(\theta) = -\frac{1}{D} \sum_{d=1}^D \ln(\max(\varepsilon, P_{\theta}(\mathbf{x}^{(d)}))), \quad (1)$$

where $\varepsilon > 0$ is a small number to be chosen. Note that the number of measurements needed to obtain an unbiased estimate of the relevant probabilities may not scale favorably with the number of qubits N . In the Supplementary Material we suggest alternative cost functions such as the moment matching error and the earth mover's distance.

After estimating the cost, we update the parameter vector θ to further minimize the cost. This can in principle be done by any suitable classical optimizer. We used a gradient-free algorithm called particle swarm optimization (PSO)^{41,42} as previously done in the context of quantum chemistry.⁷ The algorithm iterates for a fixed number of steps, or until a local minimum is reached and the cost does not decrease.

We chose the layout of the model circuit to be of the following form. Let us consider a general circuit parametrized by single qubit rotations $\{\theta_i^{(l,k)}\}$ and two-qubit entangling rotations $\{\theta_{ij}^{(l)}\}$. The subscripts denote qubits involved in the operation, l denotes the layer number and $k \in \{1, 2, 3\}$ denotes the rotation identifier. The latter is needed as we decompose an arbitrary single qubit rotation into three simpler rotations (see Section “Methods” for further details about some exceptions and potential simplifications depending on the native gates available in each specific hardware). Inspired by the gates readily available in ion trap quantum computers, we use alternating layers of arbitrary single qubit gates (odd layers) and Mølmer-Sørensen XX entangling gates^{43–46} (even layers) as our model. All parameters are initialized at random.

It is important to note that in our model the number of parameters is fixed and is independent of the size D of the data set. This means we can hope to obtain a good approximation to the target distribution only if the model is flexible enough to capture its complexity. Increasing the number of layers or changing the topology of the entangling layer alter this flexibility, potentially improving the quality of the approximation. However, we anticipate that such flexible models are more challenging to optimize because of their larger number of parameters. Principled ways to choose the circuit layout and to regularize its parameters could significantly help in such a case.

As summarized in Fig. 1, the learning algorithm iteratively adjusts all the parameters to minimize the value of the cost function. At any iteration the user-defined cost is approximated using both samples from the data set and measurements from the quantum hardware, hence the name data-driven quantum circuit learning (DDQCL).

The qBAS score

Bars and stripes (BAS)⁴⁷ is a synthetic data set of images that has been widely used to study generative models for unsupervised machine learning. For $n \times m$ pixels, there are $N_{\text{BAS}(n,m)} = 2^n + 2^m - 2$ images belonging to BAS, and they can be efficiently produced and visualized. The probability distribution is $1/N_{\text{BAS}(n,m)}$ for each pattern belonging to $\text{BAS}(n, m)$, and zero for any other pattern. Figure 2 on the top left panel shows patterns belonging to $\text{BAS}(2, 2)$, while the top central panel shows the remaining patterns.

We use DDQCL to learn a circuit that encodes all the BAS patterns in the wave function of a quantum state. This also allows us to design the $\text{qBAS}(n, m)$ score: a task-specific figure of merit to assess the performance of shallow quantum circuits. In a single number, it captures the model capacity of the circuit layout and intrinsic hardware strengths and limitations in solving a complex

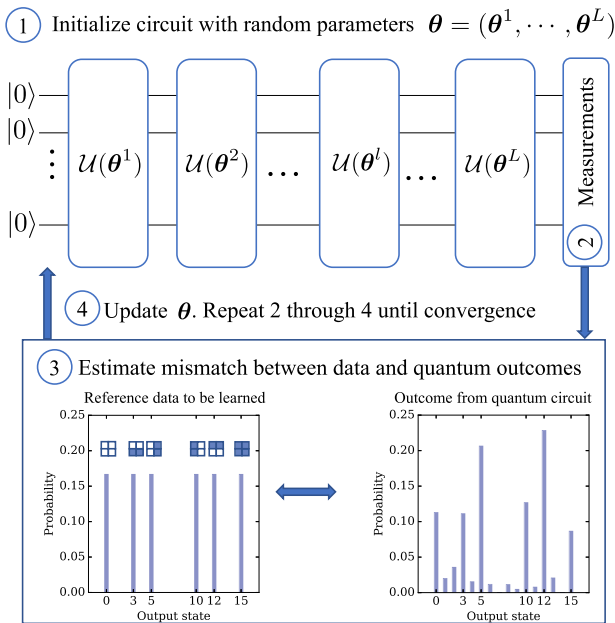


Fig. 1 General framework for data-driven quantum circuit learning (DDQCL). Data vectors are interpreted as representative samples from an unknown probability distribution and the task is to model such distribution. The 2^n amplitudes of the wave function resulting from an N -qubit quantum circuit are used to capture the correlations observed in the data. Training of the quantum circuit is achieved by successive updates of the parameters θ , corresponding to specifications of single qubit operations and entangling gates. In this work, we use arbitrary single qubit rotations for the odd layers, and Mølmer-Sørensen XX gates for the even layers. At each iteration, measurements from the quantum circuit are collected and contrasted with the data through evaluation of a cost function which tracks the learning progress

sampling task that requires a fair amount of entanglement. It takes into account the model circuit depth, gate fidelities, and any other architectural design aspects, such as the quantum hardware’s qubit-to-qubit connectivity and native set of single and two-qubit gates. It also takes classical resources such as the choice of cost function, optimizer, and hyper-parameters, into account. Therefore, it can be used to benchmark the performance of the components of the hybrid quantum-classical system.

The $\text{qBAS}(n, m)$ score is an instantiation of the F_1 score widely used in the context of information retrieval. The F_1 score is defined as the harmonic mean of the precision p and the recall r , i.e., $F_1 = 2pr/(p + r)$. The precision p indicates the ability to retrieve states which belong to the data set of interest (The meaning and usage of precision in the field of information retrieval differs from the definition of precision within other branches of science and statistics). In our context this is the number of measurements that belong to the $\text{BAS}(n, m)$ data set, divided by the total number of measurements N_{reads} performed. The recall r is the capacity of the model to retrieve the whole spectrum of patterns belonging to the desired data set. In our case, if we denote the number of unique patterns that were measured as $d(N_{\text{reads}})$, then $r = d(N_{\text{reads}})/N_{\text{BAS}(n,m)}$. To score high ($F_1 \approx 1.0$), both high precision ($p \approx 1.0$) and high recall ($r \approx 1.0$) are required.

The F_1 score is a useful measure for the quality of information retrieval and classification algorithms, but for our purposes it has a caveat: the dependence of r on the total number of measurements. As an example, consider a model that generates only BAS patterns, i.e., its precision is 1.0, but with highly heterogeneous distribution. If some of the BAS patterns have infinitesimally small probability, we can still push the recall to 1.0 by taking a large number of measurements $N_{\text{reads}} \rightarrow \infty$. This is not desirable since

our purpose is to evaluate circuits on the task of uniformly sampling all the patterns from $\text{BAS}(n, m)$. Therefore, to define a unique score it is important to fix N_{reads} to a reasonable value such that $r \approx 1.0$ under the assumption of the model distribution being equal to the target distribution $P_{\text{BAS}(n,m)}$, but not so large as to make the score insensitive to deviations from the target distribution. Assuming a perfect model distribution $P_{\text{BAS}(n,m)} = 1/N_{\text{BAS}(n,m)}$, the expected number of measurements to obtain a value of $r = 1.0$ can be estimated using the famous *coupon collector’s problem*. For our purposes here, we set N_{reads} to be equal to the expected number of samples that need to be drawn to collect all the $N_{\text{BAS}(n,m)}$ patterns (“coupons”). That is, $N_{\text{reads}} = N_{\text{BAS}(n,m)} H_{N_{\text{BAS}(n,m)}}$ where H_k is the k -th harmonic number. Computed values of N_{reads} are provided in Table 1 for different values of n and m up to 100 qubits. As shown in the table, the number of readouts required to determine $\text{qBAS}(n, m)$ are within experimental capabilities of current NISQ devices. (It is an interesting problem to optimize N_{reads} such that it maximizes the sensitivity of the score towards differentiating two probability distributions. This problem is left for future work).

For statistical robustness, we recommend as a good practice to perform R repetitions of the N_{reads} measurements leading to R independent estimates of the recall (each denoted as r_i). For estimating the precision p , all the samples collected should be used to robustly estimate this quantity. Using this value of p one can compute R independent values of the $\text{qBAS}(n, m)$ score from each of the r_i . These are subsequently bootstrapped to obtain a more robust average for the final reported value of $\text{qBAS}(n, m)$ (see details in Section “Methods”).

We note that a more general performance indicator than $\text{qBAS}(n, m)$ score is indeed the KL divergence, $D_{\text{KL}}[P_{\text{BAS}(n,m)}|P_{\theta}]$. However, this would not be robust in terms of scalability; as $n \times m$ becomes large, it is expected that the KL divergence is frequently undefined (One may argue that the same scalability issue applies to the cost function in Eq. (1) used for learning. However, we can choose alternative scalable cost functions, as we show in the Supplementary Material. A comprehensive study is left for future work). This is true when measurements yield distributions such that $P_{\theta}(\mathbf{x}^{(d)}) = 0$ for any of the $\mathbf{x}^{(d)}$ in $\text{BAS}(n, m)$. In all these cases, the qBAS score can still be computed and the number of measurements N_{reads} necessary for obtaining a robust estimate continues to remain relatively small to be practical for intermediate size $n \times m$.

Experiments

We investigated three different examples, namely, GHZ state preparation, coherent thermal state preparation, and $\text{BAS}(2, 2)$. We implemented each example of DDQCL using both numerical simulations and experiments. We explored the parameter space of DDQCL by varying the qubit-to-qubit connectivity topology (see top-right panel of Fig. 2) and the number of layers. We evaluated the performance of each instance by using the KL divergence from the circuit probability distribution in the computational basis to the target probability distribution. While explicitly computing the KL divergence is generally intractable, due to its demanding resource requirement as the size of instances to consider increases, we were able to compute this quantity explicitly for all cases considered in this paper.

GHZ state preparation

To test the capabilities of DDQCL, we started with the preparation of GHZ states, also known as “cat states”.⁴⁸ Besides their importance in quantum information, the choice is motivated by their simple description and by the availability of many studies about their preparation (see e.g., refs 49–51). From the DDQCL perspective, we explored whether it is possible to learn any of the known recipes for GHZ state preparation starting only from classical data. More

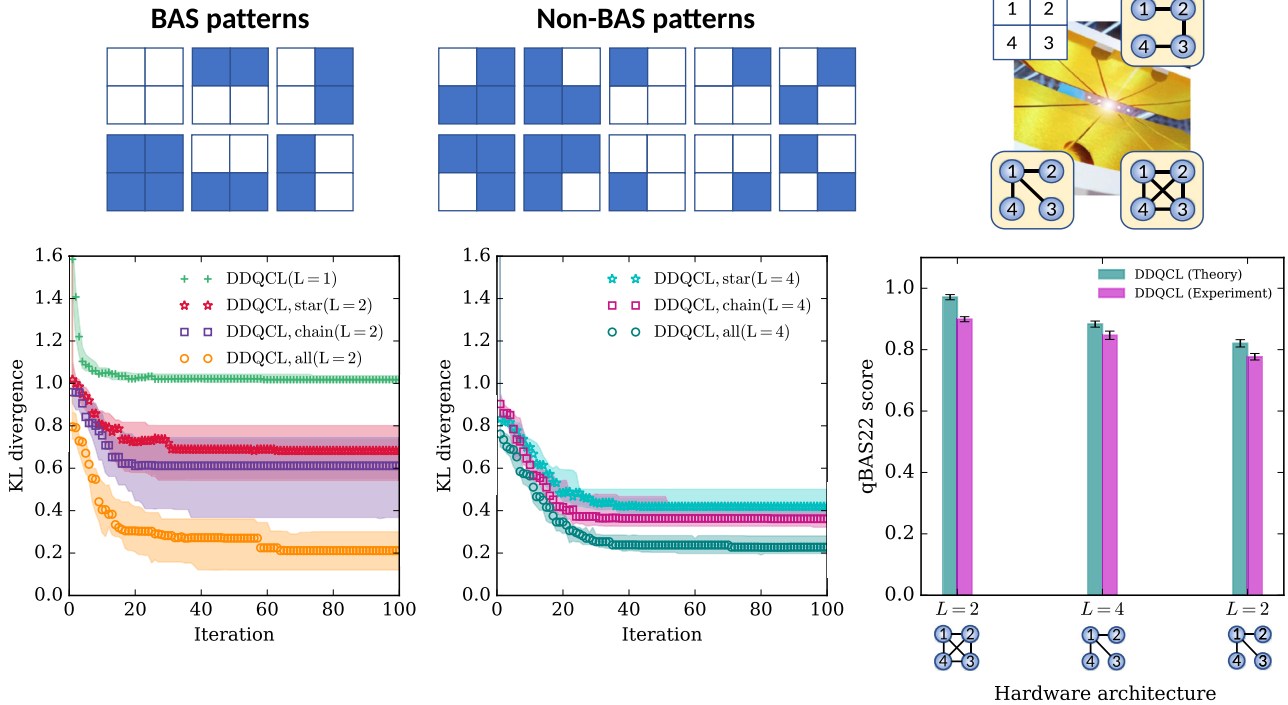


Fig. 2 DDQCL on the BAS data set. The top left panel shows patterns that belong to BAS(2, 2) our quantum circuit is to generate. The top central panel shows undesired patterns. On the top right panel, we show a possible mapping of the 4 pixels to $N = 4$ qubits, and we show some of the qubit-to-qubit connectivity topologies that can be set up in entangling layer and natively implemented by the ion trap quantum computer (e.g. *chain*, *star*, and *all*). The bottom left panel shows the results of DDQCL simulations of shallow circuits with different topologies. We show the bootstrapped median and 90% confidence interval over the distribution of medians of the KL divergence as learning progresses for 100 iterations. The mean-field-like circuit $L = 1$ (green crosses) severely underperforms. A significant improvement is obtained with $L = 2$, where most of the parameters for XX gates have been learned to their maximum entangling value. These observations indicate that entanglement is a key resource for learning the BAS data set. Note that for $L = 2$ the choice of topology becomes a key factor for improving the performance. The chain topology (purple squares) performs slightly better than the star topology (red stars) even though they have the same number of parameters. The all-to-all topology (orange circles) significantly outperform all the others as it has more expressive power. The bottom central image extends the previous analysis to deeper circuits with $L = 4$ and approximately twice the number of parameters. All the topologies achieve a lower median KL divergence and the confidence intervals shrink. The bottom right panel shows the bootstrapped mean qBAS(2, 2) score and 95% confidence interval for simulations (green bars) and experiments on the ion trap quantum computer hosted at University of Maryland (pink bars)

Table 1. Example of experimental requirements for near-term quantum computers with up to 100 qubits			
(n, m)	N_{qubits}	$N_{\text{BAS}(n,m)}$	N_{reads}
(2, 2)	4	6	15
(2, 3)	6	10	30
(3, 3)	9	14	46
(4, 4)	16	30	120
(7, 7)	49	254	1554
(8, 8)	64	510	3475
(10, 10)	100	2046	16780

Note: As described in the main text, N_{reads} is the number of readouts required for every estimation of the qBAS score

specifically, the input data consists of samples from a distribution corresponding to the two desired computational basis states; $P_0 = 0.5$ for the state $|0\dots 0\rangle$ and $P_1 = 0.5$ for the state $|1\dots 1\rangle$. Using a layer of single qubit rotations followed by an entangling layer with the all-to-all topology, DDQCL yielded many degenerate preparations of GHZ-like states differing only by a relative phase.

In particular, we first ran particle swarm optimization on 25 random initializations for 3, 4, 5, and 6-qubit instances. Then, a

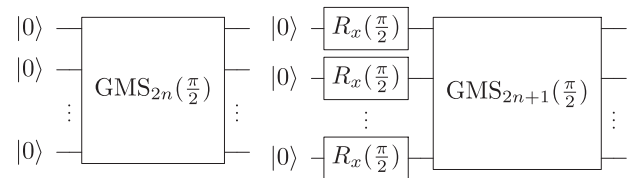


Fig. 3 GHZ-like state preparation assisted by DDQCL. Left (right) panel shows a recipe obtained by a human expert assisted by DDQCL for the even (odd) cat state preparation. R_x stands for the single qubit rotation about the x axis. GMS stands for a global Mølmer-Sørensen gate⁴⁶ acting on all the $N = 2n$ ($N = 2n + 1$) qubits and is equivalent to the application of local XX gates to all $N(N - 1)/2$ pairs of qubits. All parameters attained values very close to $\frac{\pi}{2}$. The human expert rounded the parameters to some precision and found these patterns

human expert inspected the best set of parameters learned for each size and after rounding the parameters to some precision, spotted a clear pattern. Instances of 3 and 5 qubits yielded a recipe, while instances of 4 and 6 qubit yielded another recipe. The recipes obtained are summarized in Fig. 3 and were verified for larger number of qubits, both odd and even. Indeed, DDQCL successfully reproduced the recipes previously used on ion trap quantum computers,⁵² and, to the best of our knowledge, they correspond to the most compact and efficient protocols for GHZ

state preparation using XX gates (see Fig. 3). Another commonly used approach consists of cascading entangling gates, with alternations of single qubit rotations.⁵¹ DDQCL produced approximate recipes of this kind in some of the test cases for 3 and 4 qubits when using a single entangling layer with chain topology.

It is interesting to note that in DDQCL all the parameters are learned independently and not constrained to be the same. As shown in Fig. 3, the learning process unveiled that these converge to the same value. This is not necessarily the case for the other data sets considered below. We also note that the simulations assumed noiseless hardware, making the analysis of parameters easier for the human expert. It would be much more difficult to analyze parameters found with noisy hardware, as DDQCL can learn to compensate certain types of noise, e.g., systematic parameter offsets, in non-trivial ways. The upside is that learning can be successful even in the presence of such systematic errors.

Finally, it is reassuring that DDQCL obtained circuits for cat state preparation starting from samples of a classical distribution. While this target distribution could have been modeled by a zero-temperature ferromagnet, there apparently was no other way for our circuit to reproduce such a distribution, if not by preparing a GHZ-like state. One can obtain more general solutions by allowing DDQCL to prepare mixed states. For example, consider a circuit acting on both the main qubit register and an additional ancilla register. By tracing out the ancilla register, e.g., ignoring it during measurement, the main register can implement a mixed state and can be trained to simulate a zero-temperature ferromagnet. Another example which does not resort to an ancilla register is to use decoherence as a mechanism to prepare mixed states that explain the data.

Coherent thermal states

Thermal states play an important role in statistical physics, quantum information, and machine learning. Using DDQCL, we trained quantum circuits with different number of layers $L \in \{1, 2, 3\}$ and using all-to-all topology, to approximate a target Boltzmann distribution. In particular, we considered data sets sampled from the Boltzmann distribution of 25 synthetic instances with $N = 5$ qubits. By decreasing the temperature T of the target distribution, we can increase the difficulty of the learning task. Figure 4 shows the bootstrapped median and 90% confidence interval over the distribution of medians of the KL divergence

during learning. Deeper circuits such as $L = 3$ (purple pentagons) consistently outperformed shallower circuits such as $L = 2$ (red circles) and $L = 1$ (yellow triangles). This became more evident as we went from easy learning tasks (Fig. 4a) to hard learning tasks (Fig. 4c). Results for instances of $N = 6$ qubits are shown in Fig. S2 in the Supplementary Material.

To assess how well DDQCL performs on the generative task, we compared DDQCL to the inverse Bethe approximation⁵³ (see also Eqs. (3.21) and (3.22) in ref.⁵⁴), a classical closed-form approach widely used in statistical physics to solve the inverse Ising problem. As shown in Fig. 4, the inverse Bethe approximation (green bar) performed extremely well in the easy task (a), matched the $L = 3$ quantum circuit in the intermediate task (b), and underperformed on the difficult task (c). The latter observation comes from the fact that the median performance of the inverse Bethe approximation has very large confidence intervals. We emphasize that this is not a form of quantum supremacy as the two methods are fundamentally different. DDQCL prepares a quantum state without the assumption of an underlying Boltzmann distribution, while the inverse Bethe approximation infers the parameters with such assumption. Furthermore, the error in the inverse Bethe approximation is expected to go to zero with system size, and only above the reference temperature T_c (see Section “Methods” for details). Thus, it is not surprising that we obtained bad performance in Fig. 4c with the inverse Bethe approximation. Other classical methods based on machine learning and Markov Chain Monte Carlo, such as the Boltzmann machine,⁵⁵ could achieve higher accuracy by requiring more computational resources than the inverse Bethe approximation used here. A thorough comparison is beyond the scope of this work.

BAS(2, 2)

For the purposes of benchmarking and measuring the power of NISQ devices with DDQCL, it is insufficient to have an easy-to-generate target data set; we also require the data set to represent a useful quantum state in quantum computing, while simultaneously proving to be sufficiently challenging for the quantum computer to generate. Because of the importance of entanglement in quantum information processing, we considered the entanglement entropy averaged over all two-qubit subsets⁵⁶ as a proxy measure of a specific quantum state’s usefulness for

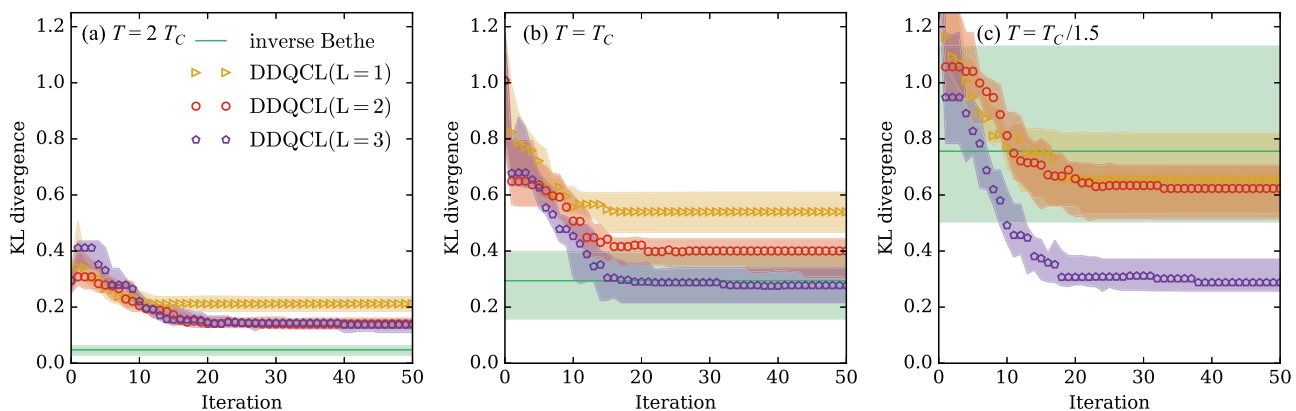


Fig. 4 DDQCL preparation of coherent thermal states. We generated 25 random instances of size $N = 5$ and varied the difficulty of the learning task by decreasing the temperature in $T \in \{2T_c, T_c, T_c/1.5\}$ where T_c is the reference temperature (see Section “Methods” for details). The model is a quantum circuit with five qubits and an all-to-all qubit connectivity for the entangling layer. We show the bootstrapped median and 90% confidence interval over the distribution of medians of the KL divergence of DDQCL as learning progresses for 50 iterations. **a** When $T > T_c$ the learning task is easy and shallow quantum circuits such as $L = 1$ (yellow triangles) and $L = 2$ (red circles) perform very well. **b** When $T \approx T_c$ a gap in performance between circuits of different depth becomes evident. **c** When $T < T_c$ the learning task becomes hard and deeper circuits perform much better than shallow ones. We also report results for the inverse Bethe approximation, which does not actually prepare a state, but produces a classical model in closed-form. The classical model so obtained (green band) is excellent for the easy task in **a**, matches the best quantum model in **b**, and underperforms for the hard task in **c**

benchmarking purposes. We start by noting that the four-qubit cat state, whose rich entangled nature makes it ideal for studying decoherence and decay of quantum information,^{49,51} has entanglement entropy $S_{GHZ} = 1$. Now consider states that encode $BAS(2, 2)$ in the computational basis. The minimum value of entanglement entropy that any such state can have is $S_{BAS(2,2)} = 1.25163$. Furthermore, the maximum value that a quantum representation of $BAS(2, 2)$ can reach is $S_{BAS(2,2)} = 1.79248$, which happens to be the maximum entanglement entropy known for any four-qubit state⁵⁶ (see also Fig. S5 and the corresponding Section in the Supplementary Material).

We also note that one of the quantum representations of $BAS(2, 2)$ found by DDQCL reached a remarkable value of $S_{BAS(2,2)} = 1.69989$ (see Fig. S3 in the Supplementary Material). This shows the power of our framework, in that DDQCL is capable of handling useful quantum states that are rich in entanglement. This is an important observation, since we know, based on our empirical results, that (i) single layer circuits with no entangling gates severely underperform in producing the output state probability distribution that is close to the target data set, and (ii) when inspecting the parameters learned for circuits with all-to-all topology with $L = 2$ layers, we found that most of the XX gates reached their maximum entangling setting.

We now discuss results for the $qBAS(2, 2)$ score, which we computed experimentally and theoretically in order to compare the entangling topologies sketched in the top right panel of Fig. 2 and for different number of layers. The process consists of two steps; first, DDQCL is used to encode $BAS(2, 2)$ in the wave function of the quantum state. Second, the best circuits, i.e., those with lowest cost, are compared using the $qBAS(2, 2)$ score.

The bottom left and bottom central panels in Fig. 2 show the bootstrapped median of the KL divergence and 90% confidence interval over 25 random initializations of DDQCL in silico. The all-to-all topology (orange circles) always outperforms sparse topologies (red stars and purple squares). However, deeper circuits do not always provide significant improvements, as it is the case for all-to-all $L = 4$ (dark green circles) versus all-to-all $L = 2$ (orange circles). A possible explanation is that, when going from two to four layers, we approximately double the number of parameters, and particle swarm optimization struggles to find enhanced local optima. Another plausible explanation is that for this small data set, the all-to-all circuits with $L = 2$ are already close to optimal performance (we show supporting evidence in Fig. S3 in the Supplementary Material).

As the best performing circuits to compare using the $qBAS$ score, we chose all-to-all $L = 2$ and star $L \in \{2, 4\}$ circuits. While they represent very different approximate solutions to the same problem, they may be compared with the help of $qBAS(2, 2)$ score. For each setting, we computed 25 scores from batches of size $N_{reads} = 15$ samples, as described in Section “Results”. The bottom right panel in Fig. 2 shows the bootstrapped mean $qBAS(2, 2)$ score and 95% confidence interval for simulations (green bars) and experiments on the ion trap quantum computer hosted at University of Maryland (pink bars). The score is sensitive to the depth of the circuit as shown by the performance improvement of $L = 4$ compared with $L = 2$ in the star topology. Note that the theoretical improvement for using $L = 4$ is larger than that observed experimentally in the ion trap. This is because the quantum computer accumulated errors while executing the deeper circuit. The score is also sensitive to the choice of topology as shown by the drop in performance of star compared with all-to-all when the same number of layers $L = 2$ is used.

Although we compared circuits implemented on the same ion trap hardware, the score may be used to compare different device generations or even completely different architectures (e.g., superconductor-based versus atomic-based). Similarly, one may use the score to compare classical resources of the hybrid system (e.g., different optimizers).

DISCUSSION

Data is an essential ingredient of any machine learning task. In this work, we presented a data-driven quantum circuit learning algorithm (DDQCL) as a framework that can assist in the characterization of NISQ devices and to implement simple generative models. The success of this approach is evidenced by the results on three different data sets.

To summarize, first, we learned a GHZ state preparation recipe for an ion trap quantum computer. Minimal intervention by a human expert allowed to generalize the recipe to any number of qubits. This is not an example of compilation, but rather an illustration of how simple classical probability distributions can guide the synthesis of interesting non-trivial quantum states. Depending on the level or type of noise in the system, the same algorithm could lead to a different circuit fulfilling the same probability distribution as that of the data. The message here is that machine learning can teach us that “there is more than one way to skin a cat (state)”.

Second, we trained circuits to prepare approximations of thermal states. This illustrates the power of Born machines²⁴ to approximate Boltzmann machines⁵⁵ when the data require thermal-like features.

Finally, tapping into the real power of near-term quantum devices and approximate algorithms implementable on them, we designed a task-specific performance estimator based on a canonical data set. The bars and stripes data is easy to generate, visualize and verify classically, while modeling it still requires significant quantum resources in the form of entanglement. Errors in the device will affect this single performance measure, the $qBAS$ score, which can be used to compare different device generations, or completely different architectures. The $qBAS$ score can also be used to benchmark the typical performance of optimizers used in hybrid quantum-classical systems. Selecting the method and optimizing the hyper-parameters can be a daunting task and is a key challenge towards a successful implementation as the number of qubits increases. Therefore, having this unique metric for benchmarking could help reduce the complexity of this fine-tuning stage. The score can be computed in any of the NISQ architectures available to date.

DDQCL is a modular framework and its performance will ultimately depend on the choices made for such modules. In this article we explored the impact of circuit layout and cost function, while subsequent work has analyzed other modules and suggested extensions to the algorithm. In ref.²⁹ the authors trained Born machines using a differentiable cost function and exploiting gradient calculations proposed in ref.⁵⁷. In refs.^{58,59} the authors compared several optimizers, and in ref.⁶⁰ the authors focused on the impact of hardware noise. The expressive power of shallow circuits was investigated in ref.⁶¹ and it was shown to outweigh that of some classes of artificial neural networks. Finally, recent work has shown successful implementation of DDQCL on the IBM Q 20 Tokyo processor,⁶⁰ on a five-qubit ion-trap hosted at University of Maryland,⁵⁸ and on the Rigetti 16Q Aspen-1 processor.⁵⁹

It is left to future work to demonstrate more realistic machine learning by allowing more flexible models and employing regularization. At a finite and fixed low circuit depth, the power of the generative model can be enhanced by including ancilla qubits, in analogy to the role of hidden units in probabilistic graphical models. Regularization can be included in the cost function. In this paper, we used the negative log-likelihood which quickly becomes expensive to estimate as the size of the system increases. We reported preliminary results on alternative cost-functions that overcome the caveat and still produce satisfactory results. Layer-wise pre-training of the quantum circuit inspired by deep learning³⁹ could initialize parameters to near-optimal locations in the cost landscape. Finally, DDQCL could be generalized to learn quantum distributions or states, assuming

experimental data coming from quantum experiments, e.g., quantum measurements beyond the computational basis. We think these are the most promising directions to be explored in future work.

Our approach has the bidirectional capability of using NISQ devices for machine learning, and machine learning for the characterization of NISQ devices. We hope the ideas presented here contribute to the development of further concrete metrics to help guide the architectural hardware design, while tapping into the computational power of NISQ devices.

METHODS

Simulation of quantum circuits in silico

We simulated quantum circuits using the `Qutip2`⁶² Python library and implemented the constraints dictated by the ion trap experimental setting. In the current experimental setup, we can perform arbitrary single qubit rotations and Mølmer-Sørensen XX entangling gates involving any two qubits. We used only these gates hence avoiding the need of further compilation. For the simulations in silico, we also assume perfect gate fidelities and error-free measurements.

In the ion trap setting, the implementation of single qubit rotations R_z is very convenient. Therefore, we perform arbitrary single qubit operations relying on the decomposition $U_i^{(l)} = R_z(\theta_i^{(l,3)})R_x(\theta_i^{(l,2)})R_z(\theta_i^{(l,1)})$ where l is the layer number, i is the qubit index, and $\theta_i^{(l,k)} \in [-\pi, +\pi]$ are Euler angles. The rotations are then expressed as exponentials of Pauli operators $R_z(\theta_i^{(l,\cdot)}) = \exp\left(-\frac{i}{2}\theta_i^{(l,\cdot)}\sigma_z^i\right)$ and $R_x(\theta_i^{(l,\cdot)}) = \exp\left(-\frac{i}{2}\theta_i^{(l,\cdot)}\sigma_x^i\right)$.

Because we execute circuits always starting from the $|0\dots 0\rangle$ state, the first set of R_z rotations would have no effect and, therefore, is not needed. When an odd number of layers is used, a similar exception occurs in the last layer. There, the last set of R_z rotations would only add a phase that becomes irrelevant when taking the amplitude squared required for the Born machine. In other words, we can slightly reduce the number of parameter without changing the expressive power of the circuit. Every other layer of arbitrary single qubit operations would in general require $3N$ parameters, where N is the number of qubits. By using an alternative decomposition, namely $U = R_xR_zR_x$, we could apply commutation rules with XX gates and obtain a reduction to $2N$ parameters in all odd layers. We decided not to do the former step for two reasons. First, there is no effective reduction of the number of parameters for experiments up to $L = 5$ layers considered here. Second, in the ion trap quantum computer used here³⁸ it is experimentally convenient to use a larger number of R_z rather than R_x rotations.

For the case of the entangling gates, we use the notation $U_{ij}^{(l)} = XX(\theta_{ij}^{(l)})$ which in exponential form reads as $XX(\theta_{ij}^{(l)}) = \exp\left(-\frac{i}{2}\theta_{ij}^{(l)}\sigma_x^i\sigma_x^j\right)$. Recalling that states that differ by a global phase are indistinguishable, a direct computation shows that the tunable parameters can be taken as $\theta_{ij}^{(l)} \in [-\pi, +\pi]$. Also, there is no need to set up an order for these gates within an entangling layer as they commute with one another.

The total number of parameters per entangling layer depends on the chosen topology: *all* is a fully-connected graph and has $N(N-1)/2$ parameters, *chain* is a one-dimensional nearest neighbor graph with $N-1$ parameters, and *star* is a star-shaped graph with $N-1$ parameters. The top right panel of Fig. 2 shows a graphical representation of these topologies for the case of $N=4$ qubits.

When executing DDQCL, we always estimated the required quantities from 1000 measurements in the computational basis.

Gradient-free optimization

Once the number of layers and topology of entangling gates is fixed, the quantum circuits described above provide a template; by adjusting the parameters we can implement a small subset of the unitaries that are in principle allowed in the Hilbert space. The variational approach aims at finding the best parameters by minimizing a cost function. For all our tests, we choose to minimize a clipped version of the negative log-likelihood.

We use a global-best particle swarm optimization algorithm⁴² implemented in the `PySwarms`⁶³ Python library. A 'particle' corresponds to a candidate solution circuit; the position of a particle is a point θ in parameter space, the velocity is a vector determining how to update the position in parameter space. Position and velocity of all the particles are

initialized at random and updated at each iteration following the schema shown in Fig. 1. There are three hyper-parameters controlling the swarm dynamics: a cognition coefficient c_1 , a social coefficient c_2 and an inertia coefficient w . After testing a grid of values, we chose to use a constant value of 0.5 for all three hyper-parameters, which we found to work well for our purpose. To avoid large jumps in parameter space, we further restrict position updates in each dimension to a maximum magnitude of π .

Finally, we set the number of particles to twice the number of parameters of the circuit. This is a conservative value compared with previous work,⁸ also because of the large number of parameters in the circuits explored here.

Data sets details

We worked with three synthetic data sets: zero-temperature ferromagnet, random thermal, and bars and stripes (BAS). In all our numerical experiments we use 1000 data points sampled exactly from these distributions.

GHZ state preparation

The zero-temperature ferromagnet distribution is equivalent to assigning 1/2 probability to both $|0\dots 0\rangle$ and $|1\dots 1\rangle$ states of the computational basis. This distribution can be easily prepared as a mixed state, but our study uses pure states prepared by the circuit. The only way to reproduce the zero-temperature ferromagnet distribution in our setting is to implement a unitary transformation that prepares a GHZ-like state.

Thermal states

A thermal data set in N dimensions is generated by exact sampling realizations of $\mathbf{x} \in \{-1, +1\}^N$ from the distribution $P(\mathbf{x}) = Z^{-1} \exp\left(\left(\sum_{ij} J_{ij} x_i x_j + \sum_i h_i x_i\right) T^{-1}\right)$ where Z is the normalization constant, J_{ij} and h_i are random coefficients sampled from a normal distribution with zero mean and \sqrt{N} standard deviation, and T is the temperature. In the large system-size limit, a phase transition is expected at $T_c \approx 1$. Although this is not true for the small-sized systems considered here, we take this value as a reference temperature. In our study, we vary $T \in \{2T_c, T_c, T_c/1.5\}$ in order to generate increasingly complex instances.

Bars and stripes

BAS⁴⁷ is a canonical machine learning data set for testing generative models. It consists of $n \times m$ pixel pictures generated by setting each row (or column) to either black (-1) or white ($+1$), at random. In generative modeling, a handful of patterns are input to the algorithm and the target is to train a model to capture correlations in the data. Assuming a successful training, the model can reconstruct and generate previously unseen patterns from partial or corrupted data. On the other hand, if we provide the algorithm with *all* the patterns we are interested in and aim to a model that generates only those, this would amount to an associative memory. Although both tasks can be done with our DDQCL pipeline, for the qBAS(n, m) score we focus on the latter task.

We now determine the number of patterns and provide an easy identification of the bitstring belonging to the BAS(n, m) class. For the total count of the number of patterns, we first count the number of single stripes, double stripes, etc. that can fit into the n rows. This number is the sum of binomial coefficients $\sum_{k=0}^n \binom{n}{k} = 2^n$. The same expression holds for the number of patterns with bars that can be placed in the m columns, that is 2^m . Note that empty (all-white) and full (all-black) patterns are counted in both the bars and the stripes. Therefore, we obtain the total count for the BAS patterns by subtracting the two extra patterns from this double count:

$$N_{\text{BAS}(n,m)} = 2^n + 2^m - 2. \quad (2)$$

In the main text, we use the BAS data set to design a task-specific performance indicator for hybrid quantum-classical systems. Table 1 shows the requirements for some values of n and m .

Bootstrapping analysis

To obtain error bars for the KL divergence, we used the following procedure. DDQCL was always executed 25 times with random initialization of the parameters. From the 25 repetitions, we sampled 10,000 data sets of size 25 with replacement and computed the median KL divergence

for each. From the distribution of 10,000 medians, we computed the median and obtained error bars from the 5-th and 95-th percentiles as the lower and upper limits, respectively, accounting for a 90% confidence interval.

For the case of qBAS score, we did the following bootstrap analysis. qBAS score was always computed 25 times from batches of samples N_{reads} . From the 25 repetitions, we sampled 10,000 data sets of size 25 with replacement and computed the mean for each. From the distribution of 10,000 means, we computed mean and obtained error bars from two standard deviations, accounting for a 95% confidence interval.

DATA AVAILABILITY

All data needed to evaluate the conclusions are available from the corresponding author upon request.

ACKNOWLEDGEMENTS

M.B. was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and by Cambridge Quantum Computing Limited (CQCL). The authors are very grateful to Prof. Christopher Monroe and his team at the University of Maryland (UMD) for their support in running the experiments presented here. Special thanks to N.M. Linke, C. Figgatt, K.A. Landsman, and D. Zhu for useful discussions and for running the several experiments used to test and validate the pipeline of this work, and the ones presented in the “Results” section. The authors acknowledge E. Edwards from the communications/publicity division at the Joint Quantum Institute at UMD for the rendering of the ion-trap graphic used in Fig. 2, and would like to thank J. Realpe-Gomez, A.M. Wilson, and G. Paz-Silva for useful discussions and feedback on an early version of this manuscript. The authors would like to thank J.I. Latorre for pointing out ref. ⁵⁶.

AUTHOR CONTRIBUTIONS

M.B. and A.P.-O. designed the generative model and the learning algorithm. M.B., Y.N. and A.P.-O. designed the data sets and the experiments. M.B. and V.L.-O. wrote the code and performed the in silico experiments. D.G.-P. wrote code for the analysis and figures. A.P.-O. designed the qBAS score. O.P. performed the analysis related to entanglement entropies. All authors analyzed the experimental results and contributed to the final version of the manuscript.

ADDITIONAL INFORMATION

Supplementary information accompanies the paper on the *npj Quantum Information* website (<https://doi.org/10.1038/s41534-019-0157-8>).

Competing interests: The authors declare no competing interests.

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

REFERENCES

- Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
- Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 EP (2014).
- McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
- Edward Farhi, S. G. & Goldstone, J. A. quantum approximate optimization algorithm. Preprint at arXiv:1411.4028 (2014).
- Hadfield, S. et al. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* **12**, 34 (2019).
- O’Malley, P. et al. Scalable quantum simulation of molecular energies. *Phys. Rev. X* **6**, 031007 (2016).
- Colless, J. I. et al. Computation of molecular spectra on a quantum processor with an error-resilient algorithm. *Phys. Rev. X* **8**, 011021 (2018).
- Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242 (2017).
- Moll, N. et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.* **3**, 030503 (2018).
- Otterbach, J. S. et al. Unsupervised machine learning on a hybrid quantum computer. Preprint at arXiv:1712.05771 (2017).
- Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2**, 045001 (2017).
- Lamata, L., Alvarez-Rodriguez, U., Martín-Guerrero, J., Sanz, M. & Solano, E. Quantum autoencoders via quantum adders with genetic algorithms. *Quantum Sci. Technol.* **4**, 014007 (2018).
- Li, R., Alvarez-Rodriguez, U., Lamata, L. & Solano, E. Approximate quantum adders with genetic algorithms: an IBM quantum experience. *Quantum Meas. Quantum Metrol.* **4**, 1–7 (2017).
- Farhi, E. & Harrow, A. W. Quantum supremacy through the quantum approximate optimization algorithm. Preprint at arXiv:1602.07674 (2016).
- Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J. & Biswas, R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Sci. Technol.* **3**, 030502 (2018).
- Harrow, A. W. & Montanaro, A. Quantum computational supremacy. *Nature* **549**, 203–209 (2017).
- Nathan Wiebe, K. M. S. & Kapoor, A. Quantum deep learning. Preprint at arXiv:1412.3489 (2015).
- Kieferová, M. & Wiebe, N. Tomography and generative training with quantum boltzmann machines. *Phys. Rev. A* **96**, 062327 (2017).
- Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Estimation of effective temperatures in quantum annealers for sampling applications: a case study with possible applications in deep learning. *Phys. Rev. A* **94**, 022308 (2016).
- Amin, M. H., Andriyash, E., Rolfe, J., Kulchitsky, B. & Melko, R. Quantum boltzmann machine. *Phys. Rev. X* **8**, 021050 (2018).
- Benedetti, M., Realpe-Gómez, J., Biswas, R. & Perdomo-Ortiz, A. Quantum-assisted learning of hardware-embedded probabilistic graphical models. *Phys. Rev. X* **7**, 041052 (2017).
- Benedetti, M., Realpe-Gómez, J. & Perdomo-Ortiz, A. Quantum-assisted helmholtz machines: a quantum–classical deep learning framework for industrial datasets in near-term devices. *Quantum Sci. Technol.* **3**, 034007 (2018).
- Witteck, P. & Gogolin, C. Quantum enhanced inference in Markov logic networks. *Sci. Rep.* **7**, 45672 (2017).
- Cheng, S., Chen, J. & Wang, L. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy* **20**, 583 (2018).
- Stoudenmire, E. & Schwab, D. J. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems* Vol. 29 (eds Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I. & Garnett, R.) 4799–4807 (Curran Associates, Inc., 2016).
- Han, Z.-Y., Wang, J., Fan, H., Wang, L. & Zhang, P. Unsupervised generative modeling using matrix product states. *Phys. Rev. X* **8**, 031012 (2018).
- Liu, D. et al. Machine learning by two-dimensional hierarchical tensor networks: a quantum information theoretic perspective on deep architectures. Preprint at arXiv:1710.04833 (2017).
- Gao, X., Zhang, Z. & Duan, L. A quantum machine learning algorithm based on generative models. *Sci. Adv.* **4**, eaat9004 (2018).
- Liu, J.-G. & Wang, L. Differentiable learning of quantum circuit born machines. *Phys. Rev. A* **98**, 062324 (2018).
- Verdon, G., Broughton, M. & Biamonte, J. A quantum algorithm to train neural networks using low-depth circuits. Preprint at arXiv:1712.05304 (2017).
- Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **10**, 631–633 (2014).
- Kerenidis, I. & Prakash, A. Quantum recommendation systems. Preprint at arXiv:1603.08675 (2016).
- Brandao, F. G. S. L. et al. Exponential quantum speed-ups for semidefinite programming with applications to quantum learning. Preprint at arXiv:1710.02581 (2017).
- Schuld, M., Fingerhuth, M. & Petruccione, F. Implementing a distance-based classifier with a quantum interference circuit. *EPL* **119**, 60002 (2017).
- Mohseni, M. et al. Commercialize quantum technologies in five years. *Nature* **543**, 171–174 (2017).
- Bishop, L. S., Bravyi, S., Cross, A., Gambetta, J. M. & Smolin, J. *Quantum volume*. Technical Report (2017).
- Linke, N. M. et al. Experimental comparison of two quantum computing architectures. *Proc. Natl Acad. Sci.* 201618020 (2017).
- Debnath, S. et al. Demonstration of a small programmable quantum computer with atomic qubits. *Nature* **536**, 63 EP (2016).
- Bengio, I. G. Y. & Courville, A. Deep learning. <http://www.deeplearningbook.org>. (MIT Press, 2016).
- Kullback, S. & Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **22**, 79–86 (1951).
- Kennedy, J. & Eberhart, R. Particle swarm optimization. *Proc. ICNN’95 Int. Conf. Neural Netw.* **4**, 1942–1948 (1995).
- Shi, Y. & Eberhart, R. A. modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, 69–73 (1998).

43. Sørensen, A. & Mølmer, K. Quantum computation with ions in thermal motion. *Phys. Rev. Lett.* **82**, 1971–1974 (1999).
44. Sørensen, A. & Mølmer, K. Entanglement and quantum computation with ions in thermal motion. *Phys. Rev. A* **62**, 022311 (2000).
45. Benhelm, J., Kirchmair, G., Roos, C. F. & Blatt, R. Towards fault-tolerant quantum computing with trapped ions. *Nat. Phys.* **4**, 463 EP (2008).
46. Maslov, D. & Nam, Y. Use of global interactions in efficient quantum circuit constructions. *New J. Phys.* (2017).
47. MacKay, D. J. C. *Information Theory, Inference and Learning Algorithms*. (Cambridge University Press, New York, 2002).
48. Greenberger, D. M., Horne, M. A., Shimony, A. & Zeilinger, A. Bell's theorem without inequalities. *Am. J. Phys.* **58**, 1131–1143 (1990).
49. Monz, T. et al. 14-qubit entanglement: creation and coherence. *Phys. Rev. Lett.* **106**, 130506 (2011).
50. Rocchetto, A. et al. Experimental learning of quantum states. *Sci. Adv.* **5** (2019).
51. Ozaeta, A. & McMahon, P. L. Decoherence of up to 8-qubit entangled states in a 16-qubit superconducting quantum processor. *Quantum Sci. Technol.* **4**, 025015 (2019).
52. Monz, T. et al. 14-qubit entanglement: creation and coherence. *Phys. Rev. Lett.* **106**, 130506 (2011).
53. Ricci-Tersenghi, F. The bethe approximation for solving the inverse ising problem: a comparison with other inference methods. *J. Stat. Mech. Theory Exp.* **2012**, P08015 (2012).
54. Mastromatteo, I. On the typical properties of inverse problems in statistical mechanics. Preprint at arXiv:1311.0190 (2013).
55. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for boltzmann machines. *Cogn. Sci.* **9**, 147–169 (1985).
56. Higuchi, A. & Sudbery, A. W. How entangled can two couples get? *Phys. Lett. A* **273**, 213–217 (2000).
57. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
58. Zhu, D. et al. Training of quantum circuits on a hybrid quantum computer. Preprint at arXiv:1812.08862 (2018).
59. Leyton-Ortega, V., Perdomo-Ortiz, A. & Perdomo, O. Robust implementation of generative modeling with parametrized quantum circuits. Preprint at arXiv:1901.08047 (2019).
60. Hamilton, K. E., Dumitrescu, E. F. & Pooser, R. C. Generative model benchmarks for superconducting qubits. Preprint at arXiv:1811.09905 (2018).
61. Du, Y., Hsieh, M.-H., Liu, T. & Tao, D. The expressive power of parameterized quantum circuits. Preprint at arXiv:1810.11922 (2018).
62. Johansson, J., Nation, P. & Nori, F. Qutip 2: a python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **184**, 1234–1240 (2013).
63. Miranda, L. J. V. Pyswarms, a research-toolkit for particle swarm optimization in python. <https://zenodo.org/badge/latestdoi/97002861> (2017).



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2019