A generative modeling approach for benchmarking and training shallow quantum circuits

Marcello Benedetti,[1,2] Delfina Garcia-Pintos,[3] Yunseong Nam,[4] and Alejandro Perdomo-Ortiz[5,6,3,1,2,*]

[1] *Department of Computer Science, University College London, WC1E 6BT London, UK*
[2] *Cambridge Quantum Computing Limited, CB2 1UB Cambridge, UK*
[3] *Qubitera, LLC., Mountain View, CA 94041, USA*
[4] *IonQ, Inc., College Park, MD 20740, USA*
[5] *Quantum Artificial Intelligence Lab., NASA Ames Research Center, Moffett Field, CA 94035, USA*
[6] *USRA Research Institute for Advanced Computer Science, Mountain View, CA 94043, USA*

(Dated: January 24, 2018)

Hybrid quantum-classical approaches, such as the variational quantum eigensolver, provide ways to use near-term, pre-fault-tolerant quantum computers of intermediate size 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 maximally entangled Greenberger-Horne-Zeilinger states, also known as "cat states". The circuit layout needed to prepare cat states for any number of qubits is then obtained by analyzing the resulting circuit patterns for odd and even number of qubits. We further demonstrate our hybrid 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 into the quantum distribution resulting from the shallow quantum circuit; an ideal benchmark for testing hardware proposal 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.

I. INTRODUCTION

What is a good metric for the computational power of noisy intermediate-scale quantum (NISQ) devices? [1] Can machine learning (ML) techniques provide ways to benchmark the power and usefulness of NISQ devices? How do we properly capture the performance scaling of these devices as a function of the circuit depth, the gate fidelity, or the qubit connectivity, for complex probabilistic tasks? In this work, we design a simple hybrid quantum-classical framework called data-driven quantum circuit learning (DDQCL) and address these questions with both simulation and experimental results. For our experiments, we use an ion trap quantum computer hosted at the University of Maryland [2]. We choose the ion trap architecture because of its versatile all-to-all qubit connectivity [3], allowing us to study several entangling circuit designs on the same device, addressing the connectivity topology question.

A key element of the hybrid approaches, such as the variational quantum eigensolver [4, 5] (VQE) or the quantum approximate optimization algorithm [6, 7] (QAOA), is the cost function to be optimized. In quantum chemistry applications [8–10], it often corresponds to the electronic Hamiltonian of a molecule. In the case of combinatorial optimization problems [6, 7, 11, 12], it corresponds to a classical Ising model describing the cost associated with each potential solution. In our case of generative ML task, all that is given as input is a data set, and all that is produced as an output is measurements. As shown in this work, there are several cost functions that could in principle be used to guide training, each one with their own advantages and disadvantages.

Although optimization tasks offer a great niche of applications, sampling tasks have most of the potential to prove quantum advantage in the near-term [13–15]. That is the case, for example, in the intractable training of unsupervised generative models, where quantum-assisted algorithms have been proposed for fault-tolerant gate-based [16, 17] and quantum annealing devices [18–22]. Here, we study a framework for gate-based NISQ devices, which is suitable for generative modeling and heavily relies on sampling. We use the 2^N amplitudes of the wave function that results from running a carefully designed N-qubit quantum circuit to construct and capture correlations observed in a dataset. Because in this model Born's rule determines the output probabilities, our work is also an early implementation of a Born machine [23], constructed to exploit all the resources available in the underlying NISQ hardware.

On the benchmarking of NISQ devices, quantum vol-

* Correspondence: alejandro.perdomoortiz@nasa.gov

ume [11, 24] 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 in estimating the computational power of a quantum computer. As a complementary metric based on a ML task on a canonical data set, in this paper, we propose *qBAS score*. The data set is easy to generate and is easily visualizable and verifiable for sizes up to hundreds of qubits. On the other hand, implementing a shallow circuit that can uniformly sample such data is hard, as it requires significant quantum resources in the form of entangling gates. Any miscalibration or environmental noise will thus affect this single performance number, enabling comparison between different devices or across different generations of the same device.

The structure of the article is the following. In Section II, we compare and contrast our framework with the current state-of-the-art. Section III explains the generative task and the learning algorithm. Section IV introduces the qBAS score. Section V presents results on three different data sets and section VI concludes with a discussion and potential future work.

II. RELATED WORK

Hybrid quantum-classical approaches have been one of the most successful strategies to cope with noise in near-term gate-based quantum computers. Starting with the ground-breaking work on the VQE [4, 5] as a proposal to extract properties of quantum systems, e.g. the electronic energy of molecules [8, 10], other researchers have extended its use to other domains, such as combinatorial optimization problems [6, 7, 11]. Recently, researchers at Rigetti Inc. provided a QAOA implementation for the *max-cut* problem, which in turn can be used to cluster synthetic data on a 19-qubit NISQ device [12]. Other successful hybrid approaches based on genetic algorithms were proposed for approximating quantum adders and training quantum autoencoders [25–27].

Our work differs from other implementations of Born machines in quantum-inspired ML algorithms [28–31]. With our focus in exploiting the full power of NISQ devices, we need not to rely on the efficient construction of quantum tensor networks and their efficient manipulation through graphical-processing units. This is in contrast as well with other quantum-assisted ML work (see e.g. Refs. [32–35]) which require fault-tolerant quantum computers; not expected to be readily available in the near-term.

Finally, our approach in training shallow circuits is fundamentally different from the recent work in Ref. [36]. We employ a one-shot gradient-free training that targets the data set directly, where the model consists of a quantum circuit itself. In Ref. [36], the hybrid algorithm is used to train a quantum circuit to approximately sample from a classical Gibbs distribution. The distribution is then used to update a classical model, hence requiring a full gradient-free optimization for each training iteration. For intermediate-sized data sets, the number of training iterations could vary from the hundreds to the thousands. Additionally, while the work in Ref. [36] requires compilation steps as part of the QAOA subroutines, ours is readily implementable in any available NISQ architecture, without incurring into this overhead. Our approach is also versatile enough to represent quantum models similar to that in Ref. [19], as well as to approximate classical models, as in the DDQCL approximate preparation of coherent thermal states in Section V.

III. 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, also known as generative modeling. We first describe the data and the model, then we present a simple training method. Technical details of our implementation may be found in Appendix A.

The data set $\mathcal{D} = (\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(D)})$ is a collection of 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 $\mathbf{x} \leftrightarrow |\mathbf{x}\rangle = |x_1 x_2 \cdots x_N\rangle$.

Provided with the dataset $\mathcal{D}$, our goal is now to obtain a good approximation to the target probability distribution $P_{\mathcal{D}}$. Our quantum circuit model, parametrized with $\boldsymbol{\theta}$ and with a fixed depth and gate layout, prepares a wavefunction $|\psi(\boldsymbol{\theta})\rangle$, from which probabilities are obtained according to Born's rule $P_{\boldsymbol{\theta}}(\mathbf{x}) = |\langle \mathbf{x}|\psi(\boldsymbol{\theta})\rangle|^2$. Following a standard approach from generative machine learning [37], we chose to minimize the Kullback-Leibler (KL) divergence [38] $D_{KL}[P_{\mathcal{D}}|P_{\boldsymbol{\theta}}]$ from the circuit distribution in the computational basis $P_{\boldsymbol{\theta}}$ to the target distribution $P_{\mathcal{D}}$. Minimization of this quantity is directly related to the minimization of a well known cost function: the negative log-likelihood $\mathcal{C}(\boldsymbol{\theta}) = -\frac{1}{D}\sum_{d=1}^{D} \log P_{\boldsymbol{\theta}}(\mathbf{x}^{(d)})$. However, there is a caveat; output probabilities have to be estimated from measurements but the exponentially large computational basis states potentially prevent us from measuring some of the relevant configurations with only finite number of measurements. This means $P_{\boldsymbol{\theta}}(\mathbf{x}^{(d)}) = 0$ for some $\mathbf{x}^{(d)}$ in the dataset, which in turn leads to infinite cost. To avoid singularities in the cost function, we use a simple variant

$$\mathcal{C}_{nll}(\boldsymbol{\theta}) = -\frac{1}{D}\sum_{d=1}^{D} \log \max(\epsilon, P_{\boldsymbol{\theta}}(\mathbf{x}^{(d)})), \qquad (1)$$

where $\epsilon > 0$ 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 Appendix B we discuss the advantages and disadvantages of three cost functions and compare them empirically.

After estimating the cost, we update the model parameter $\boldsymbol{\theta}$ to further minimize the cost. This process can in principle be done by any suitable classical optimizer. We used a gradient-free algorithm called particle swarm optimization (PSO) [39, 40] as previously done in the context of quantum chemistry [9]. In Appendix A, we provide details about our implementation. 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 Appendix A). 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 [41–44] (even layers) as our model.

It is important to note that in our model we fix the number of parameters, independent of the number of vectors in the dataset. This means we can hope to obtain a good approximation to the target distribution only if the aforementioned model is flexible enough to capture its complexity. Increasing the number of layers or the entangling layer's topology alter this flexibility, with the potential to enhancing 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. A direction to be explored in future work.

As summarized in Figure 1, the learning algorithm iteratively adjusts all the parameters to minimize the value of a user-defined cost function. We reiterate that, unlike other shallow quantum-circuit, parametric models such as in quantum chemistry or discrete optimization [8–11], our model does not come with a well-defined cost function or Hamiltonian, dictated by the problem. At any iteration, the user-defined cost is approximated using samples from the data set and measurements from the quantum hardware, hence the name data-driven quantum circuit learning (DDQCL).

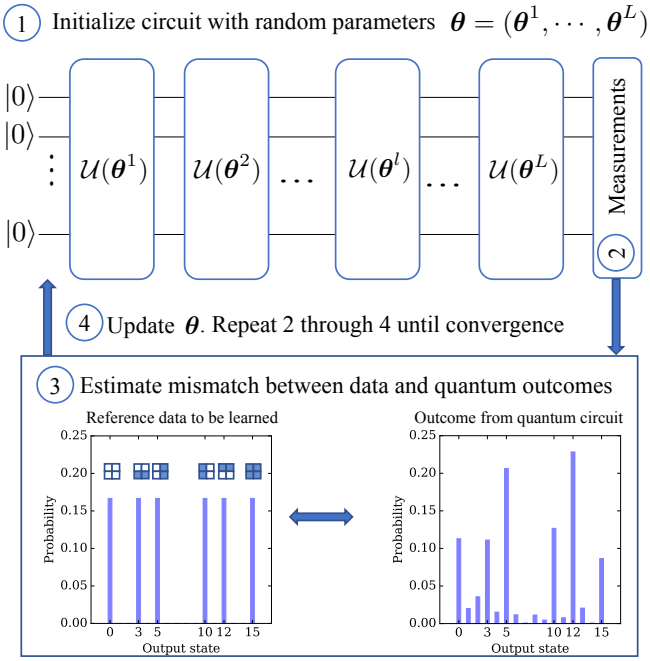

FIG. 1. *General framework for data-driven quantum circuit learning (DDQCL):* data vectors are interpreted as representative samples from an unknown probability distribution and the task is to unveil such distribution. The 2^N amplitudes of the wave function resulting from an N-qubit quantum circuit are used to capture the correlations observed in the data. Training of the quantum circuit is achieved by successive updates of the parameters $\boldsymbol{\theta}$, corresponding to specifications of single qubit operations and entangling gates. In this work, we use single qubit arbitrary 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. Many cost functions could be used, each one with its advantages and disadvantages. Here we used a clipped version of the negative log-likelihood of the data, although we explored other cost function in Appendix B.

IV. THE qBAS SCORE

Bars and stripes (BAS) [45] 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 $N_{\mathrm{BAS}nm} = 2^n + 2^m - 2$ images belonging to BAS and they can be efficiently produced (see Appendix A for details). The probability distribution is then $P_{\mathrm{BAS}nm}(\boldsymbol{x}^{(d)}) = 1/N_{\mathrm{BAS}nm}$ for all the $\boldsymbol{x}^{(d)}$ in the BASnm data set, and zero otherwise. Figure 2 on the top left panel shows patterns belonging to BAS22, while the top central panel shows the remaining patterns.

We can use DDQCL to learn a circuit that encodes all the BAS patterns in the wave function of a quantum state. This allows us to design the qBASnm 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/limitations in solving a complex sampling task that requires a fair amount of entanglement resources. It takes into account the model circuit

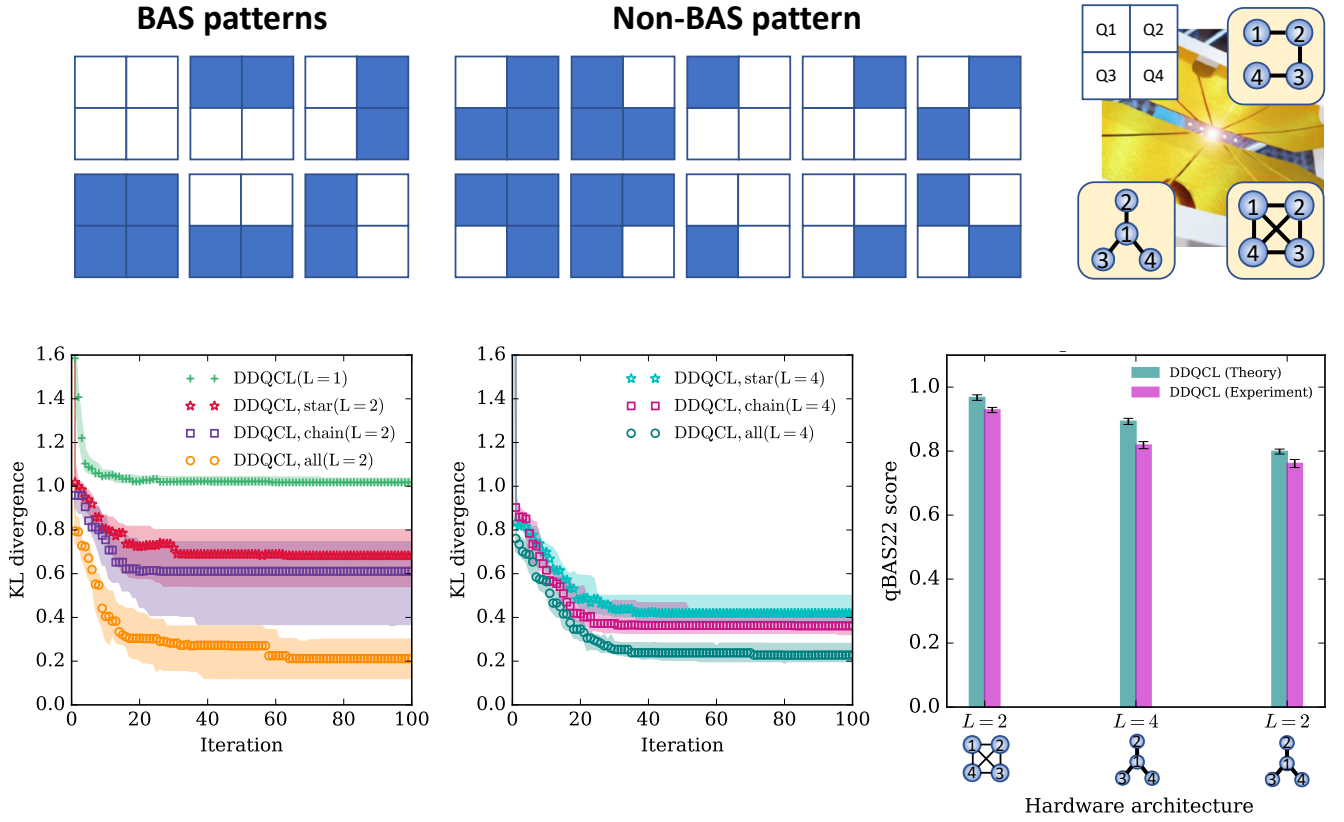

FIG. 2. *DDQCL on the bars and stripes (BAS) data set.* The top left corner shows patterns that belong to BAS22 and that we would like our quantum circuit to generate. For completeness, the top central image shows undesired patterns. On the top right corner, we show a possible mapping of the 4 pixels to $N = 4$ qubits, and we show some of the entangling layer topologies that can be set up in the ion trap (e.g *chain*, *star*, and *all*). The bottom left corner shows results of DDQCL simulations for 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 angles 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 approximatively twice the number of parameters. All the topologies achieve a lower median KL divergence and the confidence intervals shrink. The bottom right corner shows the bootstrapped mean qBAS22 and 95% confidence interval for simulations (green bars) and experiments on the ion trap quantum computer hosted at University of Maryland (pink bars).

depth, gate fidelities, and any other architectural design aspects such as its qubit-qubit connectivity, in addition to the native set of single and two-qubit gates available in hardware.

When framed in the context of information retrieval, the qBASnm score can be seen as an instantiation of the widely used F_1 score. To score high, it is insufficient to simply retrieve states, which belong to BASnm. This quantity alone corresponds to the so called *precision* (denoted here as p), and it determines the ratio between the number of measurements belonging to BASnm divided by the total number of measurements [46]. One also needs to score high in the so called *recall* (denoted here by r) which determines the capacity of the circuit model to retrieve the whole spectrum of patterns belonging to the BASnm. In our context, it is a measure of "fair sampling", or the capacity to uniformly retrieve all the states from BASnm. Within the F_1 score, recall is a general quantity that can always be computed as the number of different BASnm patterns appearing in the N_{reads} measurements divided by the total number of states $N_{\text{BAS}nm}$ that belong to the data set. If we denote the number of different patterns that were measured as $d(N_{\text{reads}})$, then $r = d(N_{\text{reads}})/N_{\text{BAS}nm}$. The F_1 score is defined as the harmonic mean of the precision and the recall, i.e., $F_1 = 2pr/(p + r)$, and to score high ($F_1 \approx 1.0$) it is required to have both a high precision ($p \approx 1.0$) and high recall in retrieving of all the $N_{\text{BAS}nm}$ patterns ($r \approx 1.0$). 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 taking a large number of measurements $N_{\text{reads}} \to \infty$. This is not desirable since our purpose is to evaluate circuit proposals on the task of uniformly sampling all the patterns from BASnm. To define a unique metric we fix N_{reads} to be just high enough to obtain a $r \approx 1.0$, under the assumption of probability distribution equal to the target distribution $P_{\text{BAS}nm}$. Any other distribution deviating from the target will have an impact in the value of r, and the larger the deviation (presence of biases towards some of the patterns), the larger the gap from ideal value of $r = 1.0$. We exploit the fact that the probability of seeing any single pattern is identical and equal to $P_s = 1/N_{\text{BAS}nm}$ for the case of the target distribution. Therefore, we need to define the value of N_{reads} such that with high confidence (with probability P_C) all the desired patterns appear. Assuming P_s is the probability for successfully observing an event, the probability of not observing the event after N_{reads} trials is equal to $(1-P_s)^{N_{\text{reads}}}$. We want the probability of such a scenario to be small and equal to $1 - P_C$. By equating the two terms and solving for N_{reads}, we obtain,

$$N_{\text{reads}}[P_C] = \left\lceil \frac{\log(1-P_C)}{\log(1-P_s)} \right\rceil . \tag{2}$$

Since P_s is automatically fixed by the calculation of $N_{\text{BAS}nm}$, the only free parameter left is P_C. We arbitrarily chose the desired high confidence to be 95%, i.e. $P_C = 0.95$. Table I shows the value of $N_{\text{BAS}nm}$ and the number of required measurements for every estimation of the recall with the proposed approach, of $P_C = 0.95$.

For statistical robustness of the metric, 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 by averaging over each p_i. From the estimate for p, we can compute R independent values of qBASnm from each of the r_i,. These are subsequently bootstrapped to obtain a more robust average for the final reported value of qBASnm (see details in Appendix A 4).

We note that a more general performance indicator than qBASnm score is indeed the KL divergence, $D_{KL}[P_{\text{BAS}nm}|P_{\boldsymbol{\theta}}]$. However, this would not be robust in terms of scalability. As $n \times m$ becomes large, while the number of measurements $N_{\text{reads}}[P_C]$ necessary for obtaining a robust qBASnm score continues to remain relatively small to be practical for intermediate size $n \times m$, it is expected that, frequently, the KL divergence is undefined [47]. This is true for all typical cases where measurements yield distributions such that $P_{\boldsymbol{\theta}}(\boldsymbol{x}^{(d)}) = 0$, for any of the $\boldsymbol{x}^{(d)}$ in BASnm. Even in the small case considered in this paper, BAS22, it is not a straightforward task to learn a quantum circuit preparing a good wave function where every desired pattern has significant probability. In all these cases, the qBASnm score can still be estimated.

V. RESULTS

We tested DDQCL via simulations and experiments on various synthetic probability distributions. To evaluate the performance, we computed the KL divergence $D_{KL}[P_{\mathcal{D}}|P_{\boldsymbol{\theta}}]$ from the circuit distribution in the computational basis $P_{\boldsymbol{\theta}}$ to the target distribution $P_{\mathcal{D}}$. The KL divergence is in general intractable, but the size of our experiments allowed us to compute this quantity exactly. On the other hand, the training was carried out using samples. Each data set consisted of 1000 data points from the target distribution and, at each iteration, we performed a fixed number of 1000 measurements in the computational basis.

A. GHZ state preparation

To test the capabilities of DDQCL, we started with the preparation of GHZ states or cat states [48]. Besides their importance in quantum information, the choice is motivated by their simple description, and the many studies around their preparation (see e.g., [49–51]). From the DDQCL perspective, we explored whether it is possible to learn any of the known recipes for GHZ state preparation by starting only from classical data. More specifically, the input data consists of samples from a distribution corresponding to the two desired states of the computational basis; $P_0 = 0.5$ for the state $|0\ldots0\rangle$ and $P_1 = 0.5$ for the state $|1\ldots1\rangle$. To our surprise, DDQCL yielded many degenerate but different preparations of GHZ-like states, only differing by a relative phase between the two main states.

We ran 25 random initializations of the particle swarm optimization for 3, 4, 5 and 6 qubit instances. A human expert inspected the best set of angles learned for each size and after removing some noise, e.g. rounding the angles 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 Figure 3 and were verified for larger number of qubits, both odd and even. DDQCL successfully reproduced precisely the recipes previously used on ion trap quantum computers [52], and, to the best of our knowledge, it corresponds to the most compact and efficient protocol for GHZ state preparation using GMS gates. Another commonly used approach consists of cascading entangling gates, with alternations of single qubit rotations (see e.g. Ref. [51]), and DDQCL was able to approximate such recipe as well for the test cases of 3 and 4 qubits.

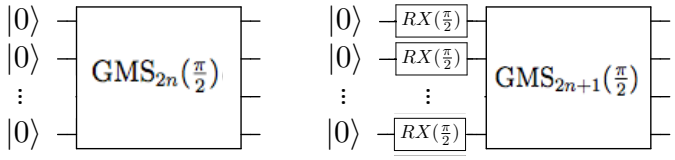

FIG. 3. *GHZ-like state preparation assisted by DDQCL:* Left (right) panel shows the recipe obtained by using DDQCL with $L = 2$ for the even (odd) cat-state preparation circuit up to a relative phase between the output integer states $|0\ldots0\rangle$ and $|1\ldots1\rangle$. Notations RX and GMS are taken verbatim from [44]. $RX(\theta)$ stands for the conventional, single-qubit rotation $e^{-i\hat{\sigma}_x\theta/2}$, where $\hat{\sigma}_x$ is the Pauli-x matrix. GMS stands for a global Mølmer-Sørensen gate [44] acting on all the $N = 2n$ ($N = 2n+1$) qubits and it is equivalent to the application of local XX gates to all $N(N-1)/2$ pairs of qubits.

It is interesting to note that in DDQCL all the angle 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. Finally, it is reassuring that DDQCL obtained many circuits for cat state preparation circuits starting from samples of a classical distribution. While this target distribution could have been modeled by a classical zero-temperature ferromagnet, there apparently was no other way for our circuit to reproduce the statistics, if not by preparing a GHZ-like state.

B. 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 $N = 5$. By decreasing the temperature T of the target distribution, we can increase the difficulty of the learning task (details can be found in Appendix A). 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 stars) consistently beat shallower circuits such as $L = 2$ (red circles) and $L = 1$ (yellow triangles). This becomes more evident as we go from easy learning tasks (Fig. 4(a)) to hard learning tasks (Fig. 4(c)). Results for instances of six spins ($N = 6$) are shown in Appendix C.

To assess how well our DDQCL performs on the generative task, we now compare our results with those obtained from a classical approach. In particular, we aim to answer the question of whether we can infer the parameters of the target distribution without resorting to quantum state preparation. To this end, we compared DDQCL to a classical closed-form approach, the inverse Bethe approximation [53] (see also Eqs. (3.21) and (3.22) in Ref. [54]). As shown in Fig. 4, the inverse Bethe approximation (green bar) performs extremely well in the easy task (a), matches the $L = 3$ quantum circuit in the intermediate task (b), and underperforms 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 T_c. It is thus not surprising that we obtain bad performance in Fig. 4 (c) with the inverse Bethe approximation. We also note in passing that there exist several other classical methods that could have performed better in inferring the parameters, e.g. Boltzmann machines [55].

C. BAS22

We computed experimentally and theoretically the qBAS22 score for some of the possible choices of entangling layer topology (sketched in the top right panel of Figure 2), and different number of layers. The process consists of two steps; first, DDQCL is used to encode BAS22 in the wave function of the quantum state. Second, the best circuits, i.e. those with lowest cost function, are compared using the qBAS22 score.

The bottom left and bottom central panels in Figure 2 show the bootstrapped median of the KL divergence and 90% confidence interval over 25 random restarts of DDQCL in silico. Supported by two empirical observations, we claim that entanglement is a key resource for learning the BAS data set. First, the mean-field-like circuit $L = 1$ (green crosses) severely underperforms compared to other circuit layouts. Second, when inspecting the angles learned for all-to-all circuits with $L = 2$ (orange circles), we found that most of the XX gates reached their maximum entangling setting. 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 (Appendix 7 shows evidence that this is the case).

From this vast collection of circuits, we chose the best all-to-all with two layers and the best star-shaped with

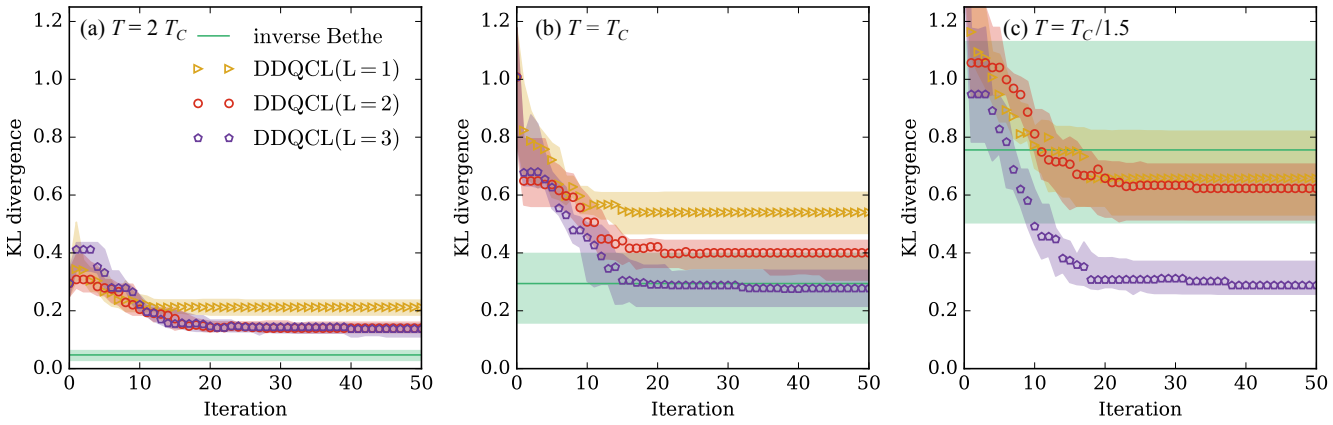

FIG. 4. *DDQCL preparation of coherent thermal states:* we generated 25 random Ising instances of size $N = 5$ and varied the difficulty of the learning task by decreasing the temperature in $T \in \{2T_c, T_c, T_c/1.5\}$ (see Appendix A for details). The model is a quantum circuit with five qubits and an all-to-all qubit connectivity for the entangling layer in the second layer ($l = 2$). 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 both shallow ones. Despite the large optimization space, 35 parameters for $L = 3$, DDQCL can prepare good approximations to the desired thermal states. 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 model so obtained (green band) is excellent classical for the easy task in (a), matches the best quantum model in (b), and underperforms for the hard task in (c).

two and four layers. Those very different approximate solutions to the same problem can be compared at the level of qBAS22 score. In silico simulations allow us to obtain the theoretical amplitudes for each of the states in the computational basis. From the six patterns belonging to BAS22, we can compute the theoretical expected value for the precision p, for the case of an error-free quantum device. To compute the recall and simulate experimental readouts, we can use this multinomial distribution over the 2^4 states to obtain $R = 25$ batches each with a size of $N_{\text{reads}}[95\%] = 17$ samples from this distribution, as described in Sec. IV. From each batch, we compute the R estimates of r_i, and along with p we compute R estimates for qBAS22 score. By bootstrapping these R values (see Appendix A), we can obtain a robust estimation of the qBAS score and its confidence interval. To have comparable experimental results, we take a large number of readouts [56] to have an accurate approximation to the multinomial experimental distribution over all of the 2^4 states. This avoids finite sampling factors from the experimental readouts, and allows us to compare with the theoretical results obtained from the error-free quantum circuit simulations. Any deviations are therefore attributed to for instance imperfect hardware, miscalibration, or misspecification of the programmable parameters defining the desired quantum circuit. As in the case of the in silico estimation of the qBAS score, we sample $R = 25$ data sets of size 17 from the experimental multinomial distribution. From each of the data sets, we compute a qBAS22 score. The final mean value and its 95% confidence interval are then obtain by bootstrapping. The bottom right panel in Figure 2 shows the score for the three circuits on both simulator (green bars) and ion trap (pink bars).

The score is clearly sensitive to the depth of the circuit as shown by the performance improvement of $L = 4$ compared to $L = 2$ in the star topology. Note that the theoretical improvement for using $L = 4$ is larger than what was observed experimentally. This shows how the error accumulation in the quantum computer running a deeper circuit impacts the score. The score is also sensitive to the choice of topology as shown by the drop in performance of the star-shaped compared to all-to-all when the same number of layers $L = 2$ is used. Although we implemented these circuits on the same ion trap hardware, the score could have been used to compare different device generations or even completely different architectures (e.g. superconductor-based versus atomic-based).

VI. SUMMARY AND OUTLOOK

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 ML generative models. The success of this approach is evidenced by the results on the three different data sets in Sec. V.

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 and illustration of how simple classical probability distributions can guide the synthesis of an interesting non-trivial quantum state. 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 machines to approximate Boltzmann machines when the data require such 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 entangling. Errors present in the device will affect this single performance measure, the qBAS score, which can be compared between different devices or across generations of the same device.

Besides the benefit of providing a tool to measure the power of specific hardware architectures and circuit layouts, qBAS score can also be used to benchmark the typical performance of black-box solvers (e.g. PSO, SPSA and Bayesian optimization to name a few) used in hybrid quantum-classical approaches for NISQ devices. Selecting the optimal method and optimizing the hyperparameters for each of these solvers can be a daunting task and is a key challenge towards a successful implementation of the solvers as the number of qubits and layers grow. Therefore, we believe having this unique metric in these benchmarks could help reduce the complexity of this fine-tuning stage. The score can be computed in any of the NISQ architectures available to date, no matter how noisy or regardless of their qubit-to-qubit connectivity limitations.

It is left to future work to show how one can perform 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 implemented at the level of the cost function. In this paper, we used the negative log-likelihood which quickly becomes expensive to estimate as the size of the system increase. We reported preliminary results on alternative cost-functions that overcome the caveat and still produce satisfactory results.

Ongoing work includes the experimental realization of the full DDQCL training on the quantum computer and the subsequent evaluation of the qBAS score from the resulting quantum circuit. To the best of our knowledge, our approach represents the first proposal with 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 developing further concrete metrics to help guide the architectural hardware design development, while tapping into the computational power of NISQ devices.

VII. ACKNOWLEDGEMENTS

M. Benedetti 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 here. The authors also would like to 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 to J. Realpe-Gomez, A. M. Wilson, V. Leyton-Ortega, and G. Paz-Silva for useful discussions and feedback on an early version of this manuscript.

Appendix A: Methods

1. Simulation of quantum circuits in silico

We simulate quantum circuits using the `QuTiP2` [57] Python library and implemented the constraints dictated by the ion trap experimental setting. In the current experimental setup, we can build circuits out of arbitrary single qubit rotations and Mølmer-Sørensen XX entangling gates involving any two qubits. We use only these gates in the ansatz circuit hence avoiding the need of further compilation. For the simulations we also assume perfect gate fidelities and error-free measurements.

In the ion trap setting, the implementation of single qubit R_z rotations is very convenient. Therefore, we perform arbitrary single gate 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 labels the layer number, i labels the qubit index, and $\theta_i^{(l,k)} \in [-\pi, +\pi]$ corresponds to Euler angles. The rotations are then expressed as exponentials of Pauli operators $R_z(\theta_i^{(l,\cdot)}) = \exp(-\frac{i}{2}\theta_i^{(l,\cdot)}\sigma_i^z)$ and $R_x(\theta_i^{(l,\cdot)}) = \exp(-\frac{i}{2}\theta_i^{(l,\cdot)}\sigma_i^x)$. In cases where we start the simulation from the $|0\cdots0\rangle$ state, the first set of R_z rotations corresponding to $\theta_i^{(l,1)}$ is not needed. In cases where we start from a uniform superposition of all the basis states (e.g. we apply Hadamard gates to all the qubits), we can neglect the R_z rotations corresponding to $\theta_i^{(l,3)}$. With the exception of the first layer, $l = 1$, which requires $2N$ parameters, every layer of arbitrary single qubit operations requires $3N$ parameters.

For the case of the entangling gates, we will use the notation $U_{ij}^{(l)} = XX(\theta_{ij}^{(l)})$ where we have a parameter $\theta_{ij}^{(l)} \in [-\frac{\pi}{2}, +\frac{\pi}{2}]$. In exponential form, the entangling gates reads as $XX(\theta_{ij}^{(l)}) = \exp(-\frac{i}{2}\theta_{ij}^{(l)}\sigma_i^x\sigma_j^x)$. 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, *line* 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 Figure 2 shows a graphical representation of these topologies for the case of $N = 4$ qubits.

At the end of each simulation, we perform 1000 measurements in the computational basis and use them to estimate the relevant quantities.

2. Gradient-free optimization

Given a fixed number of layers and topology of entangling gates, the quantum circuits described above provide a template; by adjusting the parameters we can implement a small portion of all 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 (more on this in the Appendix B).

We use a global-best particle swarm optimization algorithm [40] implemented in the `PySwarms` [58] Python library. A 'particle' corresponds to a candidate solution circuit; the position of a particle is a point $\boldsymbol{\theta}$ 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 Figure 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 $\frac{\pi}{2}$.

Finally, we set the number of particles to twice the number of parameters of the ansatz circuit. This is a conservative value compared to previous work [10], also because of the large number of angles in the circuits explored here.

3. Data sets details

We worked with three synthetic datasets: 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.

a. GHZ state preparation

The zero-temperature ferromagnet distribution is equivalent to assigning 1/2 probability to both $|0\ldots0\rangle$ and $|1\ldots1\rangle$ states of the computational basis. This distribution can be easily prepared as a mixed stated, but our study uses pure states prepared by the ansatz circuit. The only way to reproduce the zero-temperature ferromagnet distribution in our setting is to implement a transformation that prepares a GHZ-like state.

b. Thermal states

A thermal dataset in N dimensions is generated by exact sampling realizations of $\mathbf{x} \in \{-1, +1\}^N$ from the distribution $P(\mathbf{x}) = Z^{-1}\exp((\sum_{ij} J_{ij}x_ix_j + \sum_i h_ix_i)/T)$ 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.

c. Bars and Stripes

BAS [45] 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 from partial or corrupted data and generate previously unseen patterns from the target probability distribution. On the other hand, we could provide the algorithm with *all* the patterns we are interested in and aim to a model that generates only those, that is, an associative memory. Although both tasks can be done with our DDQCL pipeline, for the qBASnm score we focus in the latter task and therefore we need to determine the number of patterns and provide an easy identification of the bitstring belonging to the BASnm class.

For the total count of the number of patterns, we first count the 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:

$$N_{\mathrm{BAS}nm} = 2^n + 2^m - 2. \tag{A1}$$

In the main text, we use the BAS dataset to design a task-specific performance indicator for quantum hardware. Table I shows the requirements for some values of n and m.

(n,m)	N_{qubits}	$N_{\mathrm{BAS}nm}$	$N_{\mathrm{reads}}[95\%]$
(2,2)	4	6	17
(2,3)	6	10	29
(3,3)	9	14	41
(4,4)	16	30	89
(7,7)	49	254	760
(8,8)	64	510	1527
(10,10)	100	2046	6128

TABLE I. Example of experimental requirements for near-term quantum computers with up to 100 qubits. As described in the main text, $N_{\mathrm{reads}}[95\%]$ is the number of readouts required for every estimation of qBAS score.

4. Bootstrapping analysis

For the case of qBAS scores, we did the following bootstrap analysis: from the R estimations of the qBAS score corresponding to the calculation from each r_i, we sampled 10,000 data sets of size R with replacement, and computed the mean for each. The bootstrapped qBAS score corresponds to the mean over the distribution of the means. The standard deviation σ of the mean is then used for error bars. We plot 95% confidence interval, or 2σ.

For all other plots, corresponding to the KL divergences in each of the DDQCL curves, we used the following procedure: from the 25 repetitions of the algorithm, we sampled 10,000 data sets of size 25 with replacement, and computed the median as an estimate of the typical DDQCL performance. From the distribution of 10,000 medians, the error bars were calculated from the 5-th and 95-th percentiles as the lower and upper limits, respectively, accounting for a 90% confidence interval.

Appendix B: Comparison of cost functions

In realistic machine learning scenarios we don't have access to the true probabilities of both target and circuit. Hence, we need to compare the two distributions at the level of histograms and using a finite number of samples and measurements, respectively. Here we compare three cost functions via simulations in silico.

a. Clipped negative log-likelihood:

$$\mathcal{C}_{nll}(\boldsymbol{\theta}) = -\frac{1}{D}\sum_{d=1}^{D} \log \max(\epsilon, P_{\boldsymbol{\theta}}(\mathbf{x}^{(d)})), \tag{B1}$$

where probabilities are estimated from samples and $\epsilon > 0$ is a small number that avoids an infinite cost when $P_{\boldsymbol{\theta}}(\mathbf{x}^{(d)}) = 0$. This may happen if for example entanglement in the circuit prevents us from measuring configuration $\mathbf{x}^{(d)}$. Moreover, when the number of variables N is large, all except few configurations will ever be measured due to the finite number of samples. Note that by re-normalizing all the probabilities after the clipping, we could interpret this variant as a Laplace additive smoothing. All the experiments in the main text are carried out using the clipped negative log-likelihood with $\epsilon = 10^{-8}$.

b. Earth mover's distance [59]:

$$\mathcal{C}_{emd}(\boldsymbol{\theta}) = \min_{F} \langle d(\mathbf{x},\mathbf{y})\rangle_F, \tag{B2}$$

where $F(\mathbf{x},\mathbf{y})$ is a joint probability distribution such that $\sum_{\mathbf{y}} F(\mathbf{x},\mathbf{y}) = P_{\mathcal{D}}(\mathbf{x})$ and $\sum_{\mathbf{x}} F(\mathbf{x},\mathbf{y}) = P_{\boldsymbol{\theta}}(\mathbf{y})$, that is, its marginals correspond to the data and circuit distributions, respectively. Intuitively, this is the minimum

cost of turning one histogram into the other where the ground metric $d(\mathbf{x}, \mathbf{y})$ specifies the cost of transporting a single unit from $\mathbf{x}$ to $\mathbf{y}$. We chose $d(\mathbf{x}, \mathbf{y})$ to be the Hamming distance between strings $\mathbf{x} \in \{-1, +1\}^N$ and $\mathbf{y} \in \{-1, +1\}^N$. Since we normalize histograms to sum up to one, the Earth Mover's Distance is equivalent to the 1-st Wasserstein distance [60]. In our simulations, we use the `PyEMD` Python library for fast computation of the earth moving distance [61].

c. Moment matching via mean squared error:

$$\mathcal{C}_{mm}(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i}^{N}(\langle x_i\rangle_{P_{\mathcal{D}}} - \langle x_i\rangle_{P_{\boldsymbol{\theta}}})^2 + \frac{2}{N(N-1)}\sum_{i>j}^{N}(\langle x_i x_j\rangle_{P_{\mathcal{D}}} - \langle x_i x_j\rangle_{P_{\boldsymbol{\theta}}})^2, \tag{B3}$$

where the expectation values $P_{\mathcal{D}}$ and $P_{\boldsymbol{\theta}}$ are taken with respect to data and circuit distributions, respectively. This cost function can be generalized to include moments beyond the second as well as using different positive exponents for the error.

We compared the cost functions on the task of learning thermal states of size $N = 5$, with $L = 3$ layers and *all* topology. Figure 5 shows the bootstrapped median KL divergence on 25 realizations and 95% confidence interval as learning progress for 100 iterations. The fact that $\mathcal{C}_{nll}$ (red diamonds) outperforms other cost functions does not come as a surprise; minimization of the negative log-likelihood is indeed directly related to minimization of the KL divergence. However, we expect the performance of DDQCL based on this cost function to degrade quickly as the size of the problem increases. In realistic applications, the relevant probabilities in Eq. (B1), i.e. those associated with the data, are a vanishing fraction of the 2^N probabilities. Moreover, they need to be estimated from a finite number of measurements. The earth mover's distance $\mathcal{C}_{emd}$ (green stars) performs well, but it suffer from similar scalability issues. Fast algorithms for the computation of this distance may struggle when the number of bins in the histogram increases exponentially as in our case. However, it is reassuring to see that alternative cost functions with no relation to the KL divergence can still produce satisfactory results. Surprisingly, the moment matching $\mathcal{C}_{mm}$ (purple crosses) closely tracks the other cost functions while retaining computational efficiency. In fact, even though a large number of samples may be needed to obtain low-variance estimates for the expectation values, only $\mathcal{O}(N^2)$ terms are computed at each iteration. We expect this cost function to be a good heuristic for DDQCL on large systems.

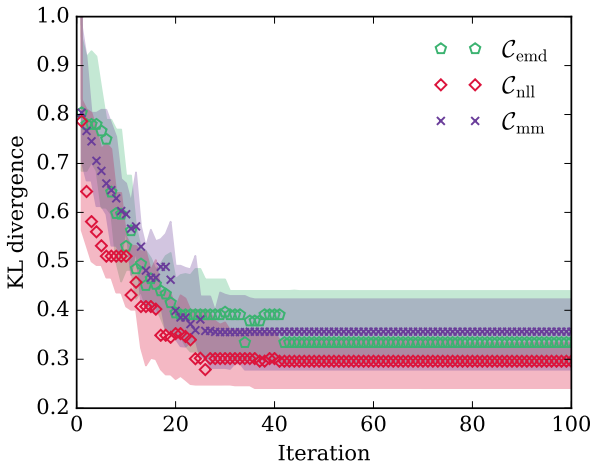

FIG. 5. Bootstrapped median KL divergence and 90% confidence interval of circuits learned by minimizing different cost functions. Both the moment matching ($\mathcal{C}_{mm}$) and earth mover's distance ($\mathcal{C}_{emd}$) closely track the clipped negative log-likelihood ($\mathcal{C}_{nll}$) used in the main text.

Appendix C: Approximate preparation of coherent thermal states for $N = 6$

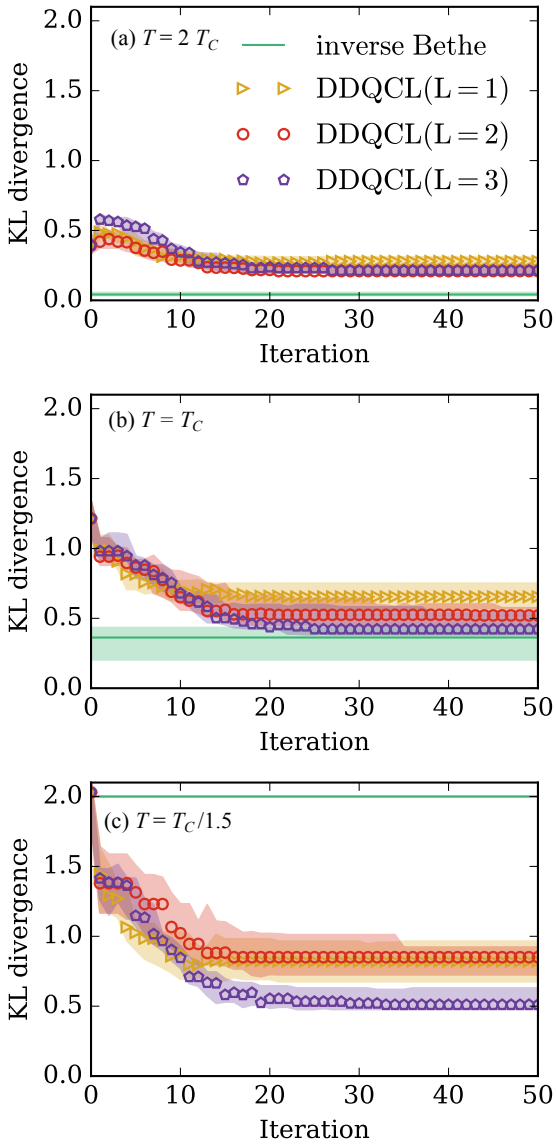

FIG. 6. DDQCL preparation of approximate coherent thermal states for the case of six qubits and different circuit depths $L \in \{1, 2, 3\}$. In these simulations, an all-to-all qubit connectivity was used for the entangling gate layer ($l = 2$). The median values for all methods are obtained from bootstrapping an ensemble of 25 random instances, with error bars corresponding to the 90% confidence interval over the distribution of medians. The value reported for the inverse Bethe approximation in the case of the lowest temperature panel (c), is much higher since only 7 out of 25 instances converged. Therefore, no median value was extracted. We plotted a value of KL = 2.0 as a reference, but it is clear that for those instances DDQCL outperformed the inverse Bethe approximation method.

Appendix D: Details for theoretical and experimental results

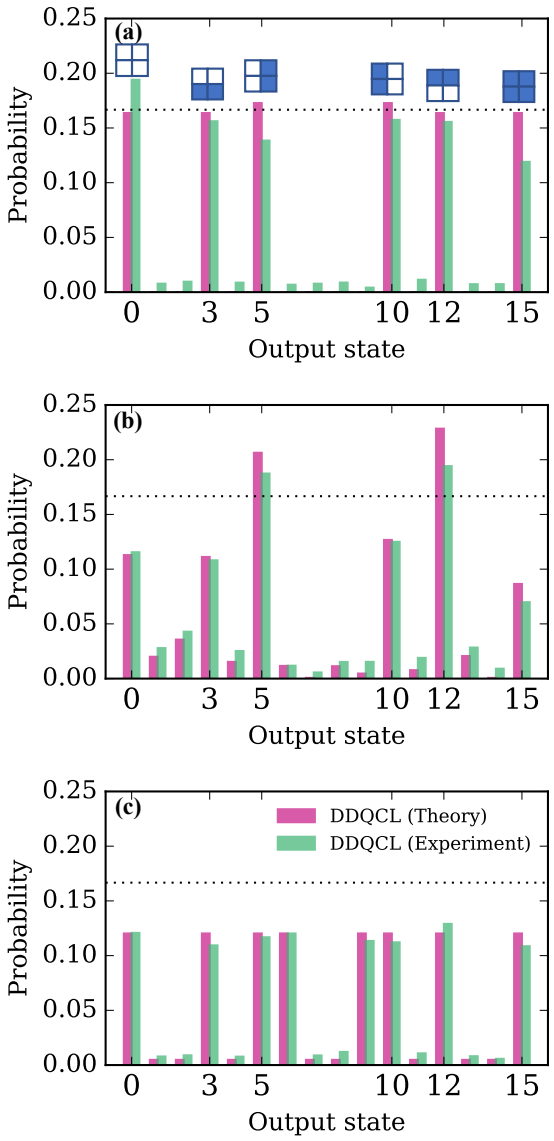

FIG. 7. Detailed comparison at the level of output states between simulation of circuits obtained with DDQCL and the respective experimental implementations of the final optimized circuits. Here we focus only on the star and all-to-all topologies, and with two circuit depths: (a) All-to-all with $L = 2$, (b) star with $L = 4$, and (c) star with $L = 2$.

[1] John Preskill, "Quantum computing in the nisq era and beyond," arXiv:1801.00862 (2018).

[2] S. Debnath, N. M. Linke, C. Figgatt, K. A. Landsman, K. Wright, and C. Monroe, "Demonstration of a small programmable quantum computer with atomic qubits," Nature **536**, 63 EP – (2016).

[3] Norbert M Linke, Dmitri Maslov, Martin Roetteler, Shantanu Debnath, Caroline Figgatt, Kevin A Landsman, Kenneth Wright, and Christopher Monroe, "Experimental comparison of two quantum computing architectures," Proceedings of the National Academy of Sciences , 201618020 (2017).

[4] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien, "A variational eigenvalue solver on a photonic quantum processor," Nature Communications **5**, 4213 EP – (2014).

[5] Jarrod R McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," New Journal of Physics **18**, 023023 (2016).

[6] Sam Gutmann Edward Farhi, Jeffrey Goldstone, "A quantum approximate optimization algorithm," arXiv:1411.4028 (2014).

[7] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas, "From the quantum approximate optimization algorithm to a quantum alternating operator ansatz," arXiv:1709.03489 (2017).

[8] PJJ O'Malley, Ryan Babbush, ID Kivlichan, Jonathan Romero, JR McClean, Rami Barends, Julian Kelly, Pedram Roushan, Andrew Tranter, Nan Ding, *et al.*, "Scalable quantum simulation of molecular energies," Physical Review X **6**, 031007 (2016).

[9] James I Colless, Vinay V Ramasesh, Dar Dahlen, Machiel S Blok, Jarrod R McClean, Jonathan Carter, Wibe A de Jong, and Irfan Siddiqi, "Robust determination of molecular spectra on a quantum processor," arXiv preprint arXiv:1707.06408 (2017).

[10] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta, "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets," Nature **549**, 242 (2017).

[11] Nikolaj Moll, Panagiotis Barkoutsos, Lev S. Bishop, Jerry M. Chow, Andrew Cross, Daniel J. Egger, Stefan Filipp, Andreas Fuhrer, Jay M. Gambetta, Marc Ganzhorn, Abhinav Kandala, Antonio Mezzacapo, Peter Müller, Walter Riess, Gian Salis, John Smolin, Ivano Tavernelli, and Kristan Temme, "Quantum optimization using variational algorithms on near-term quantum devices," arXiv:1710.01022 (2017).

[12] JS Otterbach, R Manenti, N Alidoust, A Bestwick, M Block, B Bloom, S Caldwell, N Didier, E Schuyler Fried, S Hong, *et al.*, "Unsupervised machine learning on a hybrid quantum computer," arXiv preprint arXiv:1712.05771 (2017).

[13] Edward Farhi and Aram W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," arXiv:1602.07674 (2016).

[14] Alejandro Perdomo-Ortiz, Marcello Benedetti, John Realpe-Gómez, and Rupak Biswas, "Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers," arXiv:1708.09757 (2017).

[15] Aram W. Harrow and Ashley Montanaro, "Quantum computational supremacy," Nature **549**, 203–209 (2017).

[16] Krysta M. Svore Nathan Wiebe, Ashish Kapoor, "Quantum deep learning," arXiv:1412.3489 (2015).

[17] Mária Kieferová and Nathan Wiebe, "Tomography and generative training with quantum boltzmann machines," Phys. Rev. A **96**, 062327 (2017).

[18] Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz, "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).

[19] Mohammad H. Amin and Evgeny Andriyash and Jason Rolfe and Bohdan Kulchytskyy and Roger Melko, "Quantum Boltzmann Machine," arXiv:1601.02036 (2016).

[20] Marcello Benedetti, John Realpe-Gómez, Rupak Biswas, and Alejandro Perdomo-Ortiz, "Quantum-assisted learning of hardware-embedded probabilistic graphical models," Phys. Rev. X **7**, 041052 (2017).

[21] Marcello Benedetti, John Realpe-Gómez, and Alejandro Perdomo-Ortiz, "Quantum-assisted helmholtz machines: A quantum-classical deep learning framework for industrial datasets in near-term devices," arXiv:1708.09784 (2017).

[22] Peter Wittek and Christian Gogolin, "Quantum enhanced inference in markov logic networks," Scientific Reports **7** (2017).

[23] Song Cheng, Jing Chen, and Lei Wang, "Information perspective to probabilistic modeling: Boltzmann machines versus born machines," arXiv:1712.04144 (2017).

[24] Lev S Bishop, Sergey Bravyi, Andrew Cross, Jay M Gambetta, and John Smolin, "Quantum volume," (2017).

[25] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik, "Quantum autoencoders for efficient compression of quantum data," Quantum Sci. Technol. **2**, 045001 (2017).

[26] L Lamata, U Alvarez-Rodriguez, JD Martín-Guerrero, M Sanz, and E Solano, "Quantum autoencoders via quantum adders with genetic algorithms," arXiv preprint arXiv:1709.07409 (2017).

[27] Rui Li, Unai Alvarez-Rodriguez, Lucas Lamata, and Enrique Solano, "Approximate quantum adders with genetic algorithms: An ibm quantum experience," Quantum Measurements and Quantum Metrology **4**, 1–7 (2017).

[28] E Miles Stoudenmire and David J Schwab, "Supervised learning with quantum-inspired tensor networks," arXiv preprint arXiv:1605.05775 (2016).

[29] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang, "Unsupervised generative modeling using matrix product states," arXiv preprint arXiv:1709.01662 (2017).

[30] Ding Liu, Shi-Ju Ran, Peter Wittek, Cheng Peng, Raul Blázquez García, Gang Su, and Maciej Lewenstein, "Machine learning by two-dimensional hierarchical tensor networks: A quantum information theoretic perspective on deep architectures," arXiv preprint arXiv:1710.04833

(2017).
[31] Xun Gao, Zhengyu Zhang, and Luming Duan, "An efficient quantum algorithm for generative machine learning," arXiv preprint arXiv:1711.02038 (2017).
[32] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, "Quantum principal component analysis," Nature Physics **10**, 631–633 (2014).
[33] Iordanis Kerenidis and Anupam Prakash, "Quantum recommendation systems," arXiv preprint arXiv:1603.08675 (2016).
[34] Fernando G. S. L. Brandao, Amir Kalev, Tongyang Li, Cedric Yen-Yu Lin, Krysta M. Svore, and Xiaodi Wu, "Exponential quantum speed-ups for semidefinite programming with applications to quantum learning," arXiv:1710.02581 (2017).
[35] Maria Schuld, Mark Fingerhuth, and Francesco Petruccione, "Quantum machine learning with small-scale devices: Implementing a distance-based classifier with a quantum interference circuit," arXiv preprint arXiv:1703.10793 (2017).
[36] Guillaume Verdon, Michael Broughton, and Jacob Biamonte, "A quantum algorithm to train neural networks using low-depth circuits," arXiv:1712.05304 (2017).
[37] Ian Goodfellow Yoshua Bengio and Aaron Courville, "Deep learning," (2016), mIT Press.
[38] Solomon Kullback and Richard A Leibler, "On information and sufficiency," The annals of mathematical statistics **22**, 79–86 (1951).
[39] J Kennedy and R Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, Vol. 4 (IEEE, 1995) pp. 1942–1948.
[40] Yuhui Shi and Russell Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on* (IEEE, 1998) pp. 69–73.
[41] Anders Sørensen and Klaus Mølmer, "Quantum computation with ions in thermal motion," Phys. Rev. Lett. **82**, 1971–1974 (1999).
[42] Anders Sørensen and Klaus Mølmer, "Entanglement and quantum computation with ions in thermal motion," Phys. Rev. A **62**, 022311 (2000).
[43] Jan Benhelm, Gerhard Kirchmair, Christian F. Roos, and Rainer Blatt, "Towards fault-tolerant quantum computing with trapped ions," Nature Physics **4**, 463 EP – (2008).
[44] Dmitri Maslov and Yunseong Nam, "Use of global interactions in efficient quantum circuit constructions," New Journal of Physics (2017).
[45] David J. C. MacKay, *Information Theory, Inference & Learning Algorithms* (Cambridge University Press, New York, NY, USA, 2002).
[46] 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.
[47] 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 did in Appendix B. A comprehensive study is left for future work.
[48] Daniel M. Greenberger, Michael A. Horne, Abner Shimony, and Anton Zeilinger, "Bell's theorem without inequalities," American Journal of Physics **58**, 1131–1143 (1990), https://doi.org/10.1119/1.16243.
[49] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt, "14-qubit entanglement: Creation and coherence," Phys. Rev. Lett. **106**, 130506 (2011).
[50] Andrea Rocchetto, Scott Aaronson, Simone Severini, Gonzalo Carvacho, Davide Poderini, Iris Agresti, Marco Bentivegna, and Fabio Sciarrino, "Experimental learning of quantum states," arXiv:1712.00127 (2017).
[51] Asier Ozaeta and Peter L. McMahon, "Decoherence of up to 8-qubit entangled states in a 16-qubit superconducting quantum processor," arXiv:1712.07080 (2017).
[52] Thomas Monz, Philipp Schindler, Julio T. Barreiro, Michael Chwalla, Daniel Nigg, William A. Coish, Maximilian Harlander, Wolfgang Hänsel, Markus Hennrich, and Rainer Blatt, "14-qubit entanglement: Creation and coherence," Phys. Rev. Lett. **106**, 130506 (2011).
[53] Federico Ricci-Tersenghi, "The bethe approximation for solving the inverse ising problem: a comparison with other inference methods," Journal of Statistical Mechanics: Theory and Experiment **2012**, P08015 (2012).
[54] I. Mastromatteo, "On the typical properties of inverse problems in statistical mechanics," ArXiv e-prints (2013), arXiv:1311.0190 [cond-mat.stat-mech].
[55] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski, "A learning algorithm for boltzmann machines," Cognitive science **9**, 147–169 (1985).
[56] For the all-to-all connectivity we obtained 5000 readouts and for the star topology experiments we obtained 6000 readouts.
[57] J.R. Johansson, P.D. Nation, and Franco Nori, "Qutip 2: A python framework for the dynamics of open quantum systems," Computer Physics Communications **184**, 1234 – 1240 (2013).
[58] Lester James V. Miranda, "Pyswarms, a research-toolkit for particle swarm optimization in python," (2017), 10.5281/zenodo.986300.
[59] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas, "The earth mover's distance as a metric for image retrieval," International journal of computer vision **40**, 99–121 (2000).
[60] Elizaveta Levina and Peter Bickel, "The earth mover's distance is the mallows distance: Some insights from statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, Vol. 2 (IEEE, 2001) pp. 251–256.
[61] Ofir Pele and Michael Werman, "Fast and robust earth mover's distances," in *2009 IEEE 12th International Conference on Computer Vision* (IEEE, 2009) pp. 460–467.