

Adversarial quantum circuit learning for pure state approximation

Marcello Benedetti,^{1,2,*} Edward Grant,¹ Leonard Wossnig,¹ and Simone Severini¹

¹*Department of Computer Science, University College London, WC1E 6BT London, United Kingdom*

²*Cambridge Quantum Computing Limited, CB2 1UB Cambridge, United Kingdom*

(Dated: April 11, 2019)

Adversarial learning is one of the most successful approaches to modelling high-dimensional probability distributions from data. The quantum computing community has recently begun to generalize this idea and to look for potential applications. In this work, we derive an adversarial algorithm for the problem of approximating an unknown quantum pure state. Although this could be done on error-corrected quantum computers, the adversarial formulation enables us to execute the algorithm on near-term quantum computers. Two ansatz circuits are optimized in tandem: One tries to approximate the target state, the other tries to distinguish between target and approximated state. Supported by numerical simulations, we show that resilient backpropagation algorithms perform remarkably well in optimizing the two circuits. We use the bipartite entanglement entropy to design an efficient heuristic for the stopping criteria. Our approach may find application in quantum state tomography.

I. INTRODUCTION

In February 1988 Richard Feynman wrote on his blackboard: ‘What I cannot create, I do not understand’ [1]. Since then this powerful dictum has been reused and reinterpreted in the context of many fields throughout science. In the context of machine learning, this is the perfect motivation for developing generative models: Machines that can generate realistic synthetic examples of their environment are likely to ‘understand’ such an environment.

Generative models are algorithms trained to approximate the joint probability distribution of a set of variables, given a dataset of observations. Conceptually, the quantum generalization is straightforward; Quantum generative models are algorithms trained to approximate the wave function of a set of qubits, given a dataset of quantum states. From this definition, there seems to be a link with the field of quantum state tomography. Indeed, there exist proposals of generative models for tomography such as the quantum principal component analysis [2] and the quantum Boltzmann machine [3, 4]. Other machine learning approaches to tomography have been formulated using the different framework of probably approximately correct learning [5, 6].

Quantum generative models can also be used to model classical data. As an example, Born machines [7, 8] use the probabilistic interpretation of the quantum wave function to reproduce the statistics observed in the data. Identifying classical datasets that can be modelled better via quantum correlations is an interesting open question in itself [9].

One of the most successful approaches to generative models is that of adversarial algorithms in which a discriminator is trained to distinguish between real and generated samples, and a generator is trained to confuse the discriminator [10]. The intuition is that if a generator is able to confuse a perfect discriminator, then it can generate realistic synthetic examples. Recently, researchers have begun to generalize this idea to the quantum computing paradigm [11, 12] where the discriminator has to distinguish between two sources that output copies of different quantum states. The discrimination of quantum states is so important that it was among the first problems ever considered in the field of quantum information theory [13]. The novelty of adversarial algorithms is in using the discriminator’s performance to provide a learning signal for the generator.

But how do generative models stand with respect to state-of-the art variational algorithms commonly used on available quantum hardware? Work on the variational quantum eigensolver [14] shows that shallow quantum circuits can be used to extract properties of quantum systems, e.g. the electronic energy of molecules. Similarly, the work on quantum approximate optimization [15] shows that shallow quantum circuits can be used to obtain good approximate solutions to hard combinatorial problems, e.g. the max-cut. All these problems consists of finding the ground state of a well-defined, task-specific, Hamiltonian. However, in generative models the problem is somehow inverted. We ask the question: What is the Hamiltonian that could have generated the statistics observed in the data set? Although some work has been done in this direction [8, 16], there seems to be a disconnect between these variational algorithms and the optimization routines needed for generative models. Here is where the adversarial learning framework stands out, thanks to its flexibility and implicit modelling of the Hamiltonian.

* m.benedetti@cs.ucl.ac.uk

In this manuscript, we start from information theoretic arguments and derive an adversarial algorithm that learns to generate approximations to a target pure quantum state. We parametrize shallow generator and discriminator circuits similarly to other variational approaches, and demonstrate their application for the task of approximate quantum state tomography. One of the purposes of this method is to use the underlying hardware to its fullest extent, including for the computation of gradients necessary to iteratively update the circuits. Optimization is performed using an adaptive gradient descent method known as resilient backpropagation (Rprop) [17], which performs well when the cost landscape is characterized by large plateaus with small gradients, and only requires that the sign of the gradient can be ascertained. We provide a heuristic method to assess the learning, which can in turn be used to design a stopping criteria. Although our simulations are carried out in the context of noisy intermediate-scale quantum computers (NISQ) [18], we discuss long-term realizations of the adversarial algorithm on universal quantum computers.

II. METHOD

Consider the problem of generating a pure state ρ_g close to an unknown pure target state ρ_t , where closeness is measured with respect to some distance metric to be chosen. Hereby we use subscripts g and t to label ‘generated’ and ‘target’ states, respectively. The unknown target state is provided a finite number of times by a channel. If we were able to learn the state preparation procedure, then we could generate as many ‘copies’ as we want and use these in a subsequent application. We now describe a game between two players whose outcome is an approximate state preparation for the target state.

Borrowing language from the literature of adversarial machine learning, the two players are called the generator and the discriminator. The task of the generator is to prepare a quantum state and fool the other player into thinking that it is the true target state. Thus, the generator is a unitary transformation G applied to some known initial state, say $|0\rangle$, so that $\rho_g = G|0\rangle\langle 0|G^\dagger$. We will discuss the generator’s strategy later.

The discriminator has the task of distinguishing between the target state and the generated state, and label them correctly. Hence, the discriminator can be thought of as a positive operator-valued measurement (POVM) $\{D_b\}$ such that $\sum_b D_b = I$, followed by a decision rule. The latter is a function that determines which state was measured depending on the measurement outcome. The discriminator is presented with the mixture $\rho_{mix} = P(t)\rho_t + P(g)\rho_g$, where $P(t)$ and $P(g)$ are prior probabilities summing up to one. Note that in practice the discriminator only sees one input at a time rather than the mixture of the density matrices, but we can treat the uncertainty in the input state using this picture. After performing the POVM on the system, we observe outcome b with probability $P(b) = \text{tr}[D_b \rho_{mix}]$. A straightforward application of Bayes’ theorem suggests that the decision rule should select the label for which the posterior probability is maximal, i.e., $\arg \max_{x \in \{g,t\}} P(x|b)$. This rule is called the Bayes’ decision function and is optimal in the sense that, given an optimal POVM, any other decision function will have a larger probability of a labeling error [19]. Recalling that $\max_{x \in \{g,t\}} P(x|b)$ is the probability of the correct decision using the Bayes decision function, we can formulate the probability of error as

$$\begin{aligned}
 P_{err}(\{D_b\}) &= \sum_b P(b) \left(1 - \max_{x \in \{g,t\}} P(x|b)\right) \\
 &= \sum_b P(b) \min_{x \in \{g,t\}} P(x|b) \\
 &= \sum_b \min_{x \in \{g,t\}} P(x|b) P(b) \\
 &= \sum_b \min_{x \in \{g,t\}} P(b|x) P(x) \\
 &= \sum_b \min_{x \in \{g,t\}} \text{tr}[D_b \rho_x] P(x).
 \end{aligned} \tag{1}$$

The choice of the POVM plays a key role here and the discriminator should consider finding the best possible one. Such optimal probability of error is the discriminator’s objective function and can be written in variational form as

$$P_{err}^* = \min_{\{D_b\}} P_{err}(\{D_b\}), \tag{2}$$

where the minimization is over all possible POVM elements, and the number of POVM elements is unconstrained.

It was Helstrom who carefully designed a POVM that achieves the smallest probability of error for a single sam-

ple [13]. He showed that the optimal discriminator comprises two elements, D_0 and D_1 , which are diagonal in a basis that diagonalizes $\Gamma = P(t)\rho_t - P(g)\rho_g$. When the outcome 0 is observed, the state is labeled as ‘target’, when the outcome 1 is observed the state is labeled as ‘generated’. This is the discriminator’s optimal strategy as it minimizes the probability of error. Unfortunately, from the definition we see that designing such a measurement would require knowledge of the target state beforehand, which contradicts the purpose of the game at hand.

Returning to the variational definition of the probability of error, and using Eqs. (1) and (2), assuming that $P(1|g), P(0|t)$ fulfill the Bayes’ decision rule [19], the probability of error can be written as

$$\begin{aligned} P_{err}^* &= \min_{\{D_0, D_1\}} \left(P(1|t)P(t) + P(0|g)P(g) \right) \\ &= \min_{\{D_0, D_1\}} \left(\text{tr}[D_1\rho_t]P(t) + \text{tr}[D_0\rho_g]P(g) \right) \\ &= \min_{D_0} \left(\text{tr}[(I - D_0)\rho_t]P(t) + \text{tr}[D_0\rho_g]P(g) \right) \\ &= P(t) + \min_{D_0} \left(-\text{tr}[D_0\rho_t]P(t) + \text{tr}[D_0\rho_g]P(g) \right). \end{aligned} \quad (3)$$

where we used $D_1 = I - D_0$ from the definition of POVM.

In contrast, the generator’s strategy is as follows: Assuming the discriminator be optimal, the generator achieves success by maximizing the probability of the error above with respect to the generated state ρ_g . The result is a zero-sum game similar to that of generative adversarial networks [10]

$$\begin{aligned} &\max_{\rho_g} \min_{D_0} \left(-\text{tr}[D_0\rho_t]P(t) + \text{tr}[D_0\rho_g]P(g) \right) \\ &= \min_{\rho_g} \max_{D_0} \left(\text{tr}[D_0\rho_t]P(t) - \text{tr}[D_0\rho_g]P(g) \right). \end{aligned} \quad (4)$$

Now suppose that the game is carried out in turns. On the one hand, the discriminator is after an unknown Helstrom measurement which changes over time as the generator plays her moves. On the other hand, the generator tries to imitate an unknown target state and it can only exploit the signal provided by the discriminator.

Note that when $P(t) = P(g) = \frac{1}{2}$, the probability of error in Eq. (2) is related to the trace distance between quantum states [20]

$$D(\rho_t, \rho_g) = \frac{1}{2} \|\rho_t - \rho_g\| = \max_{\{D_b\}} \frac{1}{2} \sum_b |\text{tr}[D_b(\rho_t - \rho_g)]|. \quad (5)$$

This is clearer form the variational definition on the right hand side. Hence, by playing the minimax game above with equal prior probabilities, we are implicitly minimizing the trace distance between target and generated state. We will use the trace distance to demonstrate learning in our experiments. In practice though, one does not have access to the maximum over all POVM in Eq. (5), because that would require, once again, the Helstrom measurement. We discuss this ideal scenario in a later Section where we require the availability of an error-corrected quantum computer. We shall now consider the case of implementation in NISQ computers where, due to the infeasibility of computing Eq. (5), we need to design a heuristic for the stopping criteria.

Finally, we shall mention that this game based on the Bayesian probability of error assumes the availability of one copy of ρ_{mix} at each turn. A more general minimax game could be designed based on the quantum Chernoff bound assuming the availability of multiple copies at each turn [19, 21].

A. Near-term implementation on NISQ computers

We now discuss how the game could be played in practice using shallow circuits on quantum computers without error correction. First, we assume the ability to efficiently provide the unknown target state as an input. In realistic scenarios, the target state would come from an external channel and would be loaded in the quantum computer’s register with no significant overhead. The source may be a quantum sensor or another quantum computer, while the channel may be a quantum internet.

Second, the generator’s unitary transformation would be implemented by a parametrized ansatz quantum circuit applied to a known initial state. Note that target and generated states have the same number of qubits and they are never input together, but rather as a mixture with probabilities $P(t)$ and $P(g)$, respectively, i.e., randomly selected

with a certain prior probability. Hence they can be prepared in the same quantum register.

Third, resorting to Neumark's dilation theorem [22], the discriminator's POVM shall be realized as a unitary transformation followed by a projective measurement on an extended system. Such extended system consists of the quantum register shared by the target and generated states, plus an ancilla register initialized to a known state. The number of basis states for the ancillary system needs to match the number of outcomes of the POVM, which is hence given by a power of two. Because we are here specifically looking at a rank-two POVM, the extended system consists of just one ancilla qubit. The unitary transformation on the extended system is then implemented by another parametrized ansatz quantum circuit. The measurement is described by projectors on the state space of the ancilla where outcomes 0 and 1 are associated to the labels 'target' and 'generated', respectively.

Depending on the characteristics of the two ansatz circuits, such as type of gates, depth and connectivity, we will be able to explore parts of the Hilbert space with the generator, and explore parts of the cone of positive operators with the discriminator. Without loss of generality, assume that the n -qubit target state is prepared by an unknown unitary $T \otimes I$ acting on $|0\rangle^{\otimes n+1}$. Here T acts non-trivially on a n -qubit register, and leaves the ancilla register untouched. We also construct a generator circuit $G(\boldsymbol{\theta}) \otimes I = G_L(\theta_L) \cdots G_1(\theta_1) \otimes I$ where each gate $G_l(\theta_l) = \exp(-i\theta_l \hat{\sigma}_l/2)$ is associated to an angle θ_l and a tensor product of Pauli matrices $\hat{\sigma}_l = \otimes_{i=1}^n \sigma_l^i$ where $\sigma_l^i \in \{\sigma_x, \sigma_y, \sigma_z, I\}$. To summarize the target unitary, denoted as target in the following, prepares the state $\rho_t = T|0\rangle\langle 0|T^\dagger \otimes |0\rangle\langle 0|$ and the generator unitary, denoted as generator in the following, prepares the state $\rho_g = G(\boldsymbol{\theta})|0\rangle\langle 0|G(\boldsymbol{\theta})^\dagger \otimes |0\rangle\langle 0|$.

Now, we construct a discriminator circuit $D(\boldsymbol{\phi}) = D_L(\phi_L) \cdots D_1(\phi_1)$ acting non-trivially on both registers, and append a measurement of the ancilla qubit using projectors $P_b = I^{\otimes n} \otimes |b\rangle\langle b|$ with $b \in \{0, 1\}$. Using these circuits, the minimax game in Eq. (4) can be written as $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\phi}} V(\boldsymbol{\theta}, \boldsymbol{\phi})$ with value function

$$V(\boldsymbol{\theta}, \boldsymbol{\phi}) = \text{tr} \left[P_0 D(\boldsymbol{\phi}) \left(T|0\rangle\langle 0|T^\dagger \otimes |0\rangle\langle 0| \right) D(\boldsymbol{\phi})^\dagger \right] P(t) - \text{tr} \left[P_0 D(\boldsymbol{\phi}) \left(G(\boldsymbol{\theta})|0\rangle\langle 0|G(\boldsymbol{\theta})^\dagger \otimes |0\rangle\langle 0| \right) D(\boldsymbol{\phi})^\dagger \right] P(g). \quad (6)$$

Each player optimizes this value function in turn. This optimization can in principle be done via different approaches (e.g. gradient free, first-, second-order methods, etc.) depending on the computational resources available.

Here we discuss a simple method of alternated optimization by gradient descent/ascent starting from randomly initialized parameters $\boldsymbol{\theta}^{(0)}$ and $\boldsymbol{\phi}^{(0)}$. That is, we perform iterations of the form $\boldsymbol{\theta}^{(t+1)} = \arg \min_{\boldsymbol{\theta}} V(\boldsymbol{\theta}, \boldsymbol{\phi}^{(t)})$ and $\boldsymbol{\phi}^{(t+1)} = \arg \max_{\boldsymbol{\phi}} V(\boldsymbol{\theta}^{(t+1)}, \boldsymbol{\phi})$ where we start with an arbitrary order.

Using the method proposed in Ref. [23], the analytical gradients for the generator are computed as

$$\frac{\partial V}{\partial \theta_l} = -\frac{P(g)}{2} \left\{ \text{tr} \left[P_0 D \left(G_{l+} |0\rangle\langle 0| G_{l+}^\dagger \otimes |0\rangle\langle 0| \right) D^\dagger \right] - \text{tr} \left[P_0 D \left(G_{l-} |0\rangle\langle 0| G_{l-}^\dagger \otimes |0\rangle\langle 0| \right) D^\dagger \right] \right\}, \quad (7)$$

where we omit the dependency on $\boldsymbol{\theta}^{(t)}$ and $\boldsymbol{\phi}^{(t)}$ to simplify notation, and where

$$G_{l\pm} = G_L(\theta_L) \cdots G_{l+1}(\theta_{l+1}) G_l(\theta_l \pm \pi/2) G_{l-1}(\theta_{l-1}) \cdots G_1(\theta_1). \quad (8)$$

Note that $G_{l\pm}$ can be interpreted as two new circuits, each one differing from G by an offset of $\pm \frac{\pi}{2}$ to one angle of one of the rotations. More precisely, for each parameter l , we are required to execute circuits $DG_{l\pm}$ on initial state $|0\rangle^{\otimes n+1}$ and estimate the expectation of observable P_0 from samples. Because these auxiliary circuits have the same depth as the original shallow circuit, computation of the gradient is efficient. Interestingly, up to a scale factor of $\frac{4}{\pi}$, the analytical gradient is equal to the central difference approximation carried out at $\frac{\pi}{2}$.

The analytical gradient for the discriminator has similar structure

$$\frac{\partial V}{\partial \phi_l} = \frac{P(t)}{2} \left\{ \text{tr} \left[P_0 D_{l+} \left(T|0\rangle\langle 0|T^\dagger \otimes |0\rangle\langle 0| \right) D_{l+}^\dagger \right] - \text{tr} \left[P_0 D_{l-} \left(T|0\rangle\langle 0|T^\dagger \otimes |0\rangle\langle 0| \right) D_{l-}^\dagger \right] \right\} - \frac{P(g)}{2} \left\{ \text{tr} \left[P_0 D_{l+} \left(G|0\rangle\langle 0|G^\dagger \otimes |0\rangle\langle 0| \right) D_{l+}^\dagger \right] - \text{tr} \left[P_0 D_{l-} \left(G|0\rangle\langle 0|G^\dagger \otimes |0\rangle\langle 0| \right) D_{l-}^\dagger \right] \right\}, \quad (9)$$

where again we omit the dependencies on angles to simplify notation, and where

$$D_{l\pm} = D_L(\phi_L) \cdots D_{l+1}(\phi_{l+1}) D_l(\phi_l \pm \pi/2) D_{l-1}(\phi_{l-1}) \cdots D_1(\phi_1). \quad (10)$$

In this case, for each parameter l we are required to execute four auxiliary circuits, $D_{l\pm}T$ and $D_{l\pm}G$, on initial state

$|0\rangle^{\otimes n+1}$ and estimate the expected value of observable P_0 from samples.

Finally, all angles are updated by gradient descent/ascent

$$\theta_l^{(t+1)} = \theta_l^{(t)} - \epsilon \frac{\partial V}{\partial \theta_l} \Big|_{\theta=\theta^{(t)}, \phi=\phi^{(t)}}, \quad (11)$$

$$\phi_l^{(t+1)} = \phi_l^{(t)} + \eta \frac{\partial V}{\partial \phi_l} \Big|_{\theta=\theta^{(t+1)}, \phi=\phi^{(t)}}, \quad (12)$$

where ϵ and η are hyperparameters determining the step sizes. In principle, we rely on the fine-tuning of these, as opposed to Newton's method which makes use of the Hessian matrix to determine step sizes for all parameters. Other researchers [11] designed circuits to estimate the analytical gradient and the Hessian matrix. Such approach requires the ability to execute complex controlled operations and is expected to require an error-corrected quantum computer. Our approach and others' [23, 24] require much simpler circuits, which is desirable for implementation on NISQ computers. As we discuss next, accelerated gradient techniques developed by the deep learning community could significantly improve the method used here.

B. Optimization by resilient backpropagation

In general, the problem of optimizing over unitaries is convex. However, in this paper we deal with a non-convex optimization problem due to the optimization over the angles of rotation instead, and hence the introduction of sine and cosine functions.

A recent paper [25] proposed that the error surface of circuit learning problems are challenging for gradient-based methods due to observation that these exhibit barren plateaus. The region where the gradient is close to zero does in particular not correspond to local minima of interest, but instead to exponentially large plateaus of states that have exponentially small deviations in the objective value from the average of the totally mixed state. The derivation of the above statement is for a class of random circuits, but in the general case we deal with highly structured unitary circuits [26, 27]. Here we argue that the existence of plateaus does not necessarily pose a problem for the learning of quantum circuits, provided that the sign of the gradient can be resolved, which will become clear from the following discussion. To validate this claim we refer to the classical literature and argue that similar problems have traditionally also occurred in classical neural network training and allow for efficient solutions.

Typical gradient based methods take steps of the form

$$w_i^{(t+1)} = w_i^{(t)} - \epsilon \frac{\partial}{\partial w_i} E^{(t)}, \quad (13)$$

where $w_i^{(t)}$ is the i -th weight at time t , ϵ is the learning rate, $E^{(t)}$ is the error function to be minimized and its superscript indicates evaluation at $w = w^{(t)}$. If the learning rate is too small, the derivatives are also scaled to be too small, resulting in a longer time to convergence. On the other hand, too large learning rates typically can lead to an oscillation of the error or even divergence. One of the early approaches to counter this behavior was the introduction of a momentum term, which takes into account the previous step in the update. Letting $\Delta w_i^{(t)} = w_i^{(t+1)} - w_i^{(t)}$, the update rule reads

$$w_i^{(t+1)} = w_i^{(t)} - \epsilon \frac{\partial}{\partial w_i} E^{(t)} + \mu \Delta w_i^{(t-1)}, \quad (14)$$

where μ is a momentum hyperparameter. Momentum based methods produce some resilience to plateaus in the cost landscape. However, they lose this resilience when the plateaus are characterized by having very small or zero gradients.

A family of optimizers known as resilient backpropagation (Rprop) algorithms [17] are particularly well suited for problems where the cost landscape is characterized by large plateaus with small gradients. Rprop type algorithms independently adapt the step size for each weight based on agreement between the sign of the current and previous gradients for that weight. If the sign of the current and previous gradient agree then the step size for that weight is increased multiplicatively. This allows the optimizer to traverse large areas with small gradient with an increasingly high speed. If the signs disagree, then the step size is decreased multiplicatively. The direction of the gradient descent step for each weight is then taken with a negative sign. For this reason Rprop is resilient to gradients with very small magnitude provided that the sign of the gradients can be determined.

We use a variant known as iRprop⁻ [28] which differs from the original Rprop algorithm by setting the current gradient to zero when the signs of the current and previous gradients disagree. This means that the weight is not

updated and the step size is kept constant for the next update, resulting in a repetition of the previous step with reduced step size. The pseudocode for iRprop^- is given in Algorithm 1.

Despite the resilience of Rprop to small gradients, if the magnitude of the gradient in a given direction is so small that the sign cannot be determined then Rprop will not take a step in that direction. Furthermore, noise derived from a limited number of measurements could cause the sign of the estimated gradient to flip between each step. This would quickly make the step size very small and the optimizer could become stuck on a barren plateau.

One possible modification is an explorative version of Rprop that explores areas with zero or very small gradients at the beginning of training, but still converges during the end of training. An explorative Rprop could remember the sign of the last suitably large gradient and take a step in the same direction. When the optimizer encounters a plateau it would traverse the plateau from the same direction it entered. The step size adjustment for the very small or zero gradient case could be > 1 during the beginning of training then decreased towards the end so that smaller steps would be taken to allow for convergence to a minima. Any zero or very small gradients at the first iteration could be replaced by positive gradients to ensure an initial direction is always defined. We leave investigation of an explorative Rprop algorithm to future work.

Algorithm 1 iRprop^- [28]

Require: step size increase factor η^+ , step size decrease factor η^- , minimum allowed step size Δ_{min} , and maximum allowed step size Δ_{max}

- 1: **for each** w_i **do**
 - 2: **if** $\frac{\partial}{\partial w_i} E^{(t-1)} \frac{\partial}{\partial w_i} E^{(t)} > 0$ **then**
 - 3: $\Delta_i^{(t)} := \min\{\eta^+ \Delta_i^{(t-1)}, \Delta_{max}\}$
 - 4: **if** $\frac{\partial}{\partial w_i} E^{(t-1)} \frac{\partial}{\partial w_i} E^{(t)} < 0$ **then**
 - 5: $\Delta_i^{(t)} := \max\{\eta^- \Delta_i^{(t-1)}, \Delta_{min}\}$
 - 6: $\frac{\partial}{\partial w_i} E^{(t)} := 0$ \triangleright in standard Rprop this step is omitted
 - 7: $w_i^{(t+1)} := w_i^{(t)} - \text{sgn}(\frac{\partial}{\partial w_i} E^{(t)}) \Delta_i^{(t)}$
-

C. Heuristic for the stopping criteria

Evaluating the performance of implicit generative models is often intractable. In these cases, assessment is done via application-dependent heuristics [29, 30]. Here we describe an efficient method that can be used to assess the learning in the quantum setting. In turn, this can be used to define a stopping criteria.

We first note that the value function in Eq. (6) does not provide information about the generator's performance, unless the discriminator is optimal. Unfortunately, we will not always have access to an optimal discriminator (more on this in the following Section). Second, the discriminator makes use of an ancilla register \mathcal{A} to encode its output. Should the ancilla be maximally entangled with the main register \mathcal{M} , its reduced density matrix would correspond to a maximally mixed state. Performing projective measurements on the maximally mixed state would then result in random decisions.

Ideally, the discriminator would encode its decision in the ancilla register and then remove all its correlations with the main register, obtaining a product state $\rho_d = \rho^{\mathcal{M}} \otimes \rho^{\mathcal{A}}$. Hereby we use subscript d to indicate the state output by the discriminator circuit. This scenario is similar in spirit to the uncomputation technique used in many quantum algorithms [31].

The bipartite entanglement entropy (BEE) quantifies how much entanglement there is between two partitions

$$S(\rho^{\mathcal{A}}) = \text{tr}[\rho^{\mathcal{A}} \ln \rho^{\mathcal{A}}] = \text{tr}[\rho^{\mathcal{M}} \ln \rho^{\mathcal{M}}] = S(\rho^{\mathcal{M}}), \quad (15)$$

where $\rho^{\mathcal{A}} = \text{tr}_{\mathcal{M}}[\rho_d]$ and $\rho^{\mathcal{M}} = \text{tr}_{\mathcal{A}}[\rho_d]$ are reduced density matrices obtained by tracing out one of the partitions. The BEE is intractable in general, but here we can exploit its symmetry and compute it on the smallest partition, i.e. the ancilla register \mathcal{A} . Because this register consists of a single qubit, BEE reduces to

$$S(\rho^{\mathcal{A}}) = -\frac{1 + \|\mathbf{r}\|}{2} \ln \left(\frac{1 + \|\mathbf{r}\|}{2} \right) - \frac{1 - \|\mathbf{r}\|}{2} \ln \left(\frac{1 - \|\mathbf{r}\|}{2} \right), \quad (16)$$

where $\|\mathbf{r}\| \leq 1$ and $\mathbf{r} \in \mathbb{R}^3$ is the Bloch vector such that $\rho^A = \frac{1}{2}(I + \boldsymbol{\sigma} \cdot \mathbf{r})$. Estimating the three components of this vector requires tomography of a single-qubit, for which we refer to the excellent review in Ref. [32].

Depending on the desired accuracy, prior knowledge, and available computational resources, there exist a whole spectrum of methods. In this work we consider the scaled direct inversion (SDI) [32] method, where we estimate each entry of the Bloch vector independently by measuring the corresponding Pauli operator. This is motivated by the fact that $\langle \sigma_i \rangle = \text{tr}[\sigma_i \rho^A] = \mathbf{e}_i \cdot \mathbf{r}$ where \mathbf{e}_i is the Cartesian unit vector in the i direction and $i \in \{x, y, z\}$. This type of measurement can be done in all NISQ computers we are aware of by applying the corresponding rotation gate followed by a projective measurement in the computational basis.

We can write a temporary Bloch vector $\hat{\mathbf{r}}_0 = (\langle \sigma_x \rangle, \langle \sigma_y \rangle, \langle \sigma_z \rangle)$ where all expectations are estimated from samples. Due to finite sampling error, there is non-zero probability that the vector lies outside the unite sphere, although inside the unit cube. These cases correspond to non-physical states and SDI corrects them by finding the valid state with minimum distance over all Schatten p-distances. It turns out, this is simply the rescaled vector

$$\hat{\mathbf{r}} = \begin{cases} \hat{\mathbf{r}}_0 & \text{if } \|\hat{\mathbf{r}}_0\| \leq 1 \\ \hat{\mathbf{r}}_0 / \|\hat{\mathbf{r}}_0\| & \text{otherwise.} \end{cases} \quad (17)$$

The procedure discussed so far allows us to efficiently estimate the BEE in Eq. (16). Equipped with this information, we can now design an heuristic for the stopping criteria. The reasoning is as follows. Provided that the discriminator circuit has enough connectivity, random initialization of its parameters will likely generate high entanglement between the main and ancilla registers. That is, $S(\rho^A)$ is large at the beginning. As the learning algorithm iterates, the discriminator gets better which requires $S(\rho^A)$ to decrease as discussed above. This is when the learning signal for the generator is stronger, allowing the generated state to get closer to the target. As the two become less and less distinguishable with enough iterations, the discriminator needs to increase correlations between ancilla's bases and relevant factors in the main register. That is, we expect to observe an increase of entanglement, hence an increase in BEE. The performance of the discriminator would then saturate as $S(\rho^A)$ converges to its upper bound of $\ln(2)$. We propose to detect this convergence and use it as a stopping criteria. In the Section III we analyze the behavior of BEE via numerical simulations.

D. Long-term implementation on error-corrected quantum computers

Let us briefly recall the task we try to optimize the circuit for. We have two circuits, the generator, and the discriminator. The generator provides the state ρ_t with probability $P(t)$, while the target state ρ_g is provide with probability $P(g)$. The discriminator has to successfully distinguish each state or, in other words, he must find the measurement that minimizes the probability of a labelling error.

As described earlier, Helstrom [13] made the observation that the optimal POVM that distinguishes two states, i.e., the Helstrom minimal-error probability measurement, possesses the following particular form. Let D_0 and D_1 be the elements of the measurement attaining the minimum in $P_{err}^* = \min_{\{D_0, D_1\}} P(g)\text{tr}[\rho_g D_0] + P(t)\text{tr}[\rho_t D_1]$. Then both optimal elements are diagonal in a basis that also diagonalizes the Hermitian operator

$$\Gamma = P(t)\rho_t - P(g)\rho_g. \quad (18)$$

As pointed out in Ref. [19], in this basis one can replace the diagonal elements γ_j of Γ with the diagonal elements of D_0 by the rule

$$\begin{aligned} \lambda_j &= 1 & \text{when } \gamma_j < 0 \\ \lambda_j &= 0 & \text{when } \gamma_j \geq 0. \end{aligned}$$

The operator D_1 is then obtained via the relationship $I - D_0$. Hence we see that we can construct the optimal measurement operator if we have access to the operator Γ and its diagonal elements. Note that if we measure a 1 in the optimal measurement POVM, then we obtain a negative value for the new operator Γ , and that if we 0, then we obtain some positive value for Γ .

Using the above insight, let $\rho_t = |\psi_t\rangle\langle\psi_t|$, $\rho_g = |\psi_g\rangle\langle\psi_g|$, we can observe that $\text{tr}[\Gamma\rho_g] = P(t)|\langle\psi_g|\psi_t\rangle|^2 - P(g)$ and $\text{tr}[\Gamma\rho_t] = P(t) - P(g)|\langle\psi_g|\psi_t\rangle|^2$. Under the assumption of equal prior probabilities of $1/2$, the above is minimized for a maximum overlap of the two states. Since the prior probabilities are hyperparameters we can set, we can use the swap test [33] to effectively implement an optimal discriminator.

Note, however, that the swap test bears several disadvantages. First, it requires $2n+1$ qubits, which is significantly more than what required in the adversarial approach presented here. Second, in order to perform the swap test, we

need to access both ρ_t and ρ_g simultaneously. Finally, it requires the ability to perform controlled gates under an error-corrected device.

A potential solution is to find an efficient low-depth implementation of the shallow circuit to implement the swap test. In Ref. [34] the authors implemented such via a variationally trained circuit. However, as they point out in their work, this requires (a) an order of 2^{2n} training data points for states of n qubits, and (b) each training example is given by the actual overlap between two states, requiring a circuit which gives the answer to the problem we try to solve. We hence hold the belief that this approach is not suitable for our task. Other potential approaches for finding a lower depth circuit for computing the swap test however might well be possible.

One could alternatively consider the possibility of implementing a discriminator via distance measurements based on random projections, i.e., Johnson-Lindenstrauss transformations [35]. This would require a reduced amount of resources and could be adapted for the above discussed task. For example, we could apply a quantum channel to coherently reduce the dimensionality of the input state, and then apply the standard state discrimination procedure to distinguish between the two cases in a lower dimensional space. However, in Ref. [36] the authors proved that such an operation can not be performed by a quantum channel. One way to think about this is that the Johnson-Lindenstrauss transformation is a projection onto a small random subspace and therefore a projective measurement. Since the subspace is exponentially small with respect to the initial Hilbert space, the probability that this projection succeeds will be very small.

III. RESULTS

In realistic scenarios, the target state would come from an external channel and would be loaded in the quantum computer's register with no significant overhead. For the simulation we mock this scenario using a circuit as well. That is, we have $\rho_t = T(\gamma)|0\rangle\langle 0|T(\gamma)^\dagger$ where all angles are initialized uniformly at random in $\{-\frac{\pi}{2}, +\frac{\pi}{2}\}$. Similarly, we setup a generator circuit $G(\theta)$ and a discriminator circuit $D(\phi)$, as shown in Fig. 1 on the left panel. We shall stress that neither the generator nor the discriminator are allowed to 'see' the angles γ at any time.

The gates comprising those circuits should be placed in a layout compatible with the underlying quantum hardware. For our simulations we use a 'building block' consisting of arbitrary two-qubit gates $U_{i,i+1}$ for all odd qubit indexes i , followed by arbitrary two-qubit gates $U_{j,j+1}$ for all even qubit indexes j . That is, each building block has $n-1$ arbitrary two-qubit gates where n is the number of qubits. Under the hood, each arbitrary two-qubit gate is made of three CNOT gates and 18 single-qubit rotations. We also vary the number of repetitions of this building block, and we identify that number with a complexity hyperparameter $c(\cdot)$. Therefore, a circuit K acting on n qubits will have $18(n-1)c(K)$ parameters to be learned.

Unless stated otherwise, optimization was performed using iRprop⁻. We used an initial step sizes $\Delta_i^{(0)} = 1.5\pi \times 10^{-3}$, a minimum step size $\Delta_{min} = \pi \times 10^{-6}$, and a maximum step size $\Delta_{max} = 6\pi \times 10^{-3}$. All expected values required for learning were estimated using 100 measurements. Finally, we chose $P(t) = P(g) = \frac{1}{2}$ so that the discriminator is given in input target and generated states with equal probability.

Figure 2 shows learning curves for simulations on four qubits, where lines represent mean and one standard deviation computed on 10 repetitions. The closeness of target and generated state is measured in terms of trace distance (green downward triangles). In the left panel, complexities $c(T) = c(G) = 2$ and $c(D) = 1$. That is, the complexity of the discriminator is not sufficient to provide a learning signal for the generator, and the final approximation is indeed not satisfactory. In the central panel, $c(T) = c(D) = 2$ and $c(G) = 1$. The generator is less complex than the target state, but it manages to produce a meaningful approximation in average. In the right panel, $c(T) = c(G) = c(D) = 2$. The complexity of all parties is optimal, and the generator learns an indistinguishable approximation of the target state.

Although the trace distance between pure states could be approximated using a swap test, it is assumed that we do not have access to an error-corrected quantum computer. In Section II we designed a heuristic to keep track of learning and suggested to use it as a stopping criteria. In our simulations, we estimated the required Bloch vector \mathbf{r} of the ancilla qubit using scaled direct inversion and 100 measurements for each observable σ^x , σ^y , and σ^z . Figure 2 shows mean and one standard deviation of the BEE (blue upwards triangles) computed on 10 repetitions. The left panel shows that, when the discriminator circuit is too shallow, the BEE oscillates with no clear pattern. The central and right panels shows that, when using a favorable setting instead, the initial BEE drops significantly towards zero. This is when the generator begins to approximate the target state. Note that in all cases, the ancilla qubit converges to the maximally mixed state where $S(\rho_d^A) = \ln 2 = 0.69$ (gray horizontal line). In this regime, the discriminator predicts the labels with probability equal to the prior $P(t) = P(g) = \frac{1}{2}$.

Detecting convergence of BEE can be used as a stopping criteria for training. For example, the experiments shown in the central and right panels of Fig. 2 could have been stopped at iteration 150 obtaining excellent results in average. We now show details for two cases. First, we examine the case where the generator is under-parametrized. Figure 3, right panel, shows the absolute value of the entries of the density matrix for one of the four-qubit target states. The

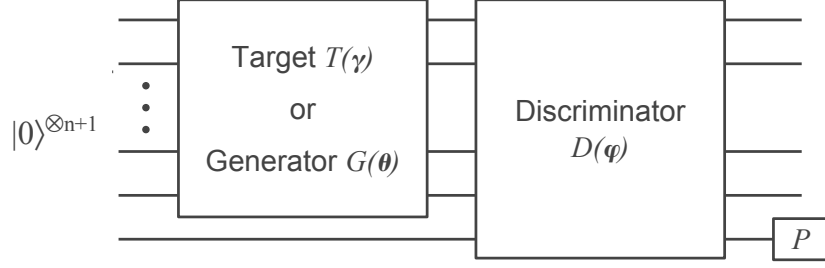


FIG. 1. The setting used in our simulations. The generator unitary $G(\theta)$ aims at preparing a state indistinguishable from that prepared by the target unitary $T(\gamma)$. A discriminator $D(\phi)$ takes in input whatever state has been prepared and classifies it as ‘target’ or ‘generated’ aiming at the smallest probability of error. The classification is done via the binary outcome of a projective measurement P on a single ancilla qubit. The minimax game is set up with $T(\gamma)$ fixed throughout and $G(\theta)$ and $D(\phi)$ updating their parameters in turns. Note that generator never ‘sees’ the parameters γ of the target, hence its learning signal comes solely from the probability of error of the discriminator.

randomly initialized generator produced the state shown in the left panel which has trace distance 0.991 to the target. By stopping the adversarial algorithm after 150 iterations, we generated the state shown in the central panel which has trace distance of 0.6. The generator managed to capture the main mode of the density matrix, that is the sharp peak on the right. Second, we examine the case where the generator is sufficiently parametrized. Figure 4, right panel, shows the absolute value of the entries of the density matrix for the target state. The generator initially produced the state shown in the left panel which has trace distance 0.951 to the target. By stopping the adversarial algorithm after 150 iterations, we generated the state shown in the central panel which has trace distance of 0.121. Visually, the target and final states are indistinguishable.

But how do the complexities of generator and discriminator affect the outcome? To verify this, we run the adversarial learning on six-qubit target states of complexity $c(T) = 3$, and varied the complexities of generator and discriminator. After 600 training iterations, we computed the mean trace distance across five repetitions. As illustrated in Fig. 5, more complex generator and discriminator circuits resulted in a generator circuit that better approximates the target state. From the complexities tested here, that is 2, 3 and 4, we conclude that a deeper discriminator results in superior performance compared to a deeper generator.

In our final experiment, we compared optimization algorithms on six-qubit target states. We ran gradient descend with momentum (GDM) and iRprop⁻ for 600 iterations. Figure 6 shows mean and one standard deviation across five repetitions. iRprop⁻ (blue downward triangles) outperformed both GDM with learning rate $\epsilon = 0.01$ (green circles) and with $\epsilon = 0.001$ (red upward triangles). This is because despite the small magnitude of the gradients when considering problems of six qubits, we were still able to estimate their sign and take relevant steps in the correct direction. This is a significant advantage of resilient backpropagation algorithms.

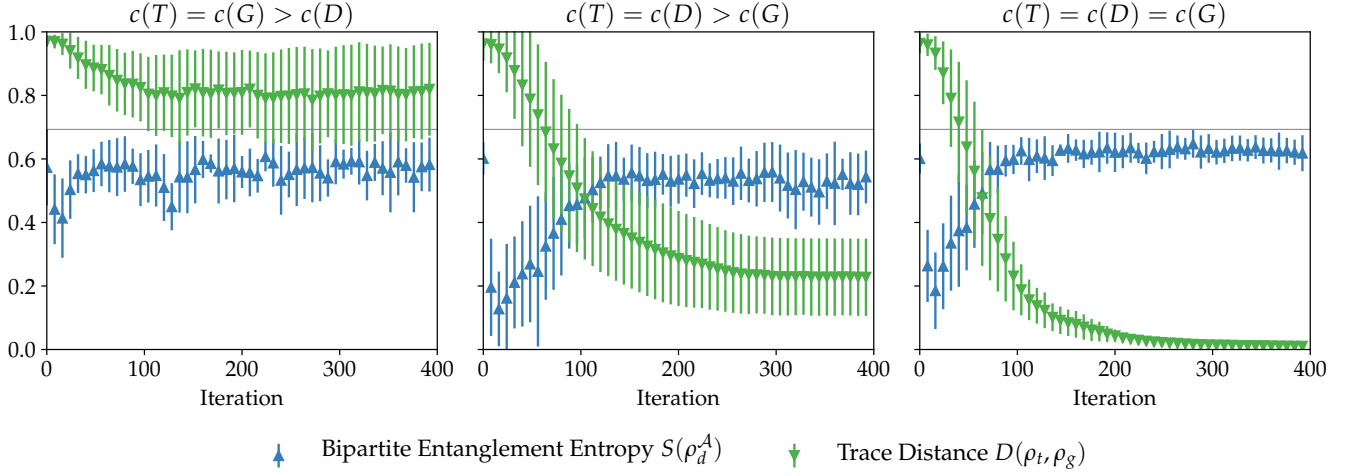


FIG. 2. Algorithm performance and heuristic for the stopping criteria in four-qubit experiments. All lines represent mean and one standard deviation computed on 10 repetitions. The performance is shown in terms of the trace distance between the target and generated state (green downward triangles). In the left panel, the complexity of the discriminator is not sufficient, hence there is no learning signal for the generator. In the central panel, the generator is less complex than the target state, hence it finds a meaningful approximation. In the right panel, the complexity of all circuits is sufficient to learn an indistinguishable approximation of the target state. The bipartite entanglement entropy (BEE) of the ancilla register (blue upward triangles) can be used as an efficient proxy to assess the progress of learning. We estimated this quantity from measurements on the ancilla qubit (see main text for the details). After the initial drop in BEE, the learning signal for the generator is large, hence the trace distance decreases sharply. As learning progresses, the ancilla qubit goes back to a mixed state and eventually converges to maximally mixed, i.e., $S(\rho_d^A) = \ln 2 = 0.69$ (gray horizontal line). Detecting such pattern can be used as a stopping criteria for training.

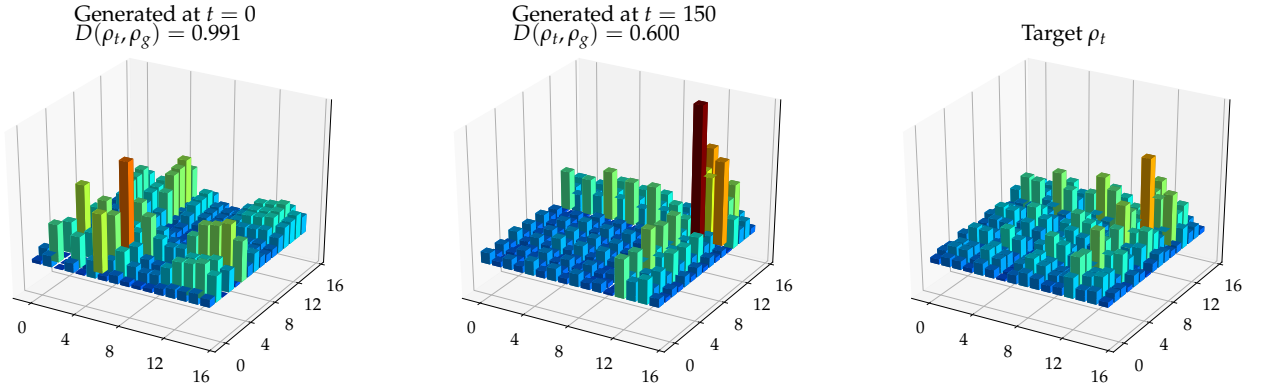


FIG. 3. Absolute value of tomographic reconstructions of a random four-qubit target state. The target is prepared by a circuit of complexity $c(T) = 2$ (see definition in the main text) and the absolute value of its density matrix is shown in the right panel. The two players of the game are a generator with $c(G) = 1$ and a discriminator with $c(D) = 2$. This means that the generator is not able to learn the target exactly, but can find a reasonable approximation. The initial generated state shown in the left panel has trace distance 0.991 to the target. After 150 iterations of adversarial learning, the generated state shown in the central panel has trace distance 0.6. The generator managed to capture the main mode of the density matrix, that is the sharp peak on the right.

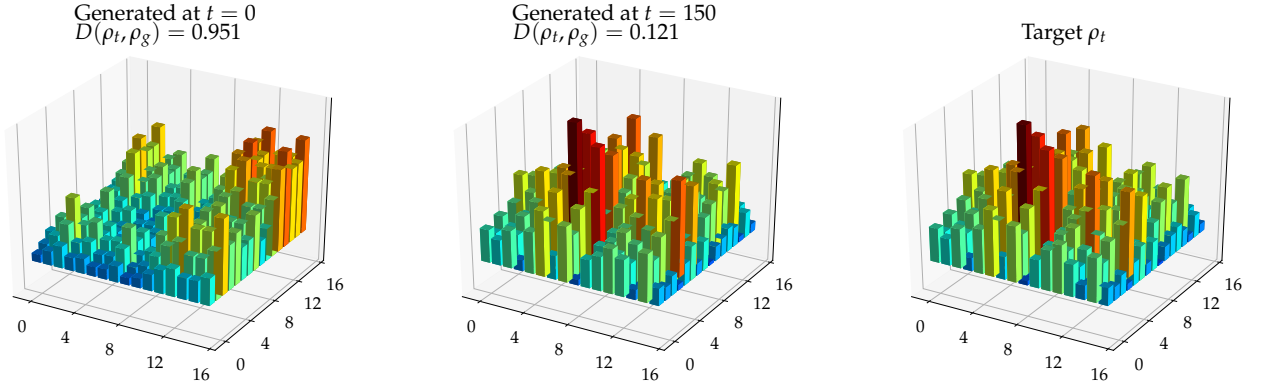


FIG. 4. Absolute values of tomographic reconstructions of a random four-qubit target state. The setting is similar to that of Fig. 3, but this time the generator is a circuit with complexity $c(G) = 2$, hence it can learn the target exactly. The randomly initialized generator produces the state shown in the left panel, which has trace distance 0.951 to the target. After 150 iterations of adversarial learning, the result is a generated state with trace distance 0.121, shown in the center. Visually, target and final states are indistinguishable.

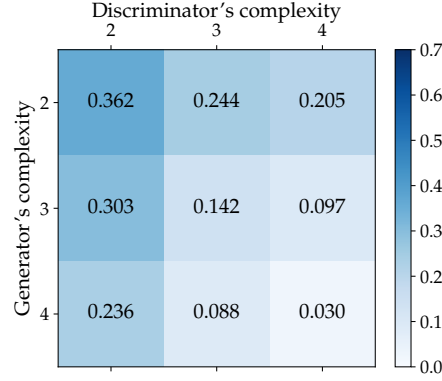


FIG. 5. Quality of the approximation against complexity of circuits for six-qubit experiments. The heat-map shows mean trace distance of five repetitions of adversarial learning computed at iteration 600. All standard deviations were < 0.1 (not shown). Increasing the complexity of discriminator and generator resulted in improved performance. From the complexities tested here, we conclude that a deeper discriminator results in superior performance than a deeper generator.

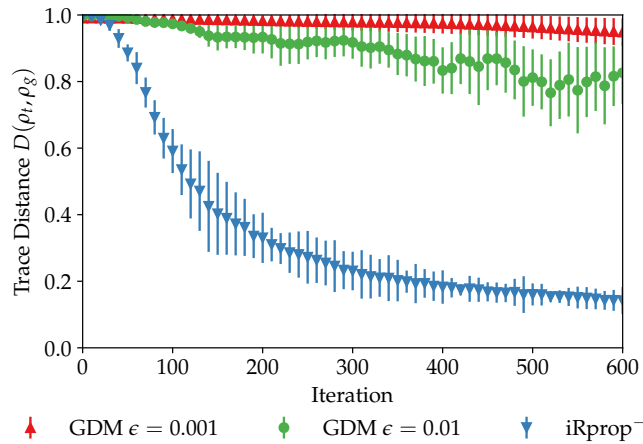


FIG. 6. Learning curves for different optimizers on six-qubit experiments. The lines represent mean and one standard deviation of the trace distance computed out of five repetitions. $iRprop^-$ gave better performance than gradient descent with momentum (GDM) with two different learning rates. Increasing the learning rate further in GDM resulted in unstable performance (not shown).

IV. DISCUSSION AND CONCLUSIONS

In this work we proposed an adversarial quantum learning algorithm and applied it to approximately learn circuits that can generate and distinguish pure states. We used information theoretic arguments to formalize the problem as a minimax game. The discriminator circuit maximizes the value function in order to better distinguish between the target and the generated state. This can be thought of as trying to perform the Helstrom measurement [13]. In turn, the generator circuit minimizes the value function in order to deceive the discriminator. This can be thought of as minimizing the trace distance to the target state. The desired outcome is to obtain the best possible approximation to the target state, under the constraints dictated by the generator circuit layout.

We demonstrated how to perform such a minimax game in near-term quantum devices, i.e., NISQ computers [18], as well as far future error-corrected quantum computers. The former implementation has the advantage that it requires less qubits and avoids the swap test. The latter implementation, on the other hand, has the advantage that it can make use of the actual Helstrom measurement, with the potential of speeding up the learning process.

Previous work on quantum circuit learning raised the concern of barren plateaus in the cost landscape [25]. We showed numerically that a class of optimizers called resilient backpropagation achieves high performance for the problem at hand, while gradient descent with momentum performs relatively poorly. These resilient optimizers require only the temporal behaviour of the sign of the gradient, and not the gradient's magnitude, to perform an update step. In our experiments on up to $6 + 1$ qubits we were able to correctly ascertain the sign of the gradients frequently enough for the optimizer to converge to a good solution. For regions of the cost landscape where the sign of the gradients cannot be reliably determined, we suggested an alternative optimization scheme that could traverse such regions. We will explore this idea in future work.

In general it is not clear how to assess the model quality in generative adversarial learning, nor how to find a stopping criteria for the optimization scheme. For example, in the classical setting of computer vision, it is often the case that generated samples are visually evaluated by humans, i.e., the Turing test, or via a proxy artificial neural network. However, the quantum setting does not allow for these approaches in a straightforward manner. We therefore designed an efficient heuristic based on an estimate of the entanglement entropy of a single qubit, and numerically showed that convergence of this quantity to an upper bound also indicate saturation of the adversarial algorithm. We therefore propose this approach as a stopping criterion for the optimization process. We conjecture that similar ideas could be used for regularization in quantum circuit learning for classification and regression.

We tested the quality of approximation as a function of the complexity of the generator and discriminator for simulations of up to six qubits. Our results indicate that investing more resources in the discriminator circuit may lead to more noticeable improvements, while the power of the generator had a relatively smaller impact.

An interesting avenue for future work is the effect of the circuit layout, i.e., type of gates and parameter initialization. If prior information about the target state is available, or can be efficiently extracted, it can also be encoded in the generator circuit by using a suitable layout. For example, in Ref. [24] the authors use the Chow-Liu tree to displace CNOT gates such that they capture most of the mutual information among variables. Similarly, more structured layouts could be used for the discriminator circuit such as hierarchical [26] and universal topologies [27]. These choices could reduce the number of parameters and simplify the cost landscape.

An adversarial learning framework capable of handling mixed states has been recently put forward [2, 11], but no implementation compatible with near-term NISQ computers was provided. In comparison, our framework works well for approximating pure target states and can be readily applied to quantum state tomography applications on NISQ computers.

We finally suggest the exploration of arguably more interesting generalizations of adversarial games in future work. These can in particular be based on more complex information theoretical quantities. In this work we relied on the variational definition of Bayesian probability of error, which assumes the availability of a single copy of the quantum state to discriminate. By assuming the availability of multiple copies, which is in practice the case, one can derive more general adversarial games. These could be based on variational definitions of the quantum Chernoff bound [21], the Umegaki relative information, and other related measures of distinguishability [19].

V. ACKNOWLEDGEMENTS

The authors want to thank Ashley Montanaro for helpful discussions on random projections and pointing out reference [36]. M.B. is supported by the UK Engineering and Physical Sciences Research Council (EPSRC) and by Cambridge Quantum Computing Limited (CQCL). E.G. is supported by EPSRC [EP/P510270/1]. L.W. is supported by the Royal Society. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. S.S. is supported by the Royal Society, EPSRC, the National Natural Science

Foundation of China, and the grant ARO-MURI W911NF17-1-0304 (US DOD, UK MOD and UK EPSRC under the Multidisciplinary University Research Initiative).

-
- [1] The Caltech Archives, “Richard Feynman’s blackboard at time of his death,” (1988).
 - [2] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost, “Quantum principal component analysis,” *Nature Physics* **10**, 631 (2014).
 - [3] Mohammad H Amin, Evgeny Andriyash, Jason Rolfe, Bohdan Kulchytskyy, and Roger Melko, “Quantum Boltzmann machine,” *Physical Review X* **8**, 021050 (2018).
 - [4] Mária Kieferová and Nathan Wiebe, “Tomography and generative training with quantum Boltzmann machines,” *Physical Review A* **96**, 062327 (2017).
 - [5] Scott Aaronson, “The learnability of quantum states,” in *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, Vol. 463 (The Royal Society, 2007) pp. 3089–3114.
 - [6] Andrea Rocchetto, Scott Aaronson, Simone Severini, Gonzalo Carvacho, Davide Poderini, Iris Agresti, Marco Bentivegna, and Fabio Sciarrino, “Experimental learning of quantum states,” arXiv preprint arXiv:1712.00127 (2017).
 - [7] Song Cheng, Jing Chen, and Lei Wang, “Information perspective to probabilistic modeling: Boltzmann machines versus Born machines,” arXiv preprint arXiv:1712.04144 (2017).
 - [8] Marcello Benedetti, Delfina Garcia-Pintos, Yunseong Nam, and Alejandro Perdomo-Ortiz, “A generative modeling approach for benchmarking and training shallow quantum circuits,” arXiv preprint arXiv:1801.07686 (2018).
 - [9] 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 preprint arXiv:1708.09757 (2017).
 - [10] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, “Generative adversarial nets,” in *Advances in neural information processing systems* (2014) pp. 2672–2680.
 - [11] Pierre-Luc Dallaire-Demers and Nathan Killoran, “Quantum generative adversarial networks,” arXiv preprint arXiv:1804.08641 (2018).
 - [12] Seth Lloyd and Christian Weedbrook, “Quantum generative adversarial learning,” arXiv preprint arXiv:1804.09139 (2018).
 - [13] Carl W Helstrom, “Quantum detection and estimation theory,” *Journal of Statistical Physics* **1**, 231–252 (1969).
 - [14] 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 (2014).
 - [15] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, “A quantum approximate optimization algorithm,” arXiv preprint arXiv:1411.4028 (2014).
 - [16] Guillaume Verdon, Michael Broughton, and Jacob Biamonte, “A quantum algorithm to train neural networks using low-depth circuits,” arXiv preprint arXiv:1712.05304 (2017).
 - [17] Martin Riedmiller and Heinrich Braun, “Rprop - a fast adaptive learning algorithm,” in *Proc. of ISCIS VII*, *Universitat (Citeseer, 1992)*.
 - [18] John Preskill, “Quantum computing in the NISQ era and beyond,” arXiv preprint arXiv:1801.00862 (2018).
 - [19] Christopher A Fuchs, “Distinguishability and accessible information in quantum theory,” arXiv preprint quant-ph/9601020 (1996).
 - [20] Michael A Nielsen and Isaac Chuang, “Quantum computation and quantum information,” (2002).
 - [21] Koenraad MR Audenaert, John Calsamiglia, Ramón Muñoz-Tapia, Emilio Bagan, Ll Masanes, Antonio Acín, and Frank Verstraete, “Discriminating states: The quantum Chernoff bound,” *Physical review letters* **98**, 160501 (2007).
 - [22] M Neumark, “Spectral functions of a symmetric operator,” *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya* **4**, 277–318 (1940).
 - [23] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii, “Quantum circuit learning,” arXiv preprint arXiv:1803.00745 (2018).
 - [24] Jin-Guo Liu and Lei Wang, “Differentiable learning of quantum circuit Born machine,” arXiv preprint arXiv:1804.04168 (2018).
 - [25] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven, “Barren plateaus in quantum neural network training landscapes,” arXiv preprint arXiv:1803.11173 (2018).
 - [26] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G Green, and Simone Severini, “Hierarchical quantum classifiers,” arXiv preprint arXiv:1804.03680 (2018).
 - [27] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni, “Universal discriminative quantum neural networks,” arXiv preprint arXiv:1805.08654 (2018).
 - [28] Christian Igel and Michael Hüsken, “Improving the Rprop learning algorithm,” in *Proceedings of the second international ICSC symposium on neural computation (NC 2000)*, Vol. 2000 (Citeseer, 2000) pp. 115–121.
 - [29] Lucas Theis, Aäron van den Oord, and Matthias Bethge, “A note on the evaluation of generative models,” arXiv preprint arXiv:1511.01844 (2015).
 - [30] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems* (2016) pp. 2234–2242.

- [31] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani, “Strengths and weaknesses of quantum computing,” *SIAM journal on Computing* **26**, 1510–1523 (1997).
- [32] Roman Schmied, “Quantum state tomography of a single qubit: comparison of methods,” *Journal of Modern Optics* **63**, 1744–1758 (2016).
- [33] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf, “Quantum fingerprinting,” *Physical Review Letters* **87**, 167902 (2001).
- [34] Lukasz Cincio, Yiğit Subaşı, Andrew T Sornborger, and Patrick J Coles, “Learning the quantum algorithm for state overlap,” *arXiv preprint arXiv:1803.04114* (2018).
- [35] Sanjoy Dasgupta and Anupam Gupta, “An elementary proof of a theorem of Johnson and Lindenstrauss,” *Random Structures & Algorithms* **22**, 60–65 (2003).
- [36] Aram W Harrow, Ashley Montanaro, and Anthony J Short, “Limitations on quantum dimensionality reduction,” in *International Colloquium on Automata, Languages, and Programming* (Springer, 2011) pp. 86–97.