




## Reinforcement-learning-assisted quantum optimization

Matteo M. Wauters <sup>1</sup>, Emanuele Panizon <sup>2</sup>, Glen B. Mbeng <sup>3</sup>, and Giuseppe E. Santoro<sup>1,4,5</sup>

<sup>1</sup>SISSA, Via Bonomea 265, I-34136 Trieste, Italy

<sup>2</sup>Fachbereich Physik, Universität Konstanz, 78464 Konstanz, Germany

<sup>3</sup>Universität Innsbruck, Technikerstraße 21 a, A-6020 Innsbruck, Austria

<sup>4</sup>International Centre for Theoretical Physics (ICTP), P.O.Box 586, I-34014 Trieste, Italy

<sup>5</sup>CNR-IOM Democritos National Simulation Center, Via Bonomea 265, I-34136 Trieste, Italy



(Received 29 April 2020; accepted 28 August 2020; published 18 September 2020)

We propose a reinforcement learning (RL) scheme for feedback quantum control within the quantum approximate optimization algorithm (QAOA). We reformulate the QAOA variational minimization as a learning task, where an RL agent chooses the control parameters for the unitaries, given partial information on the system. Such an RL scheme finds a policy converging to the optimal adiabatic solution of the quantum Ising chain that can also be successfully transferred between systems with different sizes, even in the presence of disorder. This allows for immediate experimental verification of our proposal on more complicated models: the RL agent is trained on a small control system, simulated on classical hardware, and then tested on a larger physical sample.

DOI: [10.1103/PhysRevResearch.2.033446](https://doi.org/10.1103/PhysRevResearch.2.033446)

### I. INTRODUCTION

Quantum optimization and control are at the leading edge of current research in quantum computation [1]. Quantum annealing (QA) [2–6], *alias* adiabatic quantum computation (AQC) [7,8], is a promising quantum algorithm implemented [9] in present noisy intermediate-scale quantum devices [10]. More recently, the quantum approximate optimization algorithm (QAOA) [11]—a hybrid quantum-classical variational optimization scheme [12]—has gained momentum [13–16] and has been successfully realized in several experimental platforms [17,18].

In QA/AQC, one constructs an interpolating Hamiltonian  $\hat{H}(s) = s\hat{H}_z + (1-s)\hat{H}_x$ , where, e.g., for spin-1/2 systems  $\hat{H}_z$  is the problem Hamiltonian whose ground state (GS) we are searching [19] while  $\hat{H}_x = -h \sum_j \hat{\sigma}_j^x$  is a transverse field term. Adiabatic dynamics is then attempted by slowly increasing  $s(t)$  from  $s(0) = 0$  to  $s(\tau) = 1$  in a large annealing time  $\tau$ , starting from some easy-to-prepare initial state  $|+\rangle$ , the GS of  $\hat{H}_x$ . The difficulty is usually associated with the growing annealing time  $\tau$  necessary when the system crosses a transition point, especially of first order [20].

QAOA, instead, uses a variational *Ansatz* of the form

$$|\psi_P(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle = \left( \prod_{t=1}^P e^{-i\beta_t \hat{H}_z} e^{-i\gamma_t \hat{H}_x} \right) |+\rangle, \quad (1)$$

where  $\boldsymbol{\gamma} = \gamma_1, \dots, \gamma_P$  and  $\boldsymbol{\beta} = \beta_1, \dots, \beta_P$  are  $2P$  real parameters. The state  $|\psi_P(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle$  is as a sequence of quantum gates, corresponding to  $2P$  unitaries applied to the initial state,

each parameterized by control parameters  $\gamma_t$  or  $\beta_t$ . QAOA consists of a classical minimization in the  $2P$ -dimensional energy landscape, which is in general not a trivial task [21], because local optimizations tend to get trapped into one of the many local minima, producing irregular parameters  $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ , hard to implement and sensitive to noise. To obtain stable and regular schedules  $(\boldsymbol{\gamma}^*, \boldsymbol{\beta}^*)$ , easily generalized to different values of  $P$  and implemented experimentally, iterative procedures should be employed [14,16,17]. For quantum Ising chains, smooth regular optimal parameters can be found [14], which are *adiabatic* in a digitized-QA/AQC [22] context.

One might regard QAOA as an optimal control process [23] in which one acts sequentially on the system in order to maximize a final reward. This reformulation seems particularly suited for reinforcement learning (RL) [24–27]. As schematically represented in Fig. 1(a), at each discrete time step  $t$  an “agent” is given some information, through some observables  $O_{t-1}$  measured on the state  $S_{t-1} = |\psi_{t-1}\rangle$  of the system on which it acts (the “environment”). The agent then performs an action  $a_t$ —here choosing  $(\gamma_t, \beta_t)$ —obtaining a new state  $S_t = |\psi_t\rangle$  and receiving a “reward”  $r_t$ , measuring the quality of  $|\psi_t\rangle$ .

Several questions come to mind, which have not been addressed in the recent literature on RL applied to quantum problems [28–36]: (i) is RL-assisted QAOA able to “learn” *optimal* schedules? (ii) Are the schedules found *smooth* in  $t$ ? (iii) How to dwell with the fact that getting information from  $|\psi_t\rangle$  involves quantum measurements which *destroy* the state? (iv) Are the strategies learned *transferable* to larger systems?

In this article we show, on the paradigmatic example of the transverse field Ising chain, that optimal strategies can be effectively learned with the proximal policy optimization (PPO) algorithm [37] employing very small neural networks (NN). We show that RL automatically learns *smooth* schedules, hence realizing an optimal controlled digitized-QA algorithm [14,38]. By working with disordered quantum Ising chains

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article’s title, journal citation, and DOI.

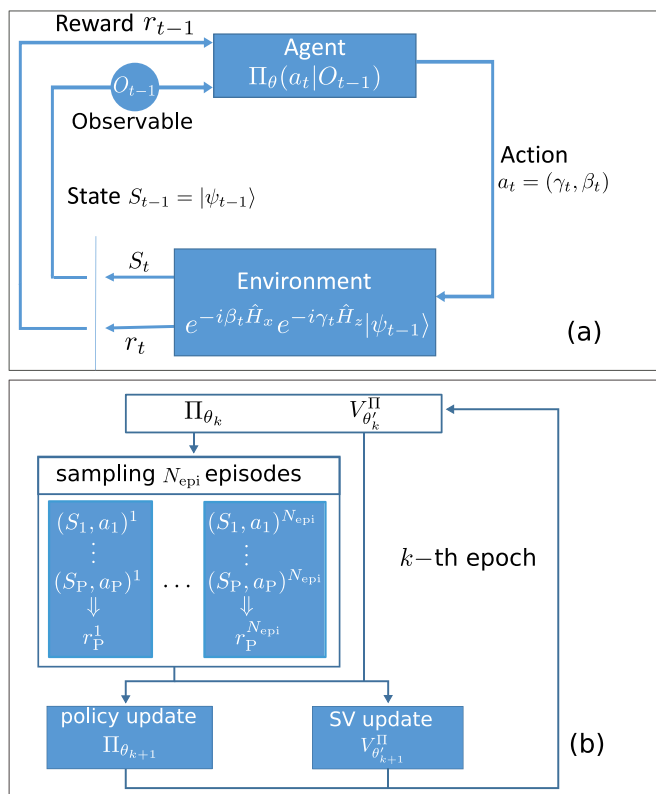


FIG. 1. Scheme of (a) a single step of Reinforcement Learning for QAOA; (b) the “episodes” loop in each  $k$ th training “epoch,” with the “policy” and “state-value” neural networks  $\Pi_{\theta_k}$  and  $V_{\theta_k}^\Pi$ .

we show that strategies “learned” on small samples can be successfully transferred to larger systems, hence alleviating the “measurement problem”: one can learn a strategy on a small problem which can be simulated on a computer, and implement it on a larger experimental setup [39].

The rest of the paper is organized as follows. In Sec. II, we describe the connection between QAOA and reinforcement learning and illustrate our method. In Sec. III, we report our results on the transverse field Ising model (TFIM) in one dimension, with periodic boundary conditions, where detailed QAOA results are already known [14,40] and exact numerical results are obtained via the Jordan-Wigner transformation [41]. Further benchmarks on the Lipkin-Meshkov-Glick model (LMG), i.e., Ising with infinite range interaction, are reported in Sec. IV. We draw our conclusions and present future outlooks in Sec. V.

## II. RL-ASSISTED QAOA

Here we describe how to implement a QAOA ground state search aided by an RL agent. In order to make our arguments clearer, we specify the discussion on the uniform TFIM model. To apply our method to a different spin model, it is sufficient to change the problem specific Hamiltonian  $\hat{H}_z$ .

We define the target Hamiltonian  $\hat{H}_{\text{targ}} = \hat{H}_z + h\hat{H}_x$  with

$$\hat{H}_z = -J \sum_{j=1}^N \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z, \quad \hat{H}_x = - \sum_j \hat{\sigma}_j^x. \quad (2)$$

Given a set of QAOA parameters  $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ , we gauge the quality of the resulting state from the residual energy density

$$\epsilon_P^{\text{res}}(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \frac{E_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) - E_{\min}}{E_{\max} - E_{\min}}, \quad (3)$$

where  $E_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) = \langle \psi_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) | \hat{H}_{\text{targ}} | \psi_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle$  is the variational energy, and  $E_{\max}$  and  $E_{\min}$  are the highest and lowest eigenvalues of the target Hamiltonian. Specifically, the results presented below will concern targeting the ground state for  $h = 0$ , although the approach can be easily extended to the case with  $h > 0$ . At  $h = 0$ , the residual energy is bounded by the inequality [14]

$$\epsilon_P^{\text{res}}(\boldsymbol{\gamma}, \boldsymbol{\beta}) \geq \begin{cases} \frac{1}{2P+2} & \text{if } 2P < N \\ 0 & \text{if } 2P \geq N \end{cases}, \quad (4)$$

which becomes an equality when  $(\boldsymbol{\gamma}, \boldsymbol{\beta})$  are optimal QAOA parameters.

The key ingredients of the RL-assisted algorithm are as follows.

*State.* The state  $S_t$  at time step  $t = 1, \dots, P$  is encoded by the wave-function  $|\psi_t\rangle$ , defined iteratively as  $|\psi_t\rangle = e^{-i\beta_t \hat{H}_x} e^{-i\gamma_t \hat{H}_z} |\psi_{t-1}\rangle$ , with  $|\psi_0\rangle = |+\rangle \equiv \frac{1}{\sqrt{2^N}} \bigotimes_i (|\uparrow\rangle_i + |\downarrow\rangle_i)$ . Due to the symmetry of both  $\hat{H}_x$  and  $\hat{H}_z$ ,  $|\psi_t\rangle$  is always  $\mathbb{Z}_2$  symmetric. The agent has partial information through a number of *observables*  $O_{t-1}$  measured on  $|\psi_{t-1}\rangle$ . Our choice (with  $t-1 \rightarrow t$ ) is

$$O_t = \{ \langle \psi_t | \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z | \psi_t \rangle, \langle \psi_t | \hat{\sigma}_j^x | \psi_t \rangle \}, \quad (5)$$

where a single value of  $j$  is enough when translational invariance is respected. Interestingly, the agent seems to achieve comparable results even with a single observable  $O_t = \langle \psi_t | \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z | \psi_t \rangle$ .

*Action.* The action  $a_t$  corresponds to  $(\gamma_t, \beta_t)$ . The conditional probability of  $a_t$  given the observables  $O_{t-1}$ —called “policy” in RL—is denoted by  $\Pi_\theta(a_t|O_{t-1})$ , where  $\theta$  are the parameters of a NN.  $\Pi_\theta(a|O)$  is a stochastic Gaussian policy [24,42], whose mean and standard deviation are computed by the NN.

*Reward.* A reward  $r_t$  is calculated at time  $t$ . In our present implementation,  $r_{t=1, \dots, P-1} = 0$  and only  $r_P > 0$ . The final reward  $r_P = R(E_P)$  is associated to minimizing the final expectation value  $E_P = \langle \psi_P | \hat{H}_{\text{targ}} | \psi_P \rangle$ . Here,  $R(E_P)$  is monotonically increasing when  $E_P$  decreases. Specifically, we take  $R(E_P) = -E_P$ , but different nonlinear choices have been tested.

*Training.* The training process consists of a number  $N_{\text{epo}}$  of “epochs”, as sketched in Fig. 1(b). During each epoch the RL agent explores, with a *fixed* policy,  $N_{\text{epi}}$  state-action trajectories, or “episodes,” each involving  $P$  steps  $t = 1, \dots, P$ . At the end of each epoch, the policy is updated to favor trajectories with higher reward. The particular RL algorithm we used is the proximal policy optimization (PPO) algorithm [37], from the OpenAI SpinningUp library [42]. PPO is an actor-critic algorithm where two independent NNs are used to parametrize the policy  $\Pi_\theta(a_t|O_{t-1})$  and the state-value function [24]  $V_{\theta'}^\Pi(O_t) = \mathbb{E}^\Pi[r_P]$ , which estimates the expected reward for a system in a state with observables  $O_t$  and evolving with the policy  $\Pi$ .  $V_{\theta'}^\Pi(O_t)$  is used to calculate the updates

after each epoch [42]. In our numerical simulations, we used NNs with two fully connected hidden layers of 32 and 16 neurons respectively, and linear-rectification (ReLU) activation function. We discuss in Appendix A our choices regarding training parameters, observables, and reward function.

### III. RESULTS: ISING MODEL

Let us now focus on the numerical results obtained on the one dimensional Ising model with PBC, both with uniform and random couplings. As a recap, the problem Hamiltonian reads

$$\hat{H}_z = - \sum_{j=1}^N J_j \hat{\sigma}_j^z \hat{\sigma}_{j+1}^z. \quad (6)$$

We start considering the uniform TFIM, where  $J_j = J$ . The model has a paramagnetic ( $h > J$ ) and a ferromagnetic ( $h < J$ ) phase, separated by a second-order transition at  $h = J$ . In the RL training, the system is initially prepared in the state  $|\psi_0\rangle = |+\rangle$ , while the NNs for the policy and the state-value function are both initialized with random parameters. The agent is then trained for  $N_{\text{epo}} = 1024$  epochs, each comprising  $N_{\text{epi}} = 100$  episodes of  $P$  steps each. After training, we test the RL algorithm with  $\sim 50$  runs.

Figure 2(a) shows the results obtained by the RL-trained policy. For  $P \leq 6$ , the trained RL agent finds optimal QAOA parameters, saturating the bound for  $\epsilon_P^{\text{res}}$  in Eq. (4). In particular, for small system sizes  $N$ , when  $P > N/2$ , the agent finds the exact target ground state, and  $\epsilon_P^{\text{res}} = 0$ . For longer episodes ( $P > 6$ ), the residual energy deviates from the lower bound due to two factors: (i) the longer the episode, the more difficult it is to learn the policy, as a larger number of training epochs are necessary to reach convergence; (ii) since we are using a stochastic policy, the error due to the finite width of the action distributions is accumulated during an episode, leading to larger relative errors for longer trajectories. To cure this fact, we supplement the RL-trained policy with a final gradient based *local optimization* (LO) of the parameters ( $\gamma, \beta$ ), employing the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm [43], see Appendix B for more details on the algorithm. This last step is computationally cheap, since the RL training brings the agent already close to a local minimum, provided  $N_{\text{epo}}$  is large enough. The residual energy data obtained in this way, denoted by RL + LO in Fig. 2(a), falls on top of the optimal curve  $\epsilon_P^{\text{res}} = \frac{1}{2P+2}$ , within numerical precision.

To visualize the action choices, we translate  $\gamma_t$  and  $\beta_t$  into the corresponding interpolation parameter  $s_t$  which a Trotter-digitized QA/AQC would show, which for  $h = 0$  is given by [14]

$$s_t = \frac{\gamma_t}{\gamma_t + \beta_t}. \quad (7)$$

Figure 2(b) shows the interpolation parameter  $s_t$  during an episode, for a chain of  $N = 128$  spins and  $P = 8$ . Different curves are obtained by repeating a test run of the same stochastic policy, trained for  $N_{\text{epo}} = 1024$  epochs. The parameters obtained through the RL policy are smooth, and different tests result in similar s-shaped profiles for  $s_t$ . When a final local minimization is added, the curves for  $s_t$  coalesce

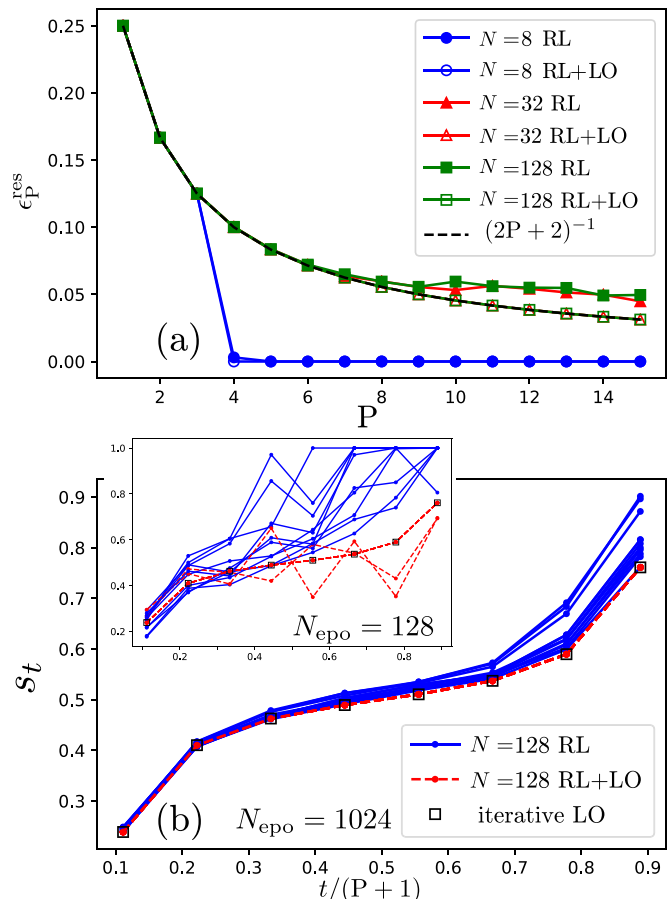


FIG. 2. (a) Residual energy density  $\epsilon_P^{\text{res}}$ , Eq. (3), vs  $P$ . The target state is ferromagnetic with  $h = 0$ . Full symbols: results from RL only; empty symbols: a local optimization (LO) supplements the RL actions (RL + LO); data are averaged over 50 test runs. The black dashed line is the lower bound of Eq. (4). (b) The schedule  $s_t = \gamma_t / (\gamma_t + \beta_t)$ . Full blue lines denote  $s_t$  learned after  $N_{\text{epo}} = 1024$  epochs on a chain of  $N = 128$  sites. After LO (dashed red lines), they all collapse on the minimum corresponding to the iterative LO solution [14] (black empty squares). (Inset) Same data for  $N_{\text{epo}} = 128$  training epochs, where not all the LO optimized actions sets fall onto the iterative LO solution.

and coincides with the smooth optimal schedule obtained in Ref. [14] through an independent iterative local optimization strategy. When the training is at an early stage, i.e., the number of epochs is small, see inset of Fig. 2(b), the profiles  $s_t$  are more irregular and do not fall all in the same smooth minimum upon performing the LO (see the three dashed red lines).

Next, we turn to the random TFIM case. Here, for each chain length  $N$  we fix the disorder instance  $\{J_j\}_{j=1, \dots, N}$  with  $J_j \in [0, 1]$ , both for the training and the test of the RL policy. Despite translational invariance is now lost, the relevant observables  $O_t$  consist only of the two chain-averaged terms  $O_t = \{\langle \psi_t | \hat{H}_z | \psi_t \rangle, \langle \psi_t | \hat{H}_x | \psi_t \rangle\}$ . All the parameters involved in training the NNs are fixed as for the uniform TFIM.

Figure 3(a) shows the residual energy  $\epsilon_P^{\text{res}}$  vs  $P$  obtained from the bare RL (full symbols) and from RL followed by a local optimization (RL + LO, empty symbols). The local optimization significantly improves the quality for large  $P \geq 10$ .

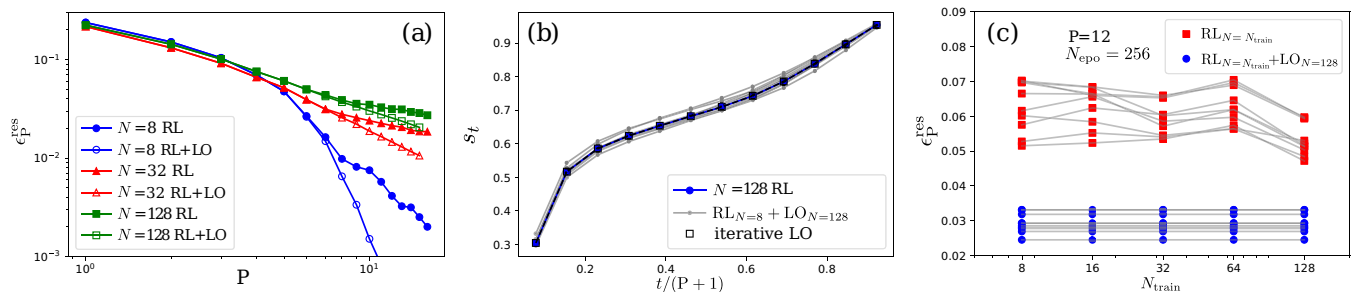


FIG. 3. (a) Residual energy, Eq. (3), vs  $P$  for a single instance of the random TFIM: comparison between bare RL and RL followed by local optimization (RL + LO) results. (b) The optimized  $s_t$  obtained with different procedures. Empty squares: the iterative LO process of Ref. [14]; Blue circles: RL + LO performed directly on a  $N = 128$  chain; gray lines:  $\text{RL}_{N=8} + \text{LO}_{N=128}$ , i.e., training of a  $N = 8$  chain used as *Ansatz* for LO of the  $N = 128$  chain. (c) The residual energy obtained by training the policy on a single disordered instance with  $N_{\text{train}}$  sites and tested on 10 instances of length  $N_{\text{test}} = 128$ , before (red squares) and after (blue circles) a local optimization. Grey lines connect the same disorder instance.

A detailed study of the behavior of  $\epsilon_P^{\text{res}}$  for large  $P$  and a comparison with the results obtained [44] by a linear-QA/AQC scheme, with  $s(t) = t/\tau$ , is left to a future study.

Figure 3(b) shows the optimal parameter  $s_t = \gamma_t/(\gamma_t + \beta_t)$  found by the RL + LO method (filled circles), compared to the  $s_t$  constructed with the iterative optimization strategy described in Ref. [14] (empty squares): the agreement between the two is remarkable, showing that the RL-assisted QAOA effectively “learns” smooth action trajectories. The most remarkable fact, however, is shown by the series of grey lines present in Fig. 3(b). These are obtained by training the RL agent on a much smaller instance with  $N = 8$  sites, and testing the RL-policy to the larger chain with  $N = 128$ , followed by local optimizations of the learned parameters. These results show a large transferability of the RL policies, which holds even in the absence of the final LO. When looking at the residual energy obtained with different training instance sizes, reported in Fig. 3(c), a striking result emerges: after a local optimization (blue circles), the residual energy is independent of the particular training instance and depends *only* on the disordered sample where the LO is performed. The dispersion of the RL + LO data is due only to the 10 different disorder instances. Without LO (red squares), the residual energy displays a mild dependence on  $N_{\text{train}}$ , and the best results are obtained for  $N_{\text{train}} = N_{\text{test}}$ , as expected.

Policy transferability suggests the following way-out from the “measurement problem” involved in the construction of the state observables  $O_t$ . Indeed, in an experimental implementation of RL-assisted QAOA, the RL agent could observe a small system, efficiently simulated on classical hardware, and then use the learned actions to evolve the larger experimental one. This reduces drastically the number of measurements to be performed and allows to test RL-assisted QAOA on physical quantum platforms.

#### IV. RESULTS: FULLY CONNECTED ISING MODEL

To corroborate the results presented in the previous section, we benchmarked our method also on the infinite range Ising ferromagnet, or Lipkin-Meshkov-Glick (LMG) model,

described by the Hamiltonian

$$\hat{H} = -\frac{1}{N} \left( \sum_i \hat{\sigma}_i^z \right)^2 - h \sum_i \hat{\sigma}_i^x. \quad (8)$$

Since  $[\hat{H}, \hat{S}^2] = 0$ , with  $\hat{S}$  the total spin of the system, numerically exact dynamics is accessible in the maximally polarized subspace with  $\hat{S}^2 = \frac{N}{2}(\frac{N}{2} + 1)$ . It also provides another useful benchmark for our method, because it displays some peculiarities within the QAOA framework. It has a very rugged energy landscape [45], which makes local optimizations unstable: it is hard to find good minima when  $P < N/2$  and, in a previous work [45], we failed in finding smooth parameter sets. In particular, the iterative optimization used in Refs. [14,16,17] does not work.

Here we show that reinforcement learning makes the local optimization more reliable than other standard QAOA approaches. We consider a target Hamiltonian corresponding to a nonzero transverse field  $\hat{H}_{\text{target}} = \hat{H}_z + h\hat{H}_x$  and we focus on reaching the ferromagnetic phase when the system is initially prepared in the paramagnetic state  $|+\rangle$ . In Fig. 4(a), we present the data obtained with the different protocols for a chain of  $N = 64$  spins and target transverse field  $h = \frac{\sqrt{5}-1}{2}$ . The quality of RL alone deteriorates rapidly when  $P$  increases, even if still better than QAOA with purely random initialization. When RL is coupled to a subsequent local optimization, the results are much more stable. A data collapse shows that results for different chain length collapse nicely when  $P$  is rescaled with the logarithm of the system size  $N$ , see Fig. 4(b). Thus among QAOA variational *Ansatz* for the LMG model, exist a class of minima that allows to reach very small residual energy with an evolution time increasing only *logarithmically* with the system size  $N$ . However these minima are very hard to find with local optimization, indeed only RL-assisted QAOA is able to address it correctly, among all the techniques we tested (random initialization, linear initialization, iterative local optimization).

Another nice feature of RL-assisted QAOA is the (partial) smoothness of the interpolation parameter  $s_t$ , which was absent in standard QAOA approach. Since the transverse field of



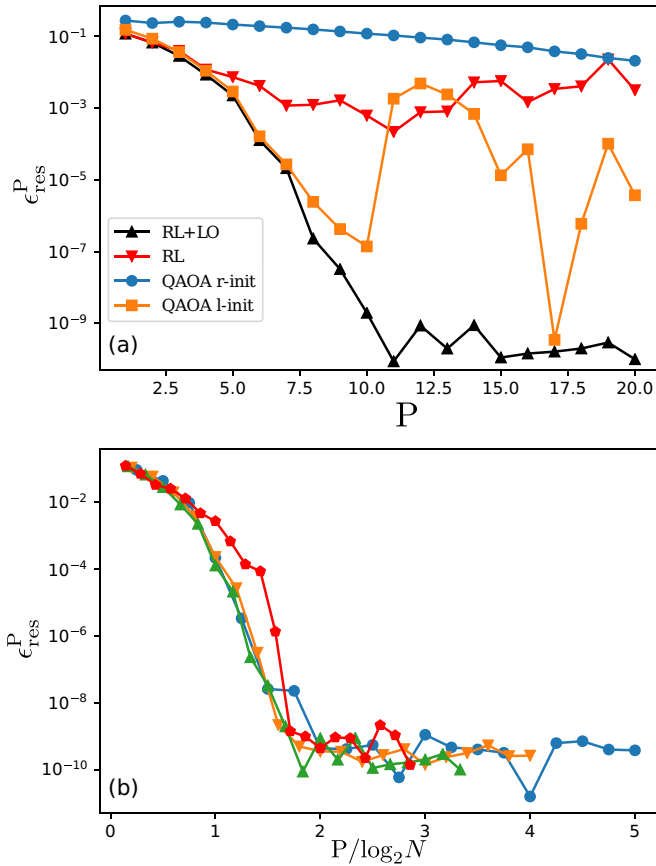


FIG. 4. (a) Comparison between the residual energy curves obtained with different protocols: QAOA with random initialization (blue circles), QAOA with linear initialization (green squares), RL (red triangles), and RL + LO (black triangles). The data refer to a chain of  $N = 64$  spins. (b) data collapse of the residual energy curves after rescaling  $P \rightarrow P/\log_2 N$ .

the target Hamiltonian is nonzero,  $s_t$  takes the form

$$s_t = \frac{\gamma_t}{(1-h)\gamma_t + \beta_t}. \quad (9)$$

In Fig. 5(a), we report the interpolation parameter  $s_t$  (blue lines) for ten different test runs of an RL policy trained on a system of  $N = 64$  spins with  $P = 10$  and target transverse field  $h = \frac{\sqrt{5}-1}{2}$ . Alongside we plot the  $s_t$  relative to the local optimization (dashed red lines) on top of the actions chosen by the RL agent. Comparing with the results obtained from l-init QAOA in Fig. 5(b), the difference in smoothness is striking. Even after the final local optimization, the RL actions are much more regular and the different trajectories clearly suggest the presence of a common basin linked to some continuous schedule  $s(t)$ . An interesting feature of the schedule learned by the algorithm is that it is not the discretization of an annealing protocol that interpolates between  $\hat{H}_{\text{drive}}$  and  $\hat{H}_{\text{target}}$ . Indeed,  $s_t$  does not start close to 0 at the beginning of the episode, but as  $t \rightarrow 0$ ,  $s_t$  reaches a finite value close to 0.5, indicating that the associated continuous schedule  $s(t)$  is not adiabatic with respect to the instantaneous Hamiltonian.

Finally, let us discuss the transferability of the policy in the LMG model. We report in Fig. 6 the residual energies we

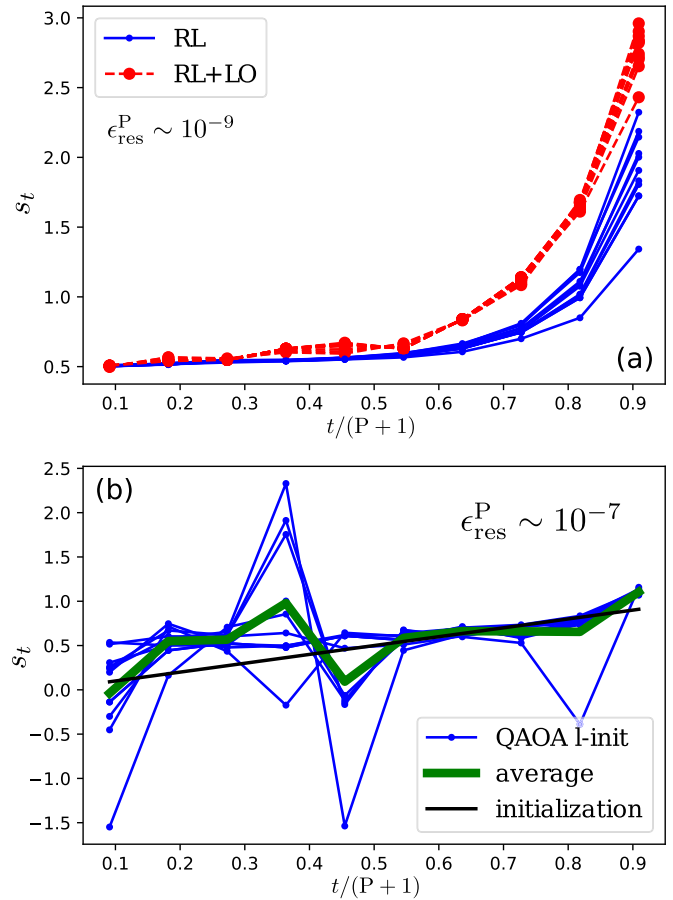


FIG. 5. (a) Learned actions after 1024 training epochs on a chain of  $N = 64$  (blue lines), and their local optimization (red dashed lines). (b) Optimal parameter sets from QAOA with local optimization (blue lines), their average (thick green line) and the linear Ansatz (black line). In both panels,  $P = 10$  and the target transverse field is  $h = \frac{\sqrt{5}-1}{2}$ . In both panels, we indicate the typical residual energy found for those parameters sets.

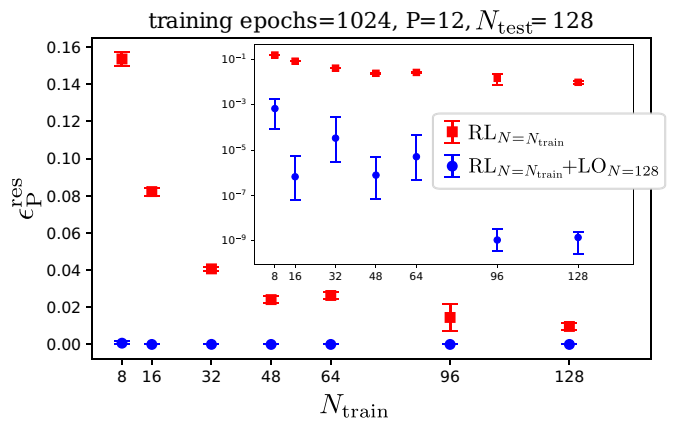


FIG. 6. Residual energy obtained by training the policy on a system of  $N_{\text{train}}$  spins and then applying the actions on a test system of  $N_{\text{test}} = 128$  spins. Red squares refer to the direct transfer of the RL actions, while blue circles have a local optimization on top. The inset shows the same data in log scale, where it is possible to appreciate the difference in the average performance depending on the value of  $N_{\text{train}}$ .

obtained by training the NN on a system of  $N_{\text{train}}$  spins for 1024 epochs and then testing the policy on a larger set with  $N_{\text{test}} = 128$ , both with and without a local optimization on top (blue circles and red squares respectively). At difference with the random TFIM model presented in the main text, now the transferred policy display a clear change in performance depending on the ratio between the training and the test system sizes. The final local optimization tends to smear out this difference in the residual energies, but clearly the performance is better when  $N_{\text{train}} = N_{\text{test}}$ , as indicated also by the smaller errorbar.

## V. CONCLUSION AND OUTLOOK

We have shown that the optimal QAOA strategies for the TFIM [14] can be effectively learned with a simple PPO algorithm [37] employing rather small NNs. The observables measured on a state, referring to the two competing terms in the Hamiltonian and providing information to the “agent,” seem to be effective in the learning process. RL learns *smooth* control parameters, hence realizing an RL-assisted feedback quantum control for the schedule  $s(t)$  of a digitized QA/AQC algorithm [14], in absence of any spectral information. By working with disordered quantum Ising chains, we showed that strategies “learned” on small samples can be successfully transferred to larger systems, hence alleviating the “measurement problem”: one can learn a strategy on a small problem simulated on a computer, and implement it on a larger experimental setup.

A discussion of recent RL-work related to QAOA is here appropriate. References [28,34,36] have all formulated RL strategies to learn optimal variational parameters  $(\boldsymbol{\gamma}, \boldsymbol{\beta})$ . While sharing similar RL tools, their approach is markedly different from ours: they identify the RL “state” with the whole set of QAOA parameters. The agent has no access to the internal quantum state, and no information on the evolution process can be exploited in the optimization. In this way, the issue of measuring the intermediate quantum state is bypassed. This choice, however, reduces RL to a heuristic optimization which forfeits the most relevant feature of the RL framework: the possibility to drive the process with a step-by-step evolution, which takes into account the effect of the previous action, including the possible noise, before choosing the next one. An alternative proposal, closer to ours in methods but tackling different physical questions, has recently appeared in Ref. [35].

Concerning future developments, we mention possible improvements to the “measurement problem.” One possibility is to introduce ancillary bits to provide intermediate information to the RL agent without destroying the state of the system, in a way similar to Ref. [30]. Possible alternatives are performing weak measurements [46], or providing the agent with a set of single-shot measurements [47], instead of the averages of observables. A second issue is the sensitivity to noise: our numerical experiments have shown that noise in the initial state preparation does not harm the ability to learn the correct strategies. Finally, the application to other models is worth pursuing: preliminary results on small Sharrington-Kirkpatrick spin glass samples are encouraging.

## ACKNOWLEDGMENTS

We thank R. Fazio for stimulating discussions. The research was partly supported by EU Horizon 2020 under ERC-ULTRADISS, Grant Agreement No. 834402. G.E.S. acknowledges that his research has been conducted within the framework of the Trieste Institute for Theoretical Quantum Technologies (TQT).

## APPENDIX A: POLICY AND TRAINING PARAMETERS

The results presented in the main text are obtained by training a PPO algorithm for 1024 epochs of 100 episodes each. The reward function is the simplest possible  $r_t = -\delta_{t,P} \langle \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) | \hat{H}_{\text{target}} | \psi(\boldsymbol{\gamma}, \boldsymbol{\beta}) \rangle$ , and the RL agent receives it only at the end of each episode ( $t = P$ ). Here we discuss briefly our choices of training parameters, the *hyperparameters* in RL language.

The PPO algorithm has been chosen because it is one of the most advanced RL methods suited for problems with a continuous action space [37,42], such as QAOA. This algorithm is implemented in the OpenAI SpinningUp [42] library with a stochastic diagonal Gaussian policy. This means that at each step the two parameters which constitute the action  $a_t = (\gamma_t, \beta_t)$  are extracted from independent Gaussian distributions, with the averages given by the output of the Neural Network that parametrizes the policy. The logarithm of the variance of the two Gaussian distributions are also parameters learned during the training process. The code for the Quantum environment is publicly available in Ref. [48].

The reward function must measure the variational quality of the final wavefunction  $|\psi(\boldsymbol{\gamma}, \boldsymbol{\beta})\rangle_P$ , hence it must be a monotonic increasing function of *minus* the final energy  $E_P(\boldsymbol{\gamma}, \boldsymbol{\beta})$ . We tested two choices:

$$R(E_P(\boldsymbol{\gamma}, \boldsymbol{\beta})) = -E_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) \quad (\text{A1})$$

and

$$R(E_P(\boldsymbol{\gamma}, \boldsymbol{\beta})) = e^{-4E_P(\boldsymbol{\gamma}, \boldsymbol{\beta})/N}, \quad (\text{A2})$$

where the system size  $N$  is used to prevent the reward to diverge. The factor 4 increases the steepness of the reward function towards the optimal control value  $E_P(\boldsymbol{\gamma}, \boldsymbol{\beta}) = -N$  (for the TFIM model). A possible advantage of the exponential choice over the linear one is indeed the higher derivative towards the maximum possible reward, which should improve policy optimization when the agent has already reached a good strategy. However, we do not see an appreciable difference between the two choices, as reported in Fig. 7, where we show the convergence of the corresponding residual energy to the QAOA optimal value  $\epsilon_P^{\text{res}} = (2P + 2)^{-1}$ . There is little difference between the two choices of the reward function. Moreover one can see that the residual energy decrease very slowly with the number of epochs, making it inconvenient to try reaching the optimal value by increasing  $N_{\text{epo}}$  instead of adding a local optimization on top of the Reinforcement Learning process.

To choose the number of episodes per training epoch, i.e., the number of trajectories used to evaluate and update the policy, we tested several values  $N_{\text{epo}} = 50, 100, 150, 200$ , on a single TFIM model with  $N = 32$  and  $P = 10$ . We found

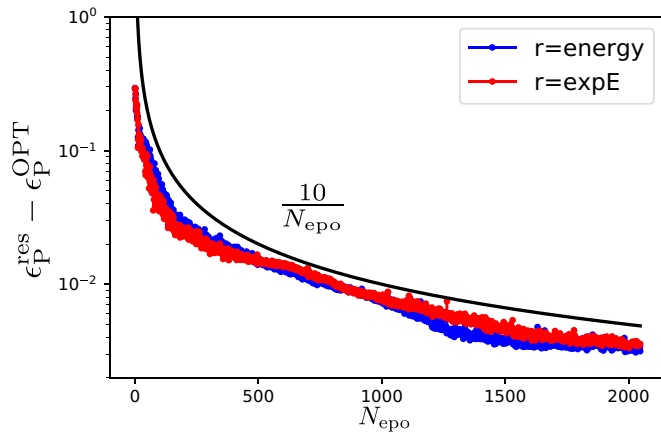


FIG. 7. Difference between the residual energy during the training and QAOA optimal value  $\frac{1}{2(P+2)}$ , for a uniform Ising chain of  $N = 32$  spins and episode length  $P = 10$ . We compare two choices of the reward function given by Eqs. (A1) and (A2), blue and red lines, respectively. The black solid line is an heuristic upper bound for the convergence speed of the residual energy obtained from RL protocol to the optimal value.

that  $N_{\text{epi}} = 100$  gives the lowest average residual energy, as reported in Fig. 8.

Regarding the observables provided to the agent, there are several available choices. The first is full tomography of the wave function, which has the huge disadvantage of requiring an exponentially large number of measurements to provide reliable information. Moreover it is a redundant description of the state, and the neural network needs first to learn how to compress it and extract the relevant information, before optimizing the policy. The performance of the method with this choice turns out to be rather poor and the convergence towards an optimal strategy very slow. Furthermore the number of nodes in the NN has to scale with the system size in order to be able to extract the information, worsening efficiency at larger sizes, and hampering transferability.

When focusing on observables, one of the most intuitive choice is provide the expectation values of the problem and

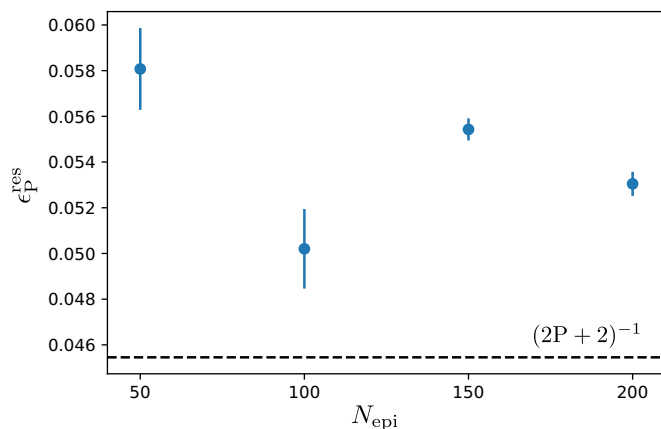


FIG. 8. Average residual energy at the end of the training versus the number of episodes per training epoch. The data are obtained for a uniform Ising chain of  $N = 32$  spins and episode length  $P = 10$ .

the driving Hamiltonians,  $\hat{H}_z$  and  $\hat{H}_x$ . This is what we used throughout this article. This choice has the advantage of being easily accessible from the Jordan-Wigner representation of the Ising chain and allows to visualize the policy as a vector function of two real variables. Moreover it is an efficient description of the state for what regards the optimization task, since  $\langle \hat{H}_z \rangle$  is directly linked to the reward function. This set of observable has been enlarged to include correlation functions at longer distances  $\langle \hat{\sigma}_i^z \hat{\sigma}_{i+k}^z \rangle$ . The test we performed did not indicate any increase in the performance. Preliminary results instead suggest that  $\langle \hat{H}_z \rangle$  alone is sufficient to learn optimal smooth schedules.

An alternative choice would be computing the expectation value of the three component of the magnetization  $\langle \hat{S}^\alpha \rangle$ , with  $\alpha = x, y, z$ , as done in Ref. [35]. This however does not work when the state preserves  $\mathbb{Z}_2$  symmetry, as in the TFIM case.

## APPENDIX B: LOCAL OPTIMIZATION WITH BFGS

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm is an iterative optimization method for nonlinear unconstrained problems, that belong to the class of quasi-Newton method. Convergence is guaranteed to a local stationary point only if the function has a quadratic Taylor expansion around the minimum. However, it can reach good performances also for non smooth problems, such as QAOA.

Let us illustrate briefly how the algorithm works. The goal of the algorithm is to minimize a scalar differentiable function  $f(\mathbf{x})$ , where  $\mathbf{x}$  is an unbounded vector in  $\mathbb{R}^n$ . Starting from an initial estimate of the stationary point  $\mathbf{x}_0$ , the algorithm proceeds iteratively to improve the estimate at each step  $k$ . At each iteration, the algorithm searches for a minimum along the direction  $\mathbf{d}_k$ , given by the solution of the Newton equation:

$$\mathbb{H}_k \mathbf{d}_k = -\nabla f(\mathbf{x}_k) \quad (\text{B1})$$

where  $\mathbb{H}_k$  is an estimated Hessian matrix, also updated iteratively, and  $\nabla f(\mathbf{x}_k)$  is the function gradient.  $\nabla f$  can either be provided analytically or obtained through algorithmic differentiation. The next point  $\mathbf{x}_{k+1}$  is then found by minimizing  $f(\mathbf{x}_k + \gamma \mathbf{d}_k)$  over a scalar parameter  $\gamma > 0$ .

The distinctive feature of BFGS is the how the Hessian estimation is updated. At each step, we impose the quasi-Newton condition on  $\mathbb{H}_{k+1}$ :

$$\mathbb{H}_{k+1} \mathbf{s}_k = \mathbf{y}_k, \quad (\text{B2})$$

where we defined the two quantities

$$\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k), \quad (\text{B3})$$

$$\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k. \quad (\text{B4})$$

The convexity of  $f(\mathbf{x}_k)$ , which is required for convergence, can be verified by checking  $\mathbf{s}_k^\top \mathbf{y}_k > 0$ . This condition must be enforced explicitly, to be sure that  $\mathbb{H}_k$  is always positive definite. Therefore, at each step, BFGS does not compute the whole new Hessian evaluated in  $\mathbf{x}_{k+1}$ , but is updated using two symmetric rank-one matrices  $\mathbb{U}_k$  and  $\mathbb{V}_k$ , chosen in such a way that their sum is a rank-two matrix:

$$\mathbb{H}_{k+1} = \mathbb{H}_k + \mathbb{U}_k + \mathbb{V}_k. \quad (\text{B5})$$

This is easily done by writing the matrices  $\mathbb{U}_k$  and  $\mathbb{V}_k$  as

$$\mathbb{U}_k = \alpha \mathbf{u} \mathbf{u}^\top, \quad (\text{B6})$$

$$\mathbb{V}_k = \beta \mathbf{v} \mathbf{v}^\top, \quad (\text{B7})$$

which indeed guarantees that  $\mathbb{H}_{k+1}$  remains positive definite. It is convenient to choose the vectors  $\mathbf{u}$  and  $\mathbf{v}$  such that  $\mathbf{u} = \mathbf{y}_k$

and  $\mathbf{v} = \mathbb{H}_k \mathbf{s}_k$ . Then imposing Eq. (B2), we obtain

$$\alpha = \frac{1}{\mathbf{y}_k^\top \mathbf{s}_k},$$

$$\beta = -\frac{1}{\mathbf{s}_k^\top \mathbb{H}_k \mathbf{s}_k}. \quad (\text{B8})$$

Finally, we substitute back  $\alpha$  and  $\beta$  into Eqs. (B6) and (B5) and obtain the update rule for the Hessian matrix.

- 
- [1] M. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).
- [2] A. B. Finnila, M. A. Gomez, C. Sebenik, C. Stenson, and J. D. Doll, *Chem. Phys. Lett.* **219**, 343 (1994).
- [3] T. Kadowaki and H. Nishimori, *Phys. Rev. E* **58**, 5355 (1998).
- [4] J. Brooke, D. Bitko, T. F. Rosenbaum, and G. Aeppli, *Science* **284**, 779 (1999).
- [5] G. E. Santoro, R. Martoňák, E. Tosatti, and R. Car, *Science* **295**, 2427 (2002).
- [6] G. E. Santoro and E. Tosatti, *J. Phys. A: Math. Gen.* **39**, R393 (2006).
- [7] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, and D. Preda, *Science* **292**, 472 (2001).
- [8] T. Albash and D. A. Lidar, *Rev. Mod. Phys.* **90**, 015002 (2018).
- [9] M. W. Johnson *et al.*, *Nature (London)* **473**, 194 (2011).
- [10] J. Preskill, *Quantum* **2**, 79 (2018), ISSN 2521-327X.
- [11] E. Farhi, J. Goldstone, and S. Gutmann, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [12] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, *New J. Phys.* **18**, 023023 (2016).
- [13] S. Lloyd, [arXiv:1812.11075](https://arxiv.org/abs/1812.11075).
- [14] G. B. Mbeng, R. Fazio, and G. Santoro, [arXiv:1906.08948](https://arxiv.org/abs/1906.08948).
- [15] M. E. S. Morales, J. Biamonte, and Z. Zimborás, [arXiv:1909.03123](https://arxiv.org/abs/1909.03123).
- [16] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, *Phys. Rev. X* **10**, 021067 (2020).
- [17] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin *et al.*, [arXiv:1906.02700](https://arxiv.org/abs/1906.02700).
- [18] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, S. Boixo, M. Broughton, B. B. Buckley, D. A. Buell *et al.*, [arXiv:2004.04197](https://arxiv.org/abs/2004.04197).
- [19] A. Lucas, *Frontiers in Physics* **2**, 5 (2014).
- [20] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi, *Phys. Rep.* **523**, 127 (2013).
- [21] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, *Nat. Commun.* **9**, 4812 (2018), ISSN 2041-1723.
- [22] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. L. Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen *et al.*, *Nature (London)* **534**, 222 EP (2016).
- [23] D. D'Alessandro, *Introduction to Quantum Control and Dynamics* (Chapman and Hall/CRC, London, 2007).
- [24] R. S. Sutton and A. G. Barto, *Reinforcement Learning, An Introduction*, 2nd ed. (The MIT Press, Cambridge Massachusetts, 2018).
- [25] J. Kober, J. A. Bagnell, and J. Peters, *Int. J. Robot. Res.* **32**, 1238 (2013).
- [26] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, [arXiv:1312.5602](https://arxiv.org/abs/1312.5602).
- [27] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, *Nature (London)* **550**, 354 (2017).
- [28] M. August and J. M. Hernández-Lobato, [arXiv:1802.04063](https://arxiv.org/abs/1802.04063).
- [29] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, *Phys. Rev. X* **8**, 031086 (2018).
- [30] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, *Phys. Rev. X* **8**, 031084 (2018).
- [31] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, *npj Quantum Inf.* **5**, 33 (2019), ISSN 2056-6387.
- [32] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, *npj Quantum Inf.* **5**, 85 (2019), ISSN 2056-6387.
- [33] Z. An and D. L. Zhou, *Europhys. Lett.* **126**, 60002 (2019).
- [34] S. Khairy, R. Shaydulin, L. Cincio, Y. Alexeev, and P. Balaprakash, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, Palo Alto, California, USA), Vol. 34.
- [35] A. Garcia-Saez and J. Riu, [arXiv:1911.09682](https://arxiv.org/abs/1911.09682).
- [36] J. Yao, M. Bukov, and L. Lin, in *Proceedings of Machine Learning Research* 107 (2020).
- [37] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, [arXiv:1707.06347](https://arxiv.org/abs/1707.06347).
- [38] G. B. Mbeng, R. Fazio, and G. E. Santoro, [arXiv:1911.12259](https://arxiv.org/abs/1911.12259).
- [39] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, [arXiv:1910.08187](https://arxiv.org/abs/1910.08187).
- [40] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, *Phys. Rev. A* **97**, 022304 (2018).
- [41] A. P. Young and H. Rieger, *Phys. Rev. B* **53**, 8486 (1996), ISSN 0163-1829.
- [42] J. Achiam, OpenAI SpinningUp libray (2018), <https://spinningup.openai.com/en/latest/algorithms/ppo.html>.
- [43] J. Nocedal and S. Wright, *Numerical Optimization* (Springer Science & Business Media, New York, 2006).
- [44] T. Caneva, R. Fazio, and G. E. Santoro, *Phys. Rev. B* **76**, 144427 (2007).
- [45] M. M. Wauters, G. B. Mbeng, and G. E. Santoro, [arXiv:2003.07419](https://arxiv.org/abs/2003.07419).
- [46] V. Vitale, G. De Filippis, A. de Candia, A. Tagliacozzo, V. Cataudella, and P. Lucignano, *Sci. Rep.* **9**, 13624 (2019).
- [47] K. A. McKiernan, E. Davis, M. S. Alam, and C. Rigetti, [arXiv:1908.08054](https://arxiv.org/abs/1908.08054).
- [48] GitHub <https://github.com/mwauters92/QuantumRL>.