Proceedings of Machine Learning Research vol 107:1044–1081, 2021 2nd Annual Conference on Mathematical and Scientific Machine Learning

Noise-Robust End-to-End Quantum Control using Deep Autoregressive Policy Networks

Jiahao Yao JIAHAOYAO@BERKELEY.EDU
Department of Mathematics, University of California, Berkeley
Berkeley, CA 94720, USA

Paul Köttering[*]
Department of Mathematics, University of California, Berkeley
Berkeley, CA 94720, USA

Hans Gundlach[*]
Department of Mathematics, University of California, Berkeley
Department of Physics, University of California, Berkeley
Berkeley, CA 94720, USA

Lin Lin
Department of Mathematics, University of California, Berkeley
Computational Research Division, Lawrence Berkeley National Laboratory
Challenge Institute for Quantum Computation, University of California, Berkeley
Berkeley, CA 94720, USA

Marin Bukov MGBUKOV@PHYS.UNI-SOFIA.BG
Department of Physics, St. Kliment Ohridski University of Sofia
5 James Bourchier Blvd, 1164 Sofia, Bulgaria

Editors: Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

Abstract

Variational quantum eigensolvers have recently received increased attention, as they enable the use of quantum computing devices to find solutions to complex problems, such as the ground energy and ground state of strongly-correlated quantum many-body systems. In many applications, it is the optimization of both continuous and discrete parameters that poses a formidable challenge. Using reinforcement learning (RL), we present a hybrid policy gradient algorithm capable of simultaneously optimizing continuous and discrete degrees of freedom in an uncertainty-resilient way. The hybrid policy is modeled by a deep autoregressive neural network to capture causality. We employ the algorithm to prepare the ground state of the nonintegrable quantum Ising model in a unitary process, parametrized by a generalized quantum approximate optimization ansatz: the RL agent solves the discrete combinatorial problem of constructing the optimal sequences of unitaries out of a predefined set and, at the same time, it optimizes the continuous durations for which these unitaries are applied. We demonstrate the noise-robust features of the agent by considering three sources of uncertainty: classical and quantum measurement noise, and errors in the control unitary durations. Our work exhibits the beneficial synergy between reinforcement learning and quantum control.

Keywords: Quantum control, Quantum approximate optimization algorithm, Quantum computing, Reinforcement learning, Policy gradient, Autoregressive policy network, Proximal policy optimization, Noise-robust optimization.

[*] P.K. & H.G. contributed equally and listed by a coin flip

1. Introduction

The last decade has seen impressive breakthroughs in Machine Learning (ML), ranging from image classification (Salakhutdinov, 2014; Krizhevsky et al., 2012) to mastering complex video and board games (Mnih et al., 2013; Silver et al., 2016). ML algorithms have opened the door to solving major scientific challenges hitherto considered intractable, such as protein modelling (Rao et al., 2019) and folding (Jumper et al., 2020), or molecular dynamics simulations (Lu et al., 2021).

Deep learning tools and methods quickly found their way into the field of physics (Dunjko and Briegel, 2018; Mehta et al., 2019; Carleo et al., 2019; Carrasquilla, 2020): Supervised learning was found efficient in identifying phase transitions and analyzing experimental data (Carrasquilla and Melko, 2017; Van Nieuwenburg et al., 2017; Bohrdt et al., 2019; Rem et al., 2019). Unsupervised learning brought a new class of variational many-body wavefunctions (Carleo and Troyer, 2017; Carrasquilla and Torlai, 2021), as well as methods to perform tomography on many-body quantum states (Torlai et al., 2018), find conservation laws from data (Iten et al., 2020), identify phase transitions (Wang, 2016; Kottmann et al., 2020), Hamiltonian learning (Valenti et al., 2019), etc. Reinforcement learning (RL) (Sutton and Barto, 2018) brought strategies for navigating turbulent flows (Reddy et al., 2016; Colabrese et al., 2017; Bellemare et al., 2020), and even exploring the string landscape (Halverson et al., 2019), while evolutionary methods have recently been applied to error correction (Théveniaut and van Nieuwenburg, 2021).

The variational character of ML models combined with their intrinsic optimization procedure, provides a natural playground for applications in quantum control (Schäfer et al., 2020; Wang et al., 2021; Sauvage and Mintert, 2020; Fösel et al., 2020; Nautrup et al., 2019; Albarrán-Arriagada et al., 2018; Sim et al., 2021; Wu et al., 2020a,b; Anand et al., 2021). Due to the close relationship between control theory and reinforcement learning, the control of quantum systems has become a major application area of RL algorithms in physics. Notable examples include policy gradient (Niu et al., 2019; Fösel et al., 2018; August and Hernández-Lobato, 2018; Porotti et al., 2019; Wauters et al., 2020; Yao et al., 2020a; Sung, 2020), Q-learning (Chen et al., 2013; Bukov, 2018; Bukov et al., 2018; Sørdal and Bergli, 2019; Bolens and Heyl, 2020) and AlphaZero (Dalgaard et al., 2020b).

Over the years, the physics community has also developed a number of successful quantum control algorithms (Khaneja et al., 2005; Caneva et al., 2011; Peruzzo et al., 2014; Dalgaard et al., 2020a; Magann et al., 2020, 2021), including GRAPE, CRAB, and VQE. One prominent example of the latter is Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014), whose versatility allows for solving complex combinatorial problems using quantum computers (Garcia-Saez and Riu, 2019; Dong et al., 2019; Khairy et al., 2019, 2020; Yao et al., 2020b; Tabi et al., 2020; Bravyi et al., 2020). Quantum control algorithms, such as CRAB or QAOA, come up with an ingenious physics-informed variational ansatz for the structure of control protocols. RL algorithms, on the other hand, are model-free and resilient to uncertainty. Hence, a natural question emerges as to how one can combine both benefits offered by RL and quantum control in a unified framework.

In this paper, our aim is to deploy a generalized QAOA ansatz in combination with an end-to-end deep RL algorithm for a versatile continuous-discrete quantum control [Sec. 2]. We adopt the continuous degrees of freedom of QAOA which offer an increased control accuracy. Additionally, we consider an enhanced variational control ansatz which contains a larger space to select the building blocks of the protocols from; this introduces a second, discrete combinatorial optimization problem. The resulting algorithm, RL-QAOA, realizes greater gains by striking a balance between robustness and versatility: it is resilient to various kinds of uncertainty, a property shared with PG-

QAOA (Yao et al., 2020a); at the same time, RL-QAOA has access to the more general variational counter-diabatic (CD) driving ansatz (Demirplak and Rice, 2005; Masuda and Nakamura, 2009; Guéry-Odelin et al., 2019) through CD-QAOA (Yao et al., 2020b).

However, RL-QAOA presents a number of new challenges, cf. Sec. 3. It requires a mixed continuous-discrete action space so that the RL agent can construct a control protocol by optimizing the order in which unitaries appear in the control sequence; simultaneously, the agent has to also choose the continuous duration to apply each unitary. This requires the use of a suitable ML model to approximate the policy, which allows us to build in temporal causality. Therefore, an essential building block of RL-QAOA is a novel monolithic deep autoregressive policy network[1] that handles continuous and discrete actions on equal footing. To train our RL agent, we derive an extension of Proximal Policy Optimization (PPO) (Schulman et al., 2017) to hybrid discrete-continuous policies.

We apply RL-QAOA to find the ground state of a nonintegrable chain of interacting spin-$1/2$ particles (a.k.a. qubits), and interacting spin-1 particles (a.k.a. qutrits) in a fixed amount of time, cf. Sec. 4. The mixed discrete-continuous degrees of freedom allow the RL agent to construct a short protocol sequence away from the adiabatic regime. We test the agent's behavior in a strongly stochastic environment, by considering three different kinds of noise: classical and quantum measurement noise, and errors in the control unitary gate duration. In Sec. 5, we demonstrate that RL-QAOA is insensitive to the types of noise applied, and outperforms previously developed algorithms based on QAOA in the regime of strong noise.

2. Preliminaries

We open up the discussion by introducing the QAOA ansatz used in quantum control. Following a short overview of reinforcement learning terminology, we review two RL-based QAOA algorithms — PG-QAOA and CD-QAOA — which we aim to blend into a homogeneous hybrid in Sec. 3. The resulting new algorithm combines the benefits of the generalized variational QAOA ansatz, with an RL algorithm performing both continuous and discrete control simultaneously.

2.1. QAOA for Ground State Preparation

Of particular interest in the quest for designing new materials with novel features (such as room-temperature superconductors or topological quantum computers), is the study of ground state properties in quantum many-body physics. Quantum simulators provide an ideal platform to bring together both theory and experiment; yet, they require the ability to prepare a system in its ground state – a formidable challenge for modern quantum computing devices, due to the presence of various sources of uncertainty and noise. The Quantum Approximate Optimization Algorithm (QAOA) (Farhi et al., 2014) provides a widely used state-of-the-art ansatz for this purpose.

Consider a quantum system of N qubits, described by the Hamiltonian H. Starting from an initial quantum state $|\psi_i\rangle$, in QAOA we apply two alternating unitary evolution operators (i.e. quantum gates) (Farhi et al., 2014):

$$|\psi(T)\rangle = U(\{\alpha_j, \beta_j\}_{j=1}^{p})\,|\psi_i\rangle = e^{-iH_2\beta_p} e^{-iH_1\alpha_p} \cdots e^{-iH_2\beta_1} e^{-iH_1\alpha_1}\,|\psi_i\rangle\,. \tag{1}$$

[1] Autoregressive deep neural networks were recently used in physics to learn variational free energies in statistical mechanics models (Wu et al., 2019), and as variational approximators for quantum many-body states (Sharir et al., 2020).

The dynamics are generated by the time-independent operators H_1 and H_2, applied for a duration of $\alpha_j \geq 0$ and $\beta_j \geq 0$, respectively ($j = 1, 2, \cdots, p$ with $p \in \mathbb{N}$). We refer to $q = 2p$ as the total circuit depth. In order to apply QAOA to many-body systems (Ho and Hsieh, 2019), the protocol durations $\{(\alpha_j, \beta_j)\}_{j=1}^p$ are variationally optimized to minimize the expected value of the energy density $\mathcal{E}(\{\alpha_j, \beta_j\}_{j=1}^p) = N^{-1} \langle\psi(T)|H|\psi(T)\rangle$:

$$\{\alpha_j^*, \beta_j^*\}_{j=1}^p = \underset{\{\alpha_j,\beta_j\}_{j=1}^p}{\arg\min}\ \mathcal{E}(\{\alpha_j, \beta_j\}_{j=1}^p), \quad \sum_{j=1}^p(\alpha_j + \beta_j) = T. \tag{2}$$

The additional constraint $\sum_{j=1}^p(\alpha_j + \beta_j) = T$ is required for the resulting protocol to remain in the regime of practical applications, and also for a fair comparison between different algorithms.

As a concrete example to keep in mind, consider the spin-$1/2$ Ising Hamiltonian

$$H = H_1 + H_2, \qquad H_1 = \sum_{i=1}^N J S_{i+1}^z S_i^z + h_z S_i^z, \quad H_2 = \sum_{i=1}^N h_x S_i^x, \tag{3}$$

where $[S_k^\alpha, S_j^\beta] = i\delta_{kj}\varepsilon^{\alpha\beta\gamma}S_j^\gamma$ are the spin-$1/2$ operators. We are interested in preparing the ground state of H, starting from a spin-up polarized initial product state. More details about the physical system are discussed later on in Sec. 4.1 and App. D.

2.2. Reinforcement Learning (RL)

While QAOA defines a variational ansatz to prepare ground states in a unitary process, it does not yet provide a self-contained optimization procedure to find the optimal protocol durations. A universal optimization framework is presented by RL (Sutton and Barto, 2018).

Reinforcement learning comprises a powerful set of algorithms designed to solve control problems. In RL, an agent aims to find a policy π which solves a specific task in a trial-and-error approach based on interactions with the agent's environment. Consider a finite-horizon Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p, r)$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, and $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ defines the transition probability which governs the environment dynamics. Upon selecting an action $a \in \mathcal{A}$, the environment transitions to a new state $s \to s' \in \mathcal{S}$, and emits a reward $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, which the RL agent uses to select subsequent actions. The action $a_j \in \mathcal{A}$ to be selected in a given state $s \in \mathcal{S}$ is determined probabilistically by the instantaneous policy $\pi(a_j|s_j) : \mathcal{A} \times \mathcal{S} \to [0, 1]$. For a given policy π, this process generates a trajectory $\tau = (s_1, a_1, ..., a_q, s_{q+1})$ with probability $\tau \sim \mathbb{P}^\pi(\cdot)$. Here, $\mathbb{P}^\pi(\tau) = p_0(s_1)\pi(a_1|s_1)p(s_2|s_1, a_1)\cdots\pi(a_q|s_q)p(s_{q+1}|s_q, a_q)$, the episode/trajectory length is q, and p_0 is the initial state distribution. The objective in RL is to find the optimal policy, i.e. the policy which maximizes the total expected return: $\mathbb{E}_{\tau\sim\mathbb{P}^\pi}\left[\sum_{j=1}^q r(s_j, a_j)\right]$.

2.3. Policy Gradient Quantum Approximate Optimization Algorithm (PG-QAOA)

A reinforcement learning based approach to QAOA was recently introduced in Ref. (Yao et al., 2020a), using a policy gradient algorithm. The basic idea behind PG-QAOA is to let the RL agent select the durations $\{\alpha_j, \beta_j\}$, which define a continuous action space $\mathcal{A}^{\mathrm{c}}$.

However, casting the quantum control problem within the RL framework comes with certain challenges. The first challenge is that quantum states cannot be directly measured in experiments, which poses questions about the proper definition of the RL state space. To remedy this in an environment following deterministic Schrödinger dynamics, we fix the initial quantum state, and define the RL state as the trajectory of actions $s_j = (a_1^{\mathrm{c}}, \cdots, a_{j-1}^{\mathrm{c}}) = (\alpha_1, \beta_1, \cdots)$ up to episode step j (Bukov, 2018). Note that this definition is clearly inferior to using the full quantum state as defined by the corresponding complex-valued probability amplitudes, since it is tied to a fixed initial state. However, (i) it allows us to accommodate the experimental non-measurability constraint, and (ii) the size of the state space does not grow exponentially with the system size N, which is important in the context of quantum simulation of many-body systems. Even in such a restricted setting, the state space is still exponentially large. Alternatively, one could use the expectation values of observables to define an RL state (Wauters et al., 2020) which also avoids the non-measurability problem. One disadvantage of this choice is that it leads to a partially-observable MDP: indeed, physically very different initial states can have the same expectation values of observables. and it is not a priori clear that the same action is optimal to take for both physical states. Moreover, expectation values presume the ability to perform an average over a number of measurements which also has to be considered with respect to applications in the lab.

The second challenge is the sparsity of the reward signal – a quantum measurement is allowed only once at the end of each episode, since projective measurements collapse the quantum wavefunction and the quantum state is lost irreversibly.

Since the protocol durations are continuous degrees of freedom, we need an RL method for *continuous* optimization. PG-QAOA defines the simplest ansatz: $q = 2p$ independent Gaussian distributions to parameterize the policy, one for each duration $\{\alpha_j, \beta_j\}_{j=1}^p$ in Eq. (2). Since a Gaussian distribution is uniquely determined by its mean μ and standard deviation σ, we need a total of $2p$ independent variational parameters $\boldsymbol{\theta} = \{\mu_{\alpha_j}, \sigma_{\alpha_j}, \mu_{\beta_j}, \sigma_{\beta_j}\}_{j=1}^p$ to parametrize the policy $\pi_{\boldsymbol{\theta}}$:

$$\pi_{\boldsymbol{\theta}}(\{\alpha_j, \beta_j\}_{j=1}^p) = \prod_{j=1}^{p} \pi(\alpha_j; \kappa_{\alpha_j}, \xi_{\alpha_j})\pi(\beta_j; \kappa_{\beta_j}, \xi_{\beta_j}), \tag{4}$$

where $\kappa_{\alpha_j} = \mu_{\alpha_j}$, $\kappa_{\beta_j} = \mu_{\beta_j}$ are the means, and $\xi_{\alpha_j} = \sigma_{\alpha_j}$, $\xi_{\beta_j} = \sigma_{\beta_j}$ are the variances of the Gaussian policy. The actual protocol durations are thus sampled according to $\alpha_j \sim \mathcal{N}(\mu_{\alpha_j}, \sigma_{\alpha_j}^2)$, and similarly for β_j. As was shown in Ref. (Yao et al., 2020a), despite its simplicity, PG-QAOA defines a particularly noise-robust algorithm. In the presence of various kinds of noise, it readily outperforms a number of alternative gradient-free optimization algorithms.

For this study, the original PG-QAOA implementation (Yao et al., 2020a) is not directly applicable, and a modification is required. First, the extensive scaling with increasing the number of qubits suggests using the energy density as a cost function, rather than the many-body fidelity; in doing this, the algorithm no longer requires an explicit reference to the target ground state we are searching for. Second, the original PG-QAOA algorithm does not support an easy implementation of the protocol duration constraint $\sum_{j=1}^p(\alpha_j + \beta_j) = T$. Here, in order to do a fair comparison among different algorithms, we enforce this constraint. Note that this is a nontrivial task for the policy gradient algorithm, for three reasons: (i) protocol durations are sampled from a Gaussian distribution which has unbounded support, (ii) a Gaussian policy supports negative as well as positive samples (yet we require $\alpha_j, \beta_j \geq 0$ for a physical time duration), and (iii) sampled values, even if bounded and nonnegative, are always random, and hence one needs to additionally fix their total sum. We consider two different approaches to resolving (i) and (ii), and apply a normalization trick to fix (iii).

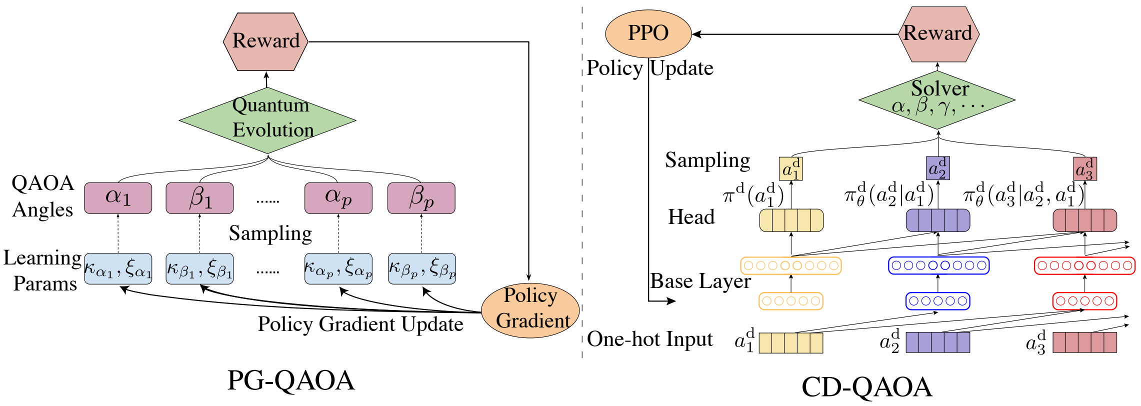

Reward
Quantum Evolution
QAOA Angles
α_1
β_1
......
α_p
β_p
Sampling
Learning Params
$\kappa_{\alpha_1}, \xi_{\alpha_1}$
$\kappa_{\beta_1}, \xi_{\beta_1}$
......
$\kappa_{\alpha_p}, \xi_{\alpha_p}$
$\kappa_{\beta_p}, \xi_{\beta_p}$
Policy Gradient Update
Policy Gradient
PG-QAOA
PPO
Policy Update
Reward
Solver
$\alpha, \beta, \gamma, \cdots$
Sampling
a_1^{d}
a_2^{d}
a_3^{d}
$\pi^{\mathrm{d}}(a_1^{\mathrm{d}})$
$\pi_\theta^{\mathrm{d}}(a_2^{\mathrm{d}}|a_1^{\mathrm{d}})$
$\pi_\theta^{\mathrm{d}}(a_3^{\mathrm{d}}|a_2^{\mathrm{d}}, a_1^{\mathrm{d}})$
Head
Base Layer
One-hot Input
a_1^{d}
a_2^{d}
a_3^{d}
CD-QAOA

Figure 1: Schematic diagram for PG-QAOA (Yao et al., 2020a) [left, see Sec. 2.3] and CD-QAOA (Yao et al., 2020b) [right, see Sec. 2.4]. The PG-QAOA samples continuous QAOA-angles from its policy and variationally updates the policy parameters via policy gradient; CD-QAOA autoregressively samples the gate sequences for the generalized QAOA ansatz and employs the gradient-free solver (Powell algorithm) to solve for their corresponding durations. The policy network is updated via Proximal Policy Optimization (PPO). For a comparison with RL-QAOA, cf. Fig. 2.

The first approach we consider is to define the policy using the Beta distribution (Chou et al., 2017), i.e. $\alpha_j, \beta_j \sim \mathrm{B}(\kappa, \xi)$, instead of a Gaussian, and learn the two nonnegative parameters κ, ξ. Since the Beta distribution is defined on the interval $\mathcal{A}^{\mathrm{c}} = [0, 1]$, it solves the boundedness and positivity problems. The policy is given by Eq. (4) with $\pi(x; \kappa, \xi) = \frac{\Gamma(\kappa+\xi)}{\Gamma(\kappa)\Gamma(\xi)} x^{\kappa-1}(1-x)^{\xi-1}$ the probability density for the Beta distribution; Γ denotes the Gamma function. Note that the number of independent variational parameters $\boldsymbol{\theta} = \{\kappa_{\alpha_j}, \xi_{\alpha_j}, \kappa_{\beta_j}, \xi_{\beta_j}\}_{j=1}^{p}$, remains equal to $4p$.

In the second approach, we pass the output of the Gaussian distribution through a sigmoid activation function (Haarnoja et al., 2018b). Due to the boundedness of the sigmoid function, this restricts the range of all actions/durations to the nonnegative interval $\mathcal{A}^{\mathrm{c}} = [0, 1]$. Hence, our policy is given by Eq. (4) where $\pi(x; \kappa, \xi) = \frac{1}{x(1-x)} \frac{1}{\sqrt{2\pi\xi^2}} \exp\left(-\frac{(\mathrm{logit}(x)-\kappa)^2}{2\xi^2}\right)$ is the probability density for the Sigmoid Gaussian distribution $\mathcal{SN}(\kappa, \xi^2)$[2]. Here, the logit function, $\mathrm{logit}(x) = \log x - \log(1-x)$, is the inverse of the sigmoid function $f(x) = 1/(1+\exp(-x))$, and the factor $\frac{1}{x(1-x)}$ is the inverse Jacobian of $x = f(y)$ over y [cf. App. C]. Notice how the action output of this policy is forced within the interval $[0, 1]$ by construction, without changing the total number of independent variational parameters $\boldsymbol{\theta}$.

Finally, to fix the total protocol duration, (iii), we normalize the sum of durations manually according to $\alpha_j = \frac{\alpha_j}{\sum_{j=1}^{p}(\alpha_j+\beta_j)} T$, $\beta_j = \frac{\beta_j}{\sum_{j=1}^{p}(\alpha_j+\beta_j)} T$. We note that the normalization procedure is considered part of the RL environment, i.e. no gradients are passed through it. In essence, it becomes part of the reward function. This requires us to slightly re-define the meaning of the policy: it generates the bare protocol durations before the normalization; to minimize energy, the durations need the extra normalization.

[2] $\mathcal{SN}(\kappa, \xi^2)$ is short-hard notation for Gaussian distribution $\mathcal{N}(\kappa, \xi^2)$ under the sigmoid transformation $f(x)$.

We mention in passing that this is not the only way to hold the protocol duration T fixed: alternatives include using the constraint to fix the last protocol duration β_p, or the addition of an extra penalty term to the reward function, cf. App. E.

2.4. Quantum Approximate Optimization Ansatz based on Counter-Diabatic Driving

In conventional QAOA, there are two possible gates, corresponding to the two unitaries $U_j = \exp(-i\alpha_j H_j), j = 1, 2$. Therefore, there exist only two distinct sequences of unitaries: $\tau_1^{\mathrm{d}} = U_1U_2U_1U_2\cdots$ and $\tau_2^{\mathrm{d}} = U_2U_1U_2U_1\cdots$. A generalization of this ansatz was considered in Ref. (Yao et al., 2020b), where an RL agent was given the complex combinatorial task to construct the sequence of unitaries τ^{d}, out of a predefined set $\mathcal{A}^{\mathrm{d}}$ of $|\mathcal{A}^{\mathrm{d}}|$ gates/unitaries. For the gate duration notation, we will use α_j [3] for all durations instead of alternating α_j, β_j due to a general ansatz. This set can, in principle, be chosen arbitrarily; however, one can also make a more physics-informed choice, e.g., inspired by counter-diabatic driving in the case of quantum many-body systems [cf. App. A]. In the latter case, the resulting generalized algorithm, called CD-QAOA, was demonstrated to drastically enhance the variational ansatz of QAOA when applied to many-body quantum chains, allowing for shorter circuit depths at no performance cost (Yao et al., 2020b).

Similar to PG-QAOA, CD-QAOA does *not* use the quantum wavefunction to perform the optimization, and the state is $s_j = (a_1^{\mathrm{d}}, \cdots, a_{j-1}^{\mathrm{d}})$ at episode step j. Rewards are given once per episode in the end, and are defined by the (negative) energy density. However, the action space is given by the set of $|\mathcal{A}^{\mathrm{d}}|$ unitary gates from which the protocol sequence τ^{d} are selected; it does not involve the continuous protocol durations which are found as part of the RL environment. In this study, we use the gradient-free Powell algorithm (Powell, 1964) instead of the gradient-based SLSQP algorithm (Kraft et al., 1988) presented in the original CD-QAOA paper.

Apart from the low-level optimization mentioned above, CD-QAOA adopts a two-level optimization schedule (Li et al., 2020; Melnikov et al., 2020): high-level discrete optimization is used to construct the optimal sequence τ^{d} out of the available set of unitaries. For this purpose, in Ref. (Yao et al., 2020b), it was suggested to employ Proximal Policy Optimization (Schulman et al., 2017) (PPO), an advanced variant of policy gradient, aided by a deep autoregressive neural network to implement causality:

$$\pi_{\boldsymbol{\theta}}^{\mathrm{d}}\left(a_1^{\mathrm{d}}, a_2^{\mathrm{d}}, \cdots, a_q^{\mathrm{d}}\right) = \pi_{\boldsymbol{\theta}}^{\mathrm{d}}\left(a_1^{\mathrm{d}}\right) \prod_{j=2}^{q} \pi_{\boldsymbol{\theta}}^{\mathrm{d}}\left(a_j^{\mathrm{d}} \mid a_1^{\mathrm{d}}, \cdots, a_{j-1}^{\mathrm{d}}\right). \tag{5}$$

Each factor in the product above is a categorical distribution over the action space. We point out that the search for the optimal sequence τ^{d} represents a *discrete* optimization problem. This should be contrasted with the low-level *continuous* optimization employed by QAOA to find the optimal durations $\{\alpha_j\}_{j=1}^{q}$, carried out using the Powell solver. Since the Powell solver only handles bounded optimization, we apply the same normalization trick to enforce the total duration constraint.

Given the complete protocol sequence $\tau^{\mathrm{d}} = (a_1^{\mathrm{d}}, \cdots, a_q^{\mathrm{d}})$, we can construct the unitary process

$$U(\{\alpha_j\}_{j=1}^{q}, \tau^{\mathrm{d}}) = \prod_{j=1}^{q} U_{\tau_j^{\mathrm{d}}}(\alpha_j) \tag{6}$$

[3] α_j represents a_j^{c} after normalization

Method	QAOA	PG-QAOA	CD-QAOA	RL-QAOA
protocol sequence optimization (discrete)	✗	✗	∇-free	∇-free
gate durations optimization (continuous)	∇-free	∇-free	∇-free	∇-free
RL optimization	✗	continuous	discrete	continuous & discrete
noise-robust	✗	✓	✗	✓
autoregressive	✗	✗	✓	✓

Table 1: Comparison between all four algorithms: QAOA, PG-QAOA, CD-QAOA and RL-QAOA.

which we use as a generalized QAOA ansatz. The sequences τ^{d}, and the durations $\{\alpha_j\}_{j=1}^q$ are found by minimizing the energy density, cf. Eq. (2). In doing so, we impose an extra constraint that the same action cannot be taken twice in a row, for otherwise one can consider a smaller sequence length by adding the corresponding durations and optimizing them together.

In order to construct the unitary $U(\{\alpha_j\}_{j=1}^q, \tau^{\mathrm{d}})$ from Eq. (6), the RL agent needs to select the sequence τ^{d} of subprocess generators $\mathcal{A}^{\mathrm{d}}$. Hence, at every step j in the RL episode, the agent's action consists of a choice of a Hermitian operator $H_{\tau_j^{\mathrm{d}}} \in \mathcal{A}^{\mathrm{d}}$. A suitable discrete actions space $\mathcal{A}^{\mathrm{d}}$ can be constructed using insights from counter-diabatic (CD) driving, cf. App. A.

3. Mixed Discrete-Continuous Policy Gradient using Deep Autoregressive Networks

Although RL is used as an optimizer in both PG-QAOA and CD-QAOA, it serves two fundamentally different purposes. In PG-QAOA it is employed for continuous optimization of the protocol durations $\{\alpha_j\}$, while in CD-QAOA it is used to find the solution to the discrete combinatorial task of ordering the unitaries in the protocol sequence. In this section, we illustrate how to combine the two aspects together into a unified monolithic RL-based algorithm.

We have seen that with the help of RL one can tremendously enhance the properties of the QAOA ansatz in very different ways, cf. Table 1. For instance, PG-QAOA has the important desired property that it is robust to noise. Moreover, it does a completely gradient-free optimization of the continuous protocol durations. On the other hand, CD-QAOA, enhances the variational ansatz itself by offering the appealing ability to select the order in which three or more unitaries can be applied in the protocol sequence. Moreover, it also introduces an autoregressive deep neural network to encode causality (i.e., which unitary is optimal at a given episode step depends on the unitaries chosen hitherto). The imminent question arises as to whether we can design an algorithm which makes the best of both worlds.

3.1. Autoregressive Policy Ansatz for Hybrid Discrete-Continuous Action Spaces

Recently, a number of studies have considered the problem of simultaneous discrete/continuous control using RL (Kulkarni et al., 2016; Fan et al., 2019; Wei et al., 2018; Hausknecht and Stone, 2016; Delalleau et al., 2019; Bester et al., 2019; Xiong et al., 2018; Fan et al., 2019; Wei et al., 2018; Neunert et al., 2020). Following notation of Ref. (Masson et al., 2016), we describe the RL problem within the framework of parametrized-action Markov decision processes (PAMDPs). The major difference, compared to ordinary MDPs, is the definition of the action space: $\mathcal{A} = \mathcal{A}^{\mathrm{d}} \otimes \mathcal{A}^{\mathrm{c}} = \bigcup_{a^{\mathrm{d}} \in \mathcal{A}^{\mathrm{d}}, a^{\mathrm{c}} \in \mathcal{A}^{\mathrm{c}}} (a^{\mathrm{d}}, a^{\mathrm{c}}),\ \mathcal{A}^{\mathrm{d}} = \{H_j\}_{j=1}^{|\mathcal{A}^{\mathrm{d}}|},\ \mathcal{A}^{\mathrm{c}} = [0,1]$, where $|\mathcal{A}^{\mathrm{d}}|$ denotes the cardinality of the discrete action set. As before, the state space contains all possible sequences of actions, and the reward is the (negative) energy density of the quantum state, given once at the end of the protocol.

In this section, we present a unified continuous-discrete quantum control algorithm, called RL-QAOA, based on a hybrid policy which optimizes simultaneously the discrete and continuous degrees of freedom. The policy can be decomposed as a product of two coupled auxiliary policies – one for the continuous actions, $\pi^{\mathrm{c}}_{\boldsymbol{\theta}}$, and the other for the discrete actions, $\pi^{\mathrm{d}}_{\boldsymbol{\theta}}$:

$$\pi_{\boldsymbol{\theta}}(\tau) = \pi^{\mathrm{c}}_{\boldsymbol{\theta}}\left(\tau^{\mathrm{c}}\right) \pi^{\mathrm{d}}_{\boldsymbol{\theta}}\left(\tau^{\mathrm{d}}\right), \tag{7}$$

where $\tau^{\nu} = (a^{\nu}_1, \ldots, a^{\nu}_q),\ \nu \in \{\mathrm{c}, \mathrm{d}\}$ defines the discrete/continuous subsequence of actions in each trajectory of length q. Denoting, as before, the RL state by $s_j = (a_1, \cdots, a_{j-1})$ with the hybrid action $a_i = (a^{\mathrm{c}}_i, a^{\mathrm{d}}_i)$, we define a generalized continuous/discrete autoregressive model for the policy, following Eq. (5). Adopting the short-hand notation $\pi^{\nu}_{\boldsymbol{\theta}}\left(a^{\nu}_j \mid s_j\right) = \pi^{\nu}_{\boldsymbol{\theta}}\left(a^{\nu}_j \mid a_1, \cdots, a_{j-1}\right)$, the policy can be written as

$$\pi_{\boldsymbol{\theta}}\left(a_1, a_2, \cdots, a_q\right) = \prod_{j=1}^{q} \pi^{\mathrm{d}}_{\boldsymbol{\theta}}\left(a^{\mathrm{d}}_j \mid s_j\right) \pi^{\mathrm{c}}_{\boldsymbol{\theta}}\left(a^{\mathrm{c}}_j \mid s_j, a^{\mathrm{d}}_j\right). \tag{8}$$

As expected, at every step j, the action a^{c}_j is sampled from a continuous distribution, whose parameters depend on the discrete action a^{d}_j selected at the same step j. This is natural, since different discrete actions may require different corresponding continuous distribution parameters κ, ξ.

Additionally, similar to CD-QAOA, we impose a further restriction: no discrete action can occur in the trajectory consecutively. We use a Sigmoid-Gaussian distribution to bound the samples for the continuous actions, and normalize the durations $\alpha_j \propto a^{\mathrm{c}}_j \sim \pi^{\mathrm{c}}_{\boldsymbol{\theta}}$ to fix the total protocol duration such that $\sum_{j=1}^{q} \alpha_j = T$; using the Beta distribution instead results in a similar performance [cf. Fig. 4].

3.2. Deep Autoregressive Policy Network

We implement the policy ansatz variationally, using a deep neural network [a.k.a. policy network]. In Fig. 2, we show a cartoon of the model for illustration purposes. The network consists of base layers with intermediate output $\boldsymbol{y}$, followed by three independent head layers with outputs $\boldsymbol{z}^{p}, \boldsymbol{z}^{\kappa}, \boldsymbol{z}^{\xi}$, respectively. The three heads learn the discrete probability distribution π^{d}, and the parameters $\kappa, \xi \in \mathbb{R}^{+}$ which define the continuous probability distribution π^{c}. Each head outputs a vector of size $|\mathcal{A}^{\mathrm{d}}|$ – so that the model can learn a set (κ, ξ) for every distinct discrete action. Notice that each head output depends on the joint base layer parameters $(\boldsymbol{W}, \boldsymbol{b})$, but not on the parameters $(\boldsymbol{V}, \boldsymbol{c})$ of any of the other two heads; thus, the base layers are shared by all three heads. In practice, we find that a base layer, comprised of two hidden layers, can already achieve a good performance; one can in principle add more layers for enhanced expressivity.

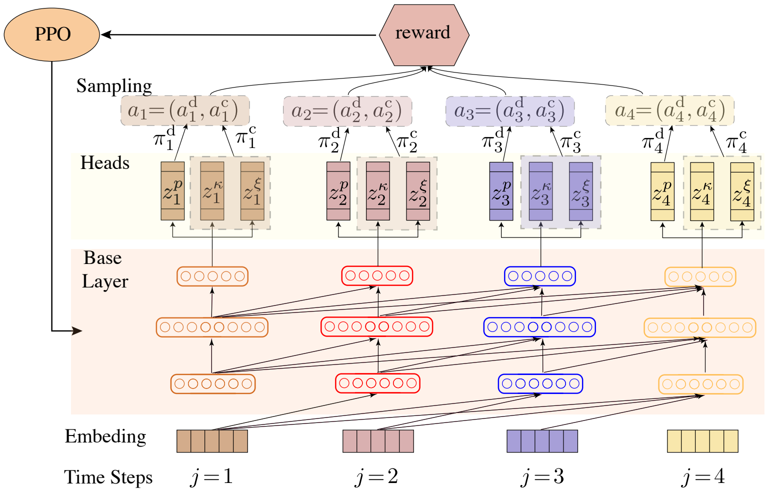


Figure 2: Schematic representation of RL-QAOA and the deep autoregressive network for $q = 4$ (see text). The time step j also corresponds to the gate index. The policy network is composed of (i) an embedding layer to encode the continuous and discrete actions as input. (ii) The base layer implements the causal autoregressive structure (see arrows). (iii) The heads are three-fold, one for the discrete distribution parameters, and two for the continuous distribution parameters. A batch of actions are sampled to evolve the quantum state and compute the negative energy density as a reward. Proximal hybrid Policy Optimization (PPO) is used to update the policy network. The pseudocode for RL-QAOA is shown in Algorithm 1.

The above description focuses on a single episode step j out of a total of q steps in an episode. The autoregressive feature of the ansatz can then be built in, by allowing the outputs of the base layers from previous steps to become inputs into the layers at subsequent episode steps [Fig. 2].

Let us denote the input to the autogressive network by $(x_1, x_2, \cdots, x_q)$, and the weights and bias parameters of the base layer by $W_j \in \mathbb{R}^{d_{\mathrm{h}} \times (j-1)|\mathcal{A}^{\mathrm{d}}|}$ and $b_j \in \mathbb{R}^{d_{\mathrm{h}}}$, respectively, where d_{h} is the hidden dimension. Then, the intermediate output $(y_1, y_2, \cdots, y_q)$ of the base layer reads as

$$y_j = g(W_j x_{<j} + b_j), \quad j = 1, 2, \cdots q, \tag{9}$$

where $x_{<j} = (x_{j-1}, \cdots, x_1)^T \in \mathbb{R}^{(j-1)|\mathcal{A}^{\mathrm{d}}|}$ denotes the input of all previous steps preceding step j; for $j=1$, we set $W_j x_{<j} + b_j = b_j$.[4] We use ReLU nonlinearities $g(\cdot)$.

The output of the base layer $(y_1, y_2, \cdots, y_q)$ can be viewed as an input to the three-head layer. The three-head layer contains three heads with independent weights $V_j^p, V_j^\kappa, V_j^\xi \in \mathbb{R}^{|\mathcal{A}^{\mathrm{d}}| \times j d_{\mathrm{h}}}$ and biases $c_j^p, c_j^\kappa, c_j^\xi \in \mathbb{R}^{|\mathcal{A}^{\mathrm{d}}|}$. The three-head layer output, $(z_1, z_2, \cdots, z_q)$, are the parameters for the discrete and continuous distributions: z_j^p are the categorical distribution parameters; z_j^κ and z_j^ξ are the two parameters for the sigmoid-Gaussian distribution [cf. App. C.1]:

$$z_j^p = \log\left(\mathrm{SoftMax}(V_j^p y_{\leq j} + c_j^p)\right), \qquad z_j^\kappa = V_j^\kappa y_{\leq j} + c_j^\kappa, \quad z_j^\xi = \exp\left(V_j^\xi y_{\leq j} + c_j^\xi\right), \tag{10}$$

[4] In practice, implementing the autoregressive constraint $_{<j}$ can be achieved using masks (one for each set of weights).

where $y_{\leq j} = (y_j, \cdots, y_1)^T \in \mathbb{R}^{j d_\mathrm{h}}$, and d_h is the hidden layer width.[5] To define a categorical distribution, we use a SoftMax[6] nonlinearity: $\mathrm{SoftMax}(v)[i] = \exp(v[i]) / \sum_{k=1}^{|\mathcal{A}^\mathrm{d}|} \exp(v[k])$, where $v = V_j^p y_{\leq j} + c_j^p \in \mathbb{R}^{|\mathcal{A}^\mathrm{d}|}$, and $[\cdot]$ takes the index; we learn the log-probability to achieve a resolution over a few orders of magnitude, and to stabilize the learning process.

We apply ancestral sampling to draw actions from the autoregressive policy. Starting from the heads layer at step $j = 1$, we first sample $a_1^\mathrm{d} \sim \pi(a_1^\mathrm{d}) = \mathrm{Categorical}(\exp(z_1^p))$; we use the sampled discrete action a_1^d to look up the corresponding parameters $\kappa = z_1^\kappa[a_1^\mathrm{d}]$ and $\xi = z_1^\xi[a_1^\mathrm{d}]$ [7] for the continuous action distribution. Then we sample the duration $a_1^\mathrm{c} \sim \pi(a_1^\mathrm{c}|a_1^\mathrm{d}) = \mathcal{SN}\left(z_1^\kappa[a_1^\mathrm{d}], (z_1^\xi[a_1^\mathrm{d}])^2\right)$. The sampling output is passed as an input at the second step $j = 2$. To do this, we use an embedding[8] for $(a_1^\mathrm{d}, a_1^\mathrm{c})$ represented by the variable x_1, where $x_1[i] = a_1^\mathrm{c}$ if $i = a_1^\mathrm{d}$, and $x_1[i] = 0$ otherwise. Going on, we repeat the process: we sample successive actions $a_2^\mathrm{d}, a_2^\mathrm{c} \sim \pi(a_2^\mathrm{d}|x_1), \pi(a_2^\mathrm{c}|x_1, a_2^\mathrm{d})$. The sampling, or forward pass, through the network is then repeated q times, until we reach the end of the episode; thus, at step j we have $a_j^\mathrm{d}, a_j^\mathrm{c} \sim \pi(a_j^\mathrm{d}|x_{<j}), \pi(a_j^\mathrm{c}|x_{<j}, a_j^\mathrm{d})$. This gives the trajectory τ of mixed discrete-continuous actions. Note that the time complexity of the process is $\mathcal{O}(q \times |\mathcal{A}^\mathrm{d}|)$.

3.3. Proximal Hybrid Policy Optimization

The set of all weights and biases, $\boldsymbol{\theta} = \{W_j, b_j, V_j^p, V_j^\kappa, V_j^\xi, c_j^p, c_j^\kappa, c_j^\xi\}_{j=1}^q$, defines the learnable parameters of the autoregressive policy network. We now discuss how to compute the policy gradients and define an update rule for $\boldsymbol{\theta}$.

Our goal is to maximize the RL objective within the trust region (Schulman et al., 2015) for the continuous and discrete policy:

$$\mathbb{E}_\tau\left[\frac{\pi_{\boldsymbol{\theta}}(\tau)}{\pi_{\boldsymbol{\theta}_t}(\tau)} A_{\boldsymbol{\theta}_t}(\tau)\right], \quad \text{subject to} \quad \mathbb{E}_\tau\left[\mathrm{D}_{\mathrm{KL}}\left[\pi^\nu_{\boldsymbol{\theta}_t}(\cdot), \pi^\nu_{\boldsymbol{\theta}}(\cdot)\right]\right] \leq \delta^\nu, \tag{11}$$

where $\mathbb{E}_\tau[\,\cdot\,]$ is a shorthand notation for $\mathbb{E}_{\tau=(a_1,\cdots,a_q)\sim\pi_{\boldsymbol{\theta}_t}}[\,\cdot\,]$. The Kullback–Leibler (KL) divergence is defined as $\mathrm{D}_{\mathrm{KL}}(\pi^\mathrm{c}_{\boldsymbol{\theta}_t}, \pi^\mathrm{c}_{\boldsymbol{\theta}}) = \int_{x\in\mathcal{A}^\mathrm{c}} \pi^\mathrm{c}_{\boldsymbol{\theta}_t}(x) \log\left(\frac{\pi^\mathrm{c}_{\boldsymbol{\theta}_t}(x)}{\pi^\mathrm{c}_{\boldsymbol{\theta}}(x)}\right) \mathrm{d}x$, and similarly for $\nu = \mathrm{d}$; δ^ν defines a constraint on the size of the discrete/continuous policy updates in distribution space. Here, $\boldsymbol{\theta}_t$ denotes the parameters before the update, usually the parameters from the last update, i.e. t-th iteraion; $A_{\boldsymbol{\theta}_t}(\tau) = R(\tau) - b$ is the advantage function – the return (negative energy density) for a given trajectory w.r.t. the baseline b.

In practice, we utilize a clipped surrogate RL objective (Schulman et al., 2017) with two clipping parameters ϵ^ν. The idea is to update the continuous and discrete policies adaptively using different ϵ^ν during policy optimization. This allows for the discrete policy $\pi^\mathrm{d}_{\boldsymbol{\theta}}$ to change more quickly/more slowly as compared to the continuous policy $\pi^\mathrm{c}_{\boldsymbol{\theta}}$. Hence, the hybrid PPO RL objective reads as

$$\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_\tau\Big[\mathcal{G}^\mathrm{d}(\tau^\mathrm{d}; \boldsymbol{\theta}, \epsilon^\mathrm{d}) + \mathcal{G}^\mathrm{c}(\tau^\mathrm{c}; \boldsymbol{\theta}, \epsilon^\mathrm{c})\Big] + \beta_S^{-1}(\mathcal{S}^\mathrm{d} + \mathcal{S}^\mathrm{c}), \tag{12}$$

[5] Note that here we are able to use the ”=” sign because the previous layer of operation has already filtered out the ”=” sign for those steps.

[6] Note that this function is not operated element-wise like the others; it is applied on the whole vector of dimension $|\mathcal{A}^\mathrm{d}|$).

[7] Here, $[a_1^\mathrm{d}]$ means taking the component by index.

[8] The embedding can be viewed as one-hot encoding of the discrete action a_1^d but multiplied by the continuous action value a_1^c.

with

$$\mathcal{G}^{\nu}(\tau^{\nu};\boldsymbol{\theta},\epsilon^{\nu}) = \min\left\{\rho^{\nu}_{\boldsymbol{\theta}}(\tau^{\nu})A^{\nu}_{\boldsymbol{\theta}_t}(\tau^{\nu}),\ \mathrm{clip}\left(\rho^{\nu}_{\boldsymbol{\theta}}(\tau^{\nu}),1-\epsilon^{\nu},1+\epsilon^{\nu}\right)A^{\nu}_{\boldsymbol{\theta}_t}(\tau^{\nu})\right\}, \tag{13}$$

where $\rho^{\nu}_{\boldsymbol{\theta}}(\tau^{\nu}) = \frac{\pi^{\nu}_{\boldsymbol{\theta}}(\tau^{\nu})}{\pi^{\nu}_{\boldsymbol{\theta}_t}(\tau^{\nu})}$ is the importance weight ratio of two policies associated with trajectory τ^{ν}. The clip function, defined as $\mathrm{clip}(\rho,x,y) = \max\big(\min\left(\rho,x\right),y\big)$ sets the value of $\rho_{\boldsymbol{\theta}}$ to be within the interval $[x,y]$, and constrains the likelihood ratio from Eq. (11) to the range $[1-\epsilon,1+\epsilon]$. The entropy terms [right-most part of Eq. (12)] are discussed below. Our goal is to find those parameters $\boldsymbol{\theta}$ which maximize $\mathcal{J}(\boldsymbol{\theta})$.

To understand the hybrid PPO algorithm, consider two limiting cases first. In the extreme case when $\epsilon^{\mathrm{d}} \to 0$, i.e. the discrete policy $\pi^{\mathrm{d}}_{\boldsymbol{\theta}}$ is kept fixed, our algorithm reduces to PG-QAOA. On the other hand, when $\epsilon^{\mathrm{c}} \to 0$, the continuous policy is kept fixed; if this fixed policy additionally corresponds to the greedy "expert policy" defined by the Powell optimizer, the algorithm is reduced to CD-QAOA. In this sense, for finite values of $\epsilon^{\mathrm{c}},\epsilon^{\mathrm{d}} > 0$, RL-QAOA can be viewed as a smooth interpolation between PG-QAOA and CD-QAOA.

In order to incentivize the agent to explore the action space during the early stages of training, we also added entropy to the RL objective, cf. Eq. (12). The entropy for a discrete/continuous policy is defined as $\mathcal{S}^{\mathrm{d}}(\pi^{\mathrm{d}}) = -\sum_{x\in\mathcal{X}}\pi^{\mathrm{d}}(x)\log\pi^{\mathrm{d}}(x)$ or $\mathcal{S}^{\mathrm{c}}(\pi^{\mathrm{c}}) = -\int_{x\in\mathcal{X}}\pi^{\mathrm{c}}(x)\log\pi^{\mathrm{c}}(x)\mathrm{d}x$, respectively. The coefficient β_S^{-1} in Eq. (12) defines an effective temperature, which we anneal with increasing the number of iterations. It is easy to see that the total entropy $\mathcal{S} = \mathcal{S}^{\mathrm{d}} + \mathcal{S}^{\mathrm{c}}$ associated with the hybrid policy consists of both discrete $\mathcal{S}^{\mathrm{d}} = \sum_{j=1}^{q}\mathbb{E}_{a_{<j}\sim\pi_{\boldsymbol{\theta}}}\mathcal{S}^{\mathrm{d}}\big(\pi^{\mathrm{d}}_{\boldsymbol{\theta}}(\,\cdot\,|a_{<j})\big)$, and continuous $\mathcal{S}^{\mathrm{c}} = \sum_{j=1}^{q}\mathbb{E}_{a_{<j}\sim\pi_{\boldsymbol{\theta}},a^{\mathrm{d}}_j\sim\pi^{\mathrm{d}}_{\boldsymbol{\theta}}}\mathcal{S}^{\mathrm{c}}\big(\pi^{\mathrm{c}}_{\boldsymbol{\theta}}(\,\cdot\,|a_{<j},a^{\mathrm{d}}_j)\big)$ contribution. The RL agent has to maximize the total expected return while also maximizing the entropy associated with the policy.

In RL, there are two common ways to incorporate entropy in practice (Levine, 2018): (i) whenever one can compute a closed-form expression for the entropy, entropy is added as a separate term to the objective which can be thought of as entropy regularization. Note that it is the autoregresssive structure that makes it possible to obtain the exact value for the entropy $\mathcal{S}^{\mathrm{d}}$: for $\pi^{\mathrm{d}}_{\boldsymbol{\theta}}(\,\cdot\,|a_{<j}) = \mathrm{Categorical}\Big(\exp\Big(z^p_j\Big)\Big)$, the entropy is $\mathcal{S}^{\mathrm{d}}\big(\pi^{\mathrm{d}}_{\boldsymbol{\theta}}(\,\cdot\,|a_{<j})\big) = -\sum_{k=1}^{|\mathcal{A}^{\mathrm{d}}|} z^p_j[k]\cdot\exp\Big(z^p_j[k]\Big)$. (ii) Often times it is not always possible to compute the value for the entropy, since the expression is not analytically tractable; in such cases, the maximum entropy formulation (Haarnoja et al., 2018b,a, 2017) still allows us to add to the reward a empirical estimate of the entropy, known as an entropy bonus: $R^{\mathrm{c}}(\tau) \leftarrow R^{\mathrm{c}}(\tau) + \beta_S^{-1}\,\mathbb{E}_{a^{\mathrm{c}}\sim\pi^{\mathrm{c}}_{\boldsymbol{\theta}}}\big[-\log\pi^{\mathrm{c}}_{\boldsymbol{\theta}}\big]$. In this study, we add an entropy bonus to take into account the entropy of the continuous policy π^{c}.

4. Applications and Noise Models

4.1. Quantum Ising Model

To test the performance of RL-QAOA, we investigate the ground state preparation problem for a system of N interacting qubits (i.e. spin-$1/2$ degrees of freedom), described by the Ising Hamiltonian introduced in Eq. (3):

$$H = H_1 + H_2, \qquad H_1 = \sum_{i=1}^{N} J S^z_{i+1}S^z_i + h_z S^z_i, \quad H_2 = \sum_{i=1}^{N} h_x S^x_i,$$

We use periodic boundary conditions and work in the zero momentum sector of positive parity, which contains the antiferromagnetic ground state. We emphasize that this model is non-integrable, i.e., it does not have an extensive number of local integrals of motion; as a consequence, no closed-form analytical description is known for its eigenstates and eigenenergies. Moreover, the lack of integrability results in chaotic quantum dynamics. This makes manipulating it in the presence of noise particularly challenging.

In the following, $J = 1$ sets the energy unit, $h_z/J = 0.4523$ and $h_x/J = 0.4045$. In the thermodynamic limit, $N \to \infty$, these parameters are close to the critical line of the model, where a quantum phase transition occurs in the ground state between an antiferromagnet and a paramagnet; for the finite system sizes we can simulate, the critical behavior is smeared out over a small finite region. In Ref. (Matos et al., 2021), using QAOA, it was shown that this region of parameter space appears most challenging in the noise-free system.

We initialize the system in the z-polarized product state $|\psi_i\rangle = |\uparrow \cdots \uparrow\rangle$, and aim to prepare the ground state of H. We use the negative energy density $-\mathcal{E} = -E/N$ as a reward for the RL agent, cf. Eq. (2), which is an intensive quantity as the number of qubits N increases. In this study, we are mostly interested in exploring the behavior of the system subject to various kinds of noise/uncertainty. Our primary focus is quantifying the effects of noise on the achievable fidelity, w.r.t. the noise-free values. We deliberately select a fixed duration of $JT = 10$ far from the adiabatic regime, such as to exhibit the benefits of the CD-QAOA ansatz over QAOA [cf. App. D].

We point out that, working at a fixed duration T, it is not always possible to achieve high-fidelity ground states. This is easy to see for decoupled qubits, where the magnitude of the spin precession frequency on the Bloch sphere (so-called Larmor precession frequency) is set by the fixed strength of the magnetic field $(h_x, 0, h_z)$: hence, fixing the total protocol duration T, it may be physically impossible to reach the target state in the allotted time. This behavior leads to the notion of the quantum speed limit (QSL) – the minimum time required to prepare the ground state with unit fidelity.

4.2. Spin-1 Heisenberg Model

To demonstrate that RL-QAOA applies equally well to a system other than the Ising model, we consider the noisy state preparation in the spin-1 Heisenberg chain, described by the Hamiltonian

$$H = H_1 + H_2, \qquad H_1 = J \sum_{j=1}^{N} (S^x_{j+1} S^x_j + S^y_{j+1} S^y_j), \quad H_2 = \Delta \sum_{j=1}^{N} S^z_{j+1} S^z_j. \tag{14}$$

Here, J is the interaction in the xy-plane and Δ characterizes the anisotropy. The model features a rich ground state phase diagram, including topological and long-range ordered phases (Chen et al., 2003; Pollmann et al., 2010; Langari et al., 2013). Such spin-1 systems present natural models to simulate on qutrit quantum computing devices (Blok et al., 2021; Ramasesh et al., 2019).

We initialize the system in the antiferromagnetic initial state $|\psi_i\rangle = \mathcal{P}\ |\uparrow\downarrow\uparrow\downarrow \cdots\rangle$, with $\mathcal{P}$ the projector onto the zero-momentum sector of positive parity; we target the ground state of the Heisenberg model in the presence of noise at $\Delta/J = -0.5$, and fix the total protocol duration to $JT = 3$.

4.3. Three Noise Models

When operating present-day quantum devices, one is confronted with various sources of uncertainty. Since the exact form and details depend on the peculiarities and particularities of the underlying experimental platform, it is desirable to construct algorithms capable of learning such details without extra human input. In this study, our RL agent learns in a simulator. To mimic the diversity of uncertain processes that can occur, we consider three types of noise.

4.3.1. Classical Measurement Gaussian Noise

Noise naturally occurs due to imperfect measurements. For instance, the measurement signal is often present in the form of currents and voltages, whose values can only be determined within the resolution of the measurement apparatus. In practice, experimentalists perform a large number of measurements and average the result in the end to obtain an estimate for the value of an observable. By the central limit theorem, in the limit of large sample sizes, the statistics of the measurement data is approximated by a Gaussian distribution. To model this behavior, we use small Gaussian noise to add uncertainty in the reward signal: $\mathcal{E}_\gamma(\{\alpha_i\}_{i=1}^q, \tau^{\mathrm{d}}) = \mathcal{E}(\{\alpha_i\}_{i=1}^q, \tau^{\mathrm{d}}) + \epsilon_\gamma$, where $\epsilon_\gamma \sim \mathcal{N}(0, \gamma^2)$.

4.3.2. Quantum Measurement Noise

In quantum mechanics, there is another intrinsic kind of noise, which arises due to the quantum nature of the controlled system. Consider the evolved state $|\psi(T)\rangle = U(\{\alpha_j\}_{j=1}^q, \tau)|\psi_i\rangle$ at the end of the protocol. The expected measurement for the energy density $\mathcal{E} = N^{-1}\langle\psi(T)|H|\psi(T)\rangle$ is obtained within a *quantum* uncertainty, $\Delta\mathcal{E} = N^{-1}\sqrt{\langle\psi(T)|H^2|\psi(T)\rangle - \langle\psi(T)|H|\psi(T)\rangle^2}$, set by the energy variance in the final state. In the limit of a large number of measurements, quantum noise can be simulated using a Gaussian distribution $\mathcal{E}_Q(\{\alpha_i\}_{i=1}^q, \tau^{\mathrm{d}}) = \mathcal{E}(\{\alpha_i\}_{i=1}^q, \tau^{\mathrm{d}}) + \epsilon_Q$, where $\epsilon_Q \sim \mathcal{N}(0, \Delta\mathcal{E}^2)$. Note that the width of the Gaussian depends on the final state $|\psi(T)\rangle$: in the early stages of training, $|\psi(T)\rangle$ is typically far away from any of the eigenstates of H; therefore, the energy variance $\Delta\mathcal{E}$ will be large and finite. However, towards the later training stages, when the agent learns to prepare a state close to the target ground state, the energy variance will go down. Hence, one can think of the quantum noise as a Gaussian noise with a time-dependent strength.

4.3.3. Noise arising from Gate Rotation Errors

Finally, we also consider the uncertainty in implementing the unitaries U_i. We focus on gate rotation errors (Sung et al.), caused by imperfections in the durations α_i: $\mathcal{E}_\delta(\{\alpha_i\}_{i=1}^q, \tau) = \mathcal{E}(\{\alpha_i + \epsilon_i\}_{i=1}^q)$, where $\epsilon_i \sim \mathcal{N}(0, \delta^2)$. This defines a simplified error model for coherent control, an important source of errors in present-day state-of-the-art quantum computing hardware (Arute et al., 2019), and which is especially pertinent to quantum computers which are utilized frequently but calibrated only periodically.

5. Numerical Experiments and Results

To evaluate the performance of the trained agent, we eliminate the uncertainty associated with the probabilistic nature of the policy: we take the discrete action which maximizes the categorical distribution π^{d}, and only keep the mean of the continuous distribution π^{c}, setting its width to zero. This defines a natural greedy policy to test the ability of the RL agent.

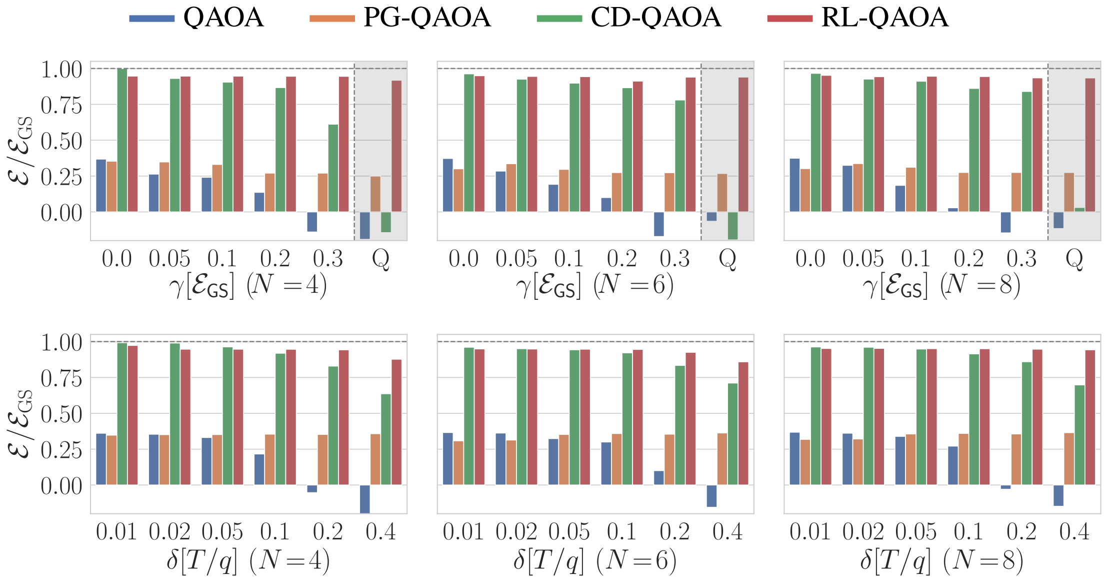


Figure 3: Ising spin-$1/2$ model: energy minimization in the Ising model against different noise levels with circuit depths $p=q/2=4$ and protocol duration $JT=10$ for four different optimization methods: QAOA, PG-QAOA, CD-QAOA, RL-QAOA. The initial and target states are $|\psi_i\rangle = |\uparrow\cdots\uparrow\rangle$ and $|\psi_*\rangle = |\psi_{\mathrm{GS}}(H)\rangle$ for $h_z/J = 0.4523$ and $h_x/J = 0.4045$. The alternating unitaries for conventional QAOA and PG-QAOA are generated by $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2\}$; for CD-QAOA and RL-QAOA, we extend this set using adiabatic gauge potential terms to $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2; Y, X|Y, Y|Z\}$. The system sizes are $N=4,6,8$. The continuous policies in PG-QAOA and RL-QAOA are both parametrized by Sigmoid-Gaussian distributions.

We performed a number of numerical experiments to study the effect of the noise on the performance of the four algorithms QAOA, PG-QAOA, CD-QAOA and RL-QAOA, for the three different sources of uncertainty: classical and quantum measurement noise, and gate rotation noise. We vary both the noise strength, and we look at three different system sizes for two protocol durations each. The results of these experiments can be summarized as follows.

Figure 3 shows the best achievable energy at a protocol duration $JT = 10$ against different noise types and system sizes of the Ising model: the top row shows data for various measurement noise strengths, with the shaded area marking the special case of quantum noise; the noise strength is measured in percentages of the achievable ground state energy density: e.g., a noise strength of $\gamma = 0.3$ corresponds to an average deviation from the actual energy of about 30%. The bottom row displays the results when varying the gate noise strength. Here, the noise strength is defined as a percentage of the mean gate duration T/q. The three columns correspond to system sizes $N = 4$ (left), $N = 6$ (middle) and $N = 8$ (right).

When $T < T_{\mathrm{QSL}}$ is chosen below the QSL, we find that QAOA and PG-QAOA fail to reach the ground state in the time allotted, as a result of having an overconstrained control space $\mathcal{A}^{\mathrm{d}} = \{H_0, H_1\}$. Nonetheless, the noise-robust character of PG-QAOA becomes pronounced at increased values of the noise strength. Since the initial quantum state is far away from the target ground state, the best ratio E/E_{GS} found by QAOA can even be negative. The $JT = 10$ duration exhibits the advantage of using the generalized QAOA ansatz brought in by CD-QAOA: suitably enlarging the discrete action space $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2, Y, X|Y, Y|Z\}$ unlocks paths in Hilbert space which are

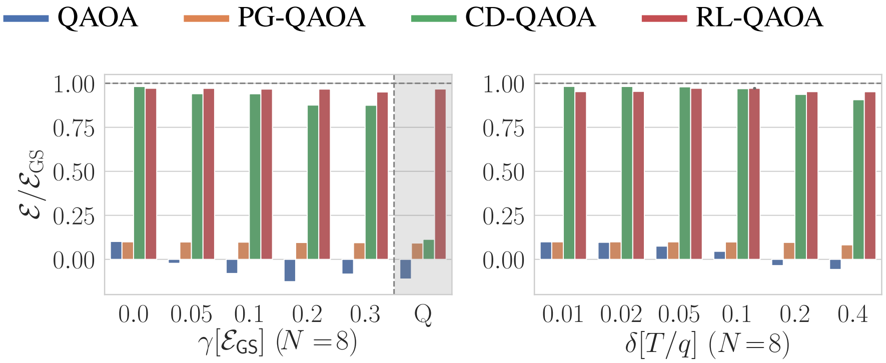


Figure 4: Heisenberg spin-1 model: energy minimization for four different optimization methods: QAOA, PG-QAOA, CD-QAOA, RL-QAOA. The left column shows measurement noise (including quantum), and the right column displays gate noise. The alternating unitaries for conventional QAOA and PG-QAOA are generated by $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2\}$; for CD-QAOA and RL-QAOA, we extend this set using adiabatic gauge potential terms to $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2, Z, X|X; Y, XY, YZ, X|Y{-}XY, Y|Z{-}YZ\}$. The model parameters are $\Delta/J = -0.5$, $T = 3$, and the system size is $N = 8$. The continuous policies in PG-QAOA and RL-QAOA are both parametrized by Beta distributions.

inaccessible to QAOA. Hence, CD-QAOA and RL-QAOA find the largest rewards in the noise-free case. A large noise strength reduces visibly the ability of CD-QAOA to find the ground state, with the performance being particularly bad for quantum measurement noise (Q). However, the hybrid policy optimizer allows RL-QAOA to emerge as a noise-robust algorithm, agnostic to the source of noise applied to the system.

In Fig. 4, we show the same comparison for the anisotropic Heisenberg chain of $N = 8$ qutrits. Clearly, RL-QAOA achieves much better results than QAOA and PG-QAOA, due to the suitably chosen enhanced space of actions. With the exception of the noise-free case, we find that RL-QAOA outperforms CD-QAOA on the measurement noise problems (left panel), and performs on par with it for the gate noise (right panel). Thus, RL-QAOA shows superior performance in the presence of noise also for the Heisenberg model.

6. Conclusion and Outlook

In summary, we presented RL-QAOA – a versatile and noise-robust quantum control algorithm based on the QAOA variational ansatz. The algorithm inherits valuable features from its ancestors: (i) the noise-robust property of PG-QAOA allows us to find optimal durations probabilistically. (ii) the generalized QAOA ansatz of CD-QAOA makes it possible to select the order in which a set of unitaries appears in the control sequence. While we focused on physically motivated unitaries, we emphasize that the ansatz is completely general and applicable to a large variety of unitaries/quantum gate sets useful for both theoretical and experimental studies. We had to modify these "ancestors" accordingly: in PG-QAOA we introduced a mechanism to fix the total protocol duration and introduced a stochastic policy based on the compactly supported Beta function; in CD-QAOA we changed the low-level optimizer to gradient-free Powell, as opposed to the gradient-based SLSQP which did not give a reasonable performance in the presence of noise. RL-QAOA extends PG-QAOA and CD-QAOA with both the use of a generalized autoregressive architecture which incorporates the parameters of the continuous policy, and the derivation of an extension of Proximal Policy Optimization applicable to hybrid continuous-discrete policies.

We tested the performance of RL-QAOA using the unitary dynamics of quantum Ising and Heisenberg chains subject to various sources of noise: classical and quantum measurement noise as well as uncertainty leading to errors in the application of quantum unitary gates. In particular, we demonstrated that RL-QAOA successfully outperforms its ancestors in the highly-constrained non-adiabatic regime, irrespective of the noise model selected. Thus, RL-QAOA is not only noise-robust but also agnostic to the physical source of noise. This opens up the exciting possibility of using machine learning to 'learn' the particularities of noisy experimental environments, which often depend on the chip architecture and can even change in the course of exploitation. However, the presented results are obtained using numerical simulations based on certain theoretical assumptions; it remains to test the performance of RL-QAOA on realistic noisy intermediate-scale quantum computing devices.

The RL-QAOA is a versatile method that can be extended along several directions. For instance, the current version of RL-QAOA defines a fixed sequence/protocol length. However, the algorithm is versatile enough to accommodate a variable length of the protocols after a slight modification. To do so, one can simply add a "stop" action to the discrete action set $\mathcal{A}^d$. If the agent happens to choose the stop action, then the episode comes to an end immediately and we measure the energy of the evolved quantum state.

There also exist a number of exciting alternatives for the policy network architecture to explore. Although it has to incorporate temporal causality, notice that the architecture is not limited to the autoregressive choice used in this study; e.g., it can be generalized to a recurrent neural network (RNN), a Long Short Term Memory (LSTM) network, or a transformer with the attention mechanism (Vaswani et al., 2017) and all its modern variants (Kitaev et al., 2020; Choromanski et al., 2020; Wang et al., 2020; Tay et al., 2020). In the present study, we chose the autoregressive network for its sheer simplicity. Moreover, the continuous policy head can be generalized to capture distributions with more than two modes using the normalizing flow method, which would additionally boost the expressivity of the policy (Tang and Agrawal, 2018).

Acknowledgments

We wish to thank Vitchyr Pong for valuable discussions. This work was partially supported by the Department of Energy under Grant No. DE-AC02-05CH11231 and No. DE-SC0017867 (L.L., J.Y.), and by the National Science Foundation under the NSF QLCI program through grant number OMA-2016245 (L.L.). M.B. was supported by the Bulgarian National Science Fund within National Science Program VIHREN, contract number KP-06-DV-5, and the Marie Sklodowska-Curie grant agreement No 890711.

We used the Powell method (Powell, 1964) implemented in SciPy (Virtanen et al., 2020) for the QAOA solver; we used W&B (Biewald, 2020) to organize and analyze the experiments. The reinforcement learning networks are implemented in NumPy(Harris et al., 2020), and TensorFlow (Abadi et al., 2015) and TensorFlow Probability (Dillon et al., 2017); the quantum systems are simulated in Quspin (Weinberg and Bukov, 2017, 2019). We thank Berkeley Research Computing (BRC) for providing the computational resources.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Francisco Albarrán-Arriagada, Juan C Retamal, Enrique Solano, and Lucas Lamata. Measurement-based adaptation protocol with quantum reinforcement learning. *Phys. Rev. A*, 98:042315, 2018. doi: 10.1103/PhysRevA.98.042315. URL https://link.aps.org/doi/10.1103/PhysRevA.98.042315.

Abhinav Anand, Matthias Degroote, and Alan Aspuru-Guzik. Natural evolutionary strategies for variational quantum computation. *Machine Learning: Science and Technology*, mar 2021. doi: 10.1088/2632-2153/abf3ac. URL https://doi.org/10.1088%2F2632-2153%2Fabf3ac.

Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. URL https://www.nature.com/articles/s41586-019-1666-5.

Moritz August and José Miguel Hernández-Lobato. Taking gradients through experiments: Lstms and memory proximal policy optimization for black-box quantum control. In *International Conference on High Performance Computing*, pages 591–613. Springer, 2018. URL https://arxiv.org/abs/1802.04063.

Marc G. Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C. Machado, Subhodeep Moitra, Sameera S. Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020. doi: 10.1038/s41586-020-2939-8. URL https://doi.org/10.1038/s41586-020-2939-8.

Craig J Bester, Steven D James, and George D Konidaris. Multi-pass q-networks for deep reinforcement learning with parameterised action spaces. *arXiv preprint arXiv:1905.04388*, 2019.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL https://www.wandb.com/. Software available from wandb.com.

M. S. Blok, V. V. Ramasesh, T. Schuster, K. O'Brien, J. M. Kreikebaum, D. Dahlen, A. Morvan, B. Yoshida, N. Y. Yao, and I. Siddiqi. Quantum information scrambling on a superconducting qutrit processor. *Physical Review X*, 11(2), apr 2021. doi: 10.1103/physrevx.11.021010. URL https://doi.org/10.1103%2Fphysrevx.11.021010.

Annabelle Bohrdt, Christie S Chiu, Geoffrey Ji, Muqing Xu, Daniel Greif, Markus Greiner, Eugene Demler, Fabian Grusdt, and Michael Knap. Classifying snapshots of the doped hubbard model with machine learning. *Nature Physics*, 15(9):921–924, 2019.

Adrien Bolens and Markus Heyl. Reinforcement learning for digital quantum simulation. *arXiv preprint arXiv:2006.16269*, 2020. URL https://arxiv.org/abs/2006.16269.

Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Hybrid quantum-classical algorithms for approximate graph coloring. *arXiv preprint arXiv:2011.13420*, 2020.

Marin Bukov. Reinforcement learning for autonomous preparation of floquet-engineered states: Inverting the quantum kapitza oscillator. *Physical Review B*, 98(22):224305, 2018. doi: 10.1103/PhysRevB.98.224305. URL https://link.aps.org/doi/10.1103/PhysRevB.98.224305.

Marin Bukov, Alexandre GR Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. Reinforcement learning in different phases of quantum control. *Physical Review X*, 8(3):031086, 2018. doi: 10.1103/PhysRevX.8.031086. URL https://link.aps.org/doi/10.1103/PhysRevX.8.031086.

Tommaso Caneva, Tommaso Calarco, and Simone Montangero. Chopped random-basis quantum optimization. *Phys. Rev. A*, 84:022326, 2011. doi: 10.1103/PhysRevA.84.022326. URL https://link.aps.org/doi/10.1103/PhysRevA.84.022326.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Rev. Mod. Phys.*, 91:045002, 2019. doi: 10.1103/RevModPhys.91.045002. URL https://link.aps.org/doi/10.1103/RevModPhys.91.045002.

Juan Carrasquilla. Machine learning for quantum matter. *Advances in Physics: X*, 5(1):1797528, 2020. doi: 10.1080/23746149.2020.1797528. URL https://doi.org/10.1080/23746149.2020.1797528.

Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.

Juan Carrasquilla and Giacomo Torlai. Neural networks in quantum many-body physics: a hands-on tutorial. *arXiv preprint arXiv:2101.11099*, 2021. URL https://arxiv.org/abs/2101.11099.

Chunlin Chen, Daoyi Dong, Han-Xiong Li, Jian Chu, and Tzyh-Jong Tarn. Fidelity-based probabilistic q-learning for control of quantum systems. *IEEE transactions on neural networks and learning systems*, 25(5):920–933, 2013. doi: 10.1109/TNNLS.2013.2283574.

Wei Chen, Kazuo Hida, and BC Sanctuary. Ground-state phase diagram of s= 1 xxz chains with uniaxial single-ion-type anisotropy. *Physical Review B*, 67(10):104401, 2003. doi: 10.1103/PhysRevB.67.104401.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers, 2020.

Po-Wei Chou, Daniel Maturana, and Sebastian A. Scherer. Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 834–843. PMLR, 2017. URL http://proceedings.mlr.press/v70/chou17a.html.

Simona Colabrese, Kristian Gustavsson, Antonio Celani, and Luca Biferale. Flow navigation by smart microswimmers via reinforcement learning. *Physical review letters*, 118(15):158004, 2017.

Mogens Dalgaard, Felix Motzoi, Jesper Hasseriis Mohr Jensen, and Jacob Sherson. Hessian-based optimization of constrained quantum control. *Physical Review A*, 102(4):042612, 2020a.

Mogens Dalgaard, Felix Motzoi, Jens Jakob Sorensen, and Jacob Sherson. Global optimization of quantum dynamics with alphazero deep exploration. *npj Quantum Information*, 6(1), 2020b. doi: 10.1038/s41534-019-0241-0.

Olivier Delalleau, Maxim Peter, Eloi Alonso, and Adrien Logut. Discrete and continuous action representation for practical rl in video games. *arXiv preprint arXiv:1912.11077*, 2019.

Mustafa Demirplak and Stuart A Rice. Assisted adiabatic passage revisited. *The Journal of Physical Chemistry B*, 109(14):6838–6844, 2005. URL http://pubs.acs.org/doi/abs/10.1021/jp040647w.

Joshua V Dillon, Ian Langmore, Dustin Tran, Eugene Brevdo, Srinivas Vasudevan, Dave Moore, Brian Patton, Alex Alemi, Matt Hoffman, and Rif A Saurous. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017. URL https://arxiv.org/abs/1711.10604.

Yongcheng Ding, Yue Ban, José D. Martín-Guerrero, Enrique Solano, Jorge Casanova, and Xi Chen. Breaking adiabatic quantum control with deep learning. *Physical Review A*, 103(4), apr 2021. doi: 10.1103/physreva.103.l040401. URL https://doi.org/10.1103%2Fphysreva.103.l040401.

Yulong Dong, Xiang Meng, Lin Lin, Robert Kosut, and K Birgitta Whaley. Robust control optimization for quantum approximate optimization algorithm. *arXiv preprint arXiv:1911.00789*, 2019.

Vedran Dunjko and Hans J Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018. URL https://iopscience.iop.org/article/10.1088/1361-6633/aab406.

Zhou Fan, Rui Su, Weinan Zhang, and Yong Yu. Hybrid actor-critic reinforcement learning in parameterized action space. In Sarit Kraus, editor, *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 2279–2285. ijcai.org, 2019. doi: 10.24963/ijcai.2019/316. URL https://doi.org/10.24963/ijcai.2019/316.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028v1*, Nov 2014. URL http://arxiv.org/abs/1411.4028v1.

Thomas Fösel, Petru Tighineanu, Talitha Weiss, and Florian Marquardt. Reinforcement learning with neural networks for quantum feedback. *Phys. Rev. X*, 8:031084, 2018. doi: 10.1103/PhysRevX.8.031084. URL https://link.aps.org/doi/10.1103/PhysRevX.8.031084.

Thomas Fösel, Stefan Krastanov, Florian Marquardt, and Liang Jiang. Efficient cavity control with snap gates. *arXiv preprint arXiv:2004.14256*, 2020. URL https://arxiv.org/abs/2004.14256.

Artur Garcia-Saez and Jordi Riu. Quantum observables for continuous control of the quantum approximate optimization algorithm via reinforcement learning. *arXiv preprint arXiv:1911.09682*, 2019. URL https://arxiv.org/abs/1911.09682.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: masked autoencoder for distribution estimation. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 881–889. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/germain15.html.

David Guéry-Odelin, Andreas Ruschhaupt, Anthony Kiely, Erik Torrontegui, Sofia Martínez-Garaot, and Juan Gonzalo Muga. Shortcuts to adiabaticity: Concepts, methods, and applications. *Reviews of Modern Physics*, 91(4):045001, 2019. doi: 10.1103/RevModPhys.91.045001. URL https://link.aps.org/doi/10.1103/RevModPhys.91.045001.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1352–1361. PMLR, 2017. URL http://proceedings.mlr.press/v70/haarnoja17a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018a. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b. URL https://arxiv.org/abs/1812.05905.

James Halverson, Brent Nelson, and Fabian Ruehle. Branes with brains: exploring string vacua with deep reinforcement learning. *Journal of High Energy Physics*, 2019(6):3, 2019.

Charles R. Harris, K. Jarrod Millman, St'efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern'andez del R'ıo, Mark Wiebe, Pearu Peterson, Pierre G'erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

Matthew J. Hausknecht and Peter Stone. Deep reinforcement learning in parameterized action space. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.04143.

Narendra N. Hegade, Koushik Paul, Yongcheng Ding, Mikel Sanz, F. Albarrán-Arriagada, Enrique Solano, and Xi Chen. Shortcuts to adiabaticity in digitized adiabatic quantum computing. *Physical Review Applied*, 15(2), feb 2021. doi: 10.1103/physrevapplied.15.024038. URL https://doi.org/10.1103%2Fphysrevapplied.15.024038.

Wen Wei Ho and Timothy H Hsieh. Efficient variational simulation of non-trivial quantum states. *SciPost Phys*, 6:29, 2019.

Raban Iten, Tony Metger, Henrik Wilming, Lídia Del Rio, and Renato Renner. Discovering physical concepts with neural networks. *Physical Review Letters*, 124(1):010508, 2020.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Kathryn Tunyasuvunakool, Olaf Ronneberger, Russ Bates, Augustin Žídek, Alex Bridgland, Clemens Meyer, Simon A A Kohl, Anna Potapenko, Andrew J Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Martin Steinegger, Michalina Pacholska, David Silver, Oriol Vinyals, Andrew W Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. High accuracy protein structure prediction using deep learning. *In preparation*, 2020.

Sami Khairy, Ruslan Shaydulin, Lukasz Cincio, Yuri Alexeev, and Prasanna Balaprakash. Reinforcement-learning-based variational quantum circuits optimization for combinatorial problems. *arXiv preprint arXiv:1911.04574*, 2019. URL https://arxiv.org/abs/1911.04574.

Sami Khairy, Ruslan Shaydulin, Lukasz Cincio, Yuri Alexeev, and Prasanna Balaprakash. Learning to optimize variational quantum circuits to solve combinatorial problems. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 2367–2375. AAAI Press, 2020. URL https://aaai.org/ojs/index.php/AAAI/article/view/5616.

Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrueggen, and Steffen J. Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. 172(2):296 – 305, 2005. ISSN 1090-7807. doi: http://dx.doi.org/10.1016/j.jmr.

2004.11.004. URL http://www.sciencedirect.com/science/article/pii/S1090780704003696.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Korbinian Kottmann, Patrick Huembeli, Maciej Lewenstein, and Antonio Acín. Unsupervised phase discovery with deep anomaly detection. *Physical Review Letters*, 125(17), oct 2020. doi: 10.1103/physrevlett.125.170603. URL https://doi.org/10.1103%2Fphysrevlett.125.170603.

Dieter Kraft et al. A software package for sequential quadratic programming. 1988.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1106–1114, 2012. URL https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Tejas D. Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum. Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3675–3683, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/f442d33fa06832082290ad8544a8da27-Abstract.html.

A Langari, F Pollmann, and M Siahatgar. Ground-state fidelity of the spin-1 heisenberg chain with single ion anisotropy: quantum renormalization group and exact diagonalization approaches. *Journal of Physics: Condensed Matter*, 25(40):406002, 2013. doi: 10.1088/0953-8984/25/40/406002.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.

Li Li, Minjie Fan, Marc Coram, Patrick Riley, Stefan Leichenauer, et al. Quantum optimization with a novel gibbs objective function and ansatz architecture search. *Physical Review Research*, 2(2):023074, 2020. doi: 10.1103/PhysRevResearch.2.023074.

Denghui Lu, Han Wang, Mohan Chen, Lin Lin, Roberto Car, Weinan E, Weile Jia, and Linfeng Zhang. 86 PFLOPS deep potential molecular dynamics simulation of 100 million atoms with ab initio accuracy. *Computer Physics Communications*, 259:107624, feb 2021. doi: 10.1016/j.cpc.2020.107624. URL https://doi.org/10.1016%2Fj.cpc.2020.107624.

Alicia B Magann, Matthew D Grace, Herschel A Rabitz, and Mohan Sarovar. Digital quantum simulation of molecular dynamics and control. *arXiv preprint arXiv:2002.12497*, 2020.

Alicia B. Magann, Christian Arenz, Matthew D. Grace, Tak-San Ho, Robert L. Kosut, Jarrod R. McClean, Herschel A. Rabitz, and Mohan Sarovar. From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms. *PRX Quantum*, 2 (1), jan 2021. doi: 10.1103/prxquantum.2.010101. URL https://doi.org/10.1103%2Fprxquantum.2.010101.

Warwick Masson, Pravesh Ranchod, and George Dimitri Konidaris. Reinforcement learning with parameterized actions. In Dale Schuurmans and Michael P. Wellman, editors, *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, pages 1934–1940. AAAI Press, 2016. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11981.

Shumpei Masuda and Katsuhiro Nakamura. Fast-forward of adiabatic dynamics in quantum mechanics. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20090446. The Royal Society, 2009. URL http://rspa.royalsocietypublishing.org/content/466/2116/1135.

Gabriel Matos, Sonika Johri, and Zlatko Papić. Quantifying the efficiency of state preparation via quantum variational eigensolvers. *PRX Quantum*, 2(1), 2021. doi: 10.1103/prxquantum.2.010309. URL https://doi.org/10.1103%2Fprxquantum.2.010309.

Pankaj Mehta, Marin Bukov, Ching-Hao Wang, Alexandre GR Day, Clint Richardson, Charles K Fisher, and David J Schwab. A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810:1–124, 2019. URL https://www.sciencedirect.com/science/article/pii/S0370157319300766.

Alexey A. Melnikov, Pavel Sekatski, and Nicolas Sangouard. Setting up experimental bell tests with reinforcement learning. *Physical Review Letters*, 125(16), oct 2020. doi: 10.1103/physrevlett.125.160401. URL https://doi.org/10.1103%2Fphysrevlett.125.160401.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Hendrik Poulsen Nautrup, Nicolas Delfosse, Vedran Dunjko, Hans J Briegel, and Nicolai Friis. Optimizing quantum error correction codes with reinforcement learning. *Quantum*, 3:215, 2019. doi: 10.22331/q-2019-12-16-215.

Michael Neunert, Abbas Abdolmaleki, Markus Wulfmeier, Thomas Lampe, Jost Tobias Springenberg, Roland Hafner, Francesco Romano, Jonas Buchli, Nicolas Heess, and Martin Riedmiller. Continuous-discrete reinforcement learning for hybrid control in robotics. *arXiv preprint arXiv:2001.00449*, 2020.

Murphy Yuezhen Niu, Sergio Boixo, Vadim N Smelyanskiy, and Hartmut Neven. Universal quantum control through deep reinforcement learning. *npj Quantum Information*, 5 (1):1–8, 2019. doi: 10.1038/s41534-019-0141-3. URL https://doi.org/10.1038/s41534-019-0141-3.

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. doi: 10.1038/ncomms5213. URL https://doi.org/10.1038/ncomms5213.

Frank Pollmann, Ari M. Turner, Erez Berg, and Masaki Oshikawa. Entanglement spectrum of a topological phase in one dimension. *Phys. Rev. B*, 81:064439, 2010. doi: 10.1103/PhysRevB.81.064439. URL https://link.aps.org/doi/10.1103/PhysRevB.81.064439.

Riccardo Porotti, Dario Tamascelli, Marcello Restelli, and Enrico Prati. Coherent transport of quantum states by deep reinforcement learning. *Communications Physics*, 2(1):1–9, 2019. doi: 10.1038/s42005-019-0169-x. URL https://doi.org/10.1038/s42005-019-0169-x.

Michael JD Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2):155–162, 1964.

Vinay Ramasesh, Machiel Blok, Kevin O'Brien, Dar Dahlen, John Mark Kreikebaum, and Irfan Siddiqi. Characterization and mitigation of noise and crosstalk in a five-qutrit transmon processor. In *APS March Meeting Abstracts*, volume 2019, pages A42–004, 2019.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John F. Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 9686–9698, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/37f65c068b7723cd7809ee2d31d7861c-Abstract.html.

Gautam Reddy, Antonio Celani, Terrence J Sejnowski, and Massimo Vergassola. Learning to soar in turbulent environments. *Proceedings of the National Academy of Sciences*, 113(33):E4877–E4884, 2016.

Benno S Rem, Niklas Käming, Matthias Tarnowski, Luca Asteria, Nick Fläschner, Christoph Becker, Klaus Sengstock, and Christof Weitenberg. Identifying quantum phase transitions using artificial neural networks on experimental data. *Nature Physics*, 15(9):917–920, 2019.

Ruslan Salakhutdinov. Deep learning. In Sofus A. Macskassy, Claudia Perlich, Jure Leskovec, Wei Wang, and Rayid Ghani, editors, *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, page 1973. ACM, 2014. doi: 10.1145/2623330.2630809. URL https://doi.org/10.1145/2623330.2630809.

Frédéric Sauvage and Florian Mintert. Optimal quantum control with poor statistics. *PRX Quantum*, 1(2), dec 2020. doi: 10.1103/prxquantum.1.020322. URL https://doi.org/10.1103%2Fprxquantum.1.020322.

Frank Schäfer, Michal Kloc, Christoph Bruder, and Niels Lörch. A differentiable programming method for quantum control. *Machine Learning: Science and Technology*, 1, 2020. doi: 10.1088/2632-2153/ab9802.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 1889–1897. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Dries Sels and Anatoli Polkovnikov. Minimizing irreversible losses in quantum systems by local counterdiabatic driving. *Proceedings of the National Academy of Sciences*, 114(20):E3909–E3916, 2017.

Or Sharir, Yoav Levine, Noam Wies, Giuseppe Carleo, and Amnon Shashua. Deep autoregressive models for the efficient variational simulation of many-body quantum systems. *Phys. Rev. Lett.*, 124:020503, 2020. doi: 10.1103/PhysRevLett.124.020503. URL https://link.aps.org/doi/10.1103/PhysRevLett.124.020503.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Sukin Sim, Jonathan Romero, Jérôme F Gonthier, and Alexander A Kunitsa. Adaptive pruning-based optimization of parameterized quantum circuits. *Quantum Science and Technology*, 6(2):025019, mar 2021. doi: 10.1088/2058-9565/abe107. URL https://doi.org/10.1088%2F2058-9565%2Fabe107.

Vegard B. Sørdal and Joakim Bergli. Deep reinforcement learning for quantum szilard engine optimization. *Physical Review A*, 100(4), oct 2019. doi: 10.1103/physreva.100.042314. URL https://doi.org/10.1103%2Fphysreva.100.042314.

Kevin Sung. *Towards the First Practical Applications of Quantum Computers*. PhD thesis, 2020.

Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. Using models to improve optimizers for variational quantum algorithms. *Quantum Science and Technology*, 5(4):044008. doi: 10.1088/2058-9565/abb6d9. URL https://doi.org/10.1088%2F2058-9565%2Fabb6d9.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Zsolt Tabi, Kareem H El-Safty, Zsófia Kallus, Péter Hága, Tamás Kozsik, Adam Glos, and Zoltán Zimborás. Quantum optimization for the graph coloring problem with space-efficient embedding. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 56–62. IEEE, 2020.

Ho Lun Tang, V.O. Shkolnikov, George S. Barron, Harper R. Grimsley, Nicholas J. Mayhall, Edwin Barnes, and Sophia E. Economou. Qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor. *PRX Quantum*, 2(2), apr 2021. doi: 10.1103/prxquantum.2.020310. URL https://doi.org/10.1103%2Fprxquantum.2.020310.

Yunhao Tang and Shipra Agrawal. Boosting trust region policy optimization by normalizing flows policy. *arXiv preprint arXiv:1809.10326v3*, Sep 2018. URL http://arxiv.org/abs/1809.10326v3.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv preprint arXiv:2009.06732v2*, Sep 2020. URL http://arxiv.org/abs/2009.06732v2.

Hugo Théveniaut and Evert van Nieuwenburg. A neat quantum error decoder. *arXiv preprint arXiv:2101.08093*, 2021. URL https://arxiv.org/abs/2101.08093.

Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018.

Agnes Valenti, Evert van Nieuwenburg, Sebastian Huber, and Eliska Greplova. Hamiltonian learning for quantum error correction. *Physical Review Research*, 1(3):033092, 2019.

Evert PL Van Nieuwenburg, Ye-Hua Liu, and Sebastian D Huber. Learning phase transitions by confusion. *Nature Physics*, 13(5):435–439, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake Vand erPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1. 0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: https://doi.org/10.1038/s41592-019-0686-2.

Guoming Wang, Dax Enshan Koh, Peter D. Johnson, and Yudong Cao. Minimizing estimation runtime on noisy quantum computers. *PRX Quantum*, 2(1), mar 2021. doi: 10.1103/prxquantum.2.010346. URL https://doi.org/10.1103%2Fprxquantum.2.010346.

Lei Wang. Discovering phase transitions with unsupervised learning. *Phys. Rev. B*, 94:195105, 2016. doi: 10.1103/PhysRevB.94.195105. URL https://link.aps.org/doi/10.1103/PhysRevB.94.195105.

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity, Jun 2020. URL http://arxiv.org/abs/2006.04768v3.

Matteo M Wauters, Emanuele Panizon, Glen B Mbeng, and Giuseppe E Santoro. Reinforcement learning assisted quantum optimization. *Phys. Rev. Research*, 2:033446, 2020. doi: 10.1103/PhysRevResearch.2.033446. URL https://link.aps.org/doi/10.1103/PhysRevResearch.2.033446.

Ermo Wei, Drew Wicke, and Sean Luke. Hierarchical approaches for reinforcement learning in parameterized action space. *arXiv preprint arXiv:1810.09656v1*, Oct 2018. URL http://arxiv.org/abs/1810.09656v1.

Phillip Weinberg and Marin Bukov. Quspin: a python package for dynamics and exact diagonalisation of quantum many body systems part i: spin chains. *SciPost Phys*, 2(1), 2017.

Phillip Weinberg and Marin Bukov. Quspin: a python package for dynamics and exact diagonalisation of quantum many body systems. part ii: bosons, fermions and higher spins. *SciPost Phys.*, 7 (arXiv: 1804.06782):020, 2019.

Dian Wu, Lei Wang, and Pan Zhang. Solving statistical mechanics using variational autoregressive networks. *Physical Review Letters*, 122(8), 2019. doi: 10.1103/physrevlett.122.080602. URL https://doi.org/10.1103%2Fphysrevlett.122.080602.

Re-Bing Wu, Xi Cao, Pinchen Xie, and Yu-xi Liu. End-to-end quantum machine learning implemented with controlled quantum dynamics. *Physical Review Applied*, 14(6), dec 2020a. doi: 10.1103/physrevapplied.14.064020. URL https://doi.org/10.1103%2Fphysrevapplied.14.064020.

Yadong Wu, Zengming Meng, Kai Wen, Chengdong Mi, Jing Zhang, and Hui Zhai. Active learning approach to optimization of experimental control. *Chinese Physics Letters*, 37(10):103201, oct 2020b. doi: 10.1088/0256-307x/37/10/103201. URL https://doi.org/10.1088%2F0256-307x%2F37%2F10%2F103201.

Jiechao Xiong, Qing Wang, Zhuoran Yang, Peng Sun, Lei Han, Yang Zheng, Haobo Fu, Tong Zhang, Ji Liu, and Han Liu. Parametrized deep q-networks learning: Reinforcement learning with discrete-continuous hybrid action space. *arXiv preprint arXiv:1810.06394*, 2018.

Jiahao Yao, Marin Bukov, and Lin Lin. Policy gradient based quantum approximate optimization algorithm. pages 605–634, 2020a. URL http://proceedings.mlr.press/v107/yao20a.html.

Jiahao Yao, Lin Lin, and Marin Bukov. Reinforcement learning for many-body ground state preparation based on counter-diabatic driving. *arXiv preprint arXiv:2010.03655*, 2020b. URL https://arxiv.org/abs/2010.03655.

Appendix A. Discrete Action Space inspired by Counter-Diabatic (CD) Driving

In Sec. 2.4, we discussed a natural way to suitably generalize the QAOA ansatz, by introducing a second discrete variational problem: to construct the unitary

$$U(\{\alpha_j\}_{j=1}^{q}, \tau^{\mathrm{d}}) = \prod_{j=1}^{q} \exp\Big(-i\alpha_j H_{\tau_j^{\mathrm{d}}}\Big),$$

cf. Eq. (6), our RL agent has to choose the sequence τ^{d} of gates $\exp\Big(-i\alpha_j H_{\tau_j^{\mathrm{d}}}\Big)$. Hence, at every step j during the RL episode, the agent's action consists of a choice of (i) a Hermitian operator $H_{\tau_j^{\mathrm{d}}} \in \mathcal{A}^{\mathrm{d}}$, in addition to (ii) a continuous duration $\alpha_j \in \mathcal{A}^{c}$ to apply H_j for.

Naturally, the set of discrete actions $\mathcal{A}^{\mathrm{d}}$ consists of the available possible controls in an experiment. In Refs. (Yao et al., 2020b; Hegade et al., 2021; Ding et al., 2021), it was shown that a particularly suitable set of actions for ground state preparation in quantum many-body systems, is given by terms appearing in the series of the variational adiabatic gauge potential, designed for many-body counter-diabatic driving (Sels and Polkovnikov, 2017). These terms provide shortcuts in the Hilbert space that may significantly decrease the time required to prepare the ground state.

For brevity, here we just list the generator set $\mathcal{A}^{\mathrm{d}}$ for the spin$-1/2$ Ising model, cf. Eq. (3), which the RL agent has access to, and refer the interested readers to Ref. (Yao et al., 2020b) for more details:

$$\mathcal{A}^{\mathrm{d}}_{\text{Ising}} = \{H_1, H_2, Y, X|Y, Y|Z\}, \tag{15}$$

with $Y = \sum_i S_i^y$, $X|Y = \sum_i S_i^x S_{i+1}^y + S_i^y S_{i+1}^x$, and $Y|Z = \sum_i S_i^y S_{i+1}^z + S_i^z S_{i+1}^y$; H_1, H_2 are defined in Eq. (3). A reader with an experienced eye will notice that, besides the conventional QAOA generators H_1 and H_2, the extra terms we consider consist of the leading order local imaginary-valued terms one can write down for the spin chain. This general rule of thumb is valued for any real-valid Hamiltonian H.

For the spin-1 Heisenberg model, the set of discrete actions reads

$$\mathcal{A}^{\mathrm{d}}_{\text{Heisenberg}} = \{H_1, H_2, Z, X|X; Y, XY, YZ, X|Y - XY, Y|Z - YZ\}, \tag{16}$$

with H_1, H_2 are defined in Eq. (14); $Z = \sum_i S_i^z$, $X|X = \sum_i S_i^x S_{i+1}^x$, and the imaginary-valued gauge potential constituent terms: $Y = \sum_i S_i^y$, $XY = \sum_i S_i^x S_i^y + S_i^y S_i^x$, $YZ = \sum_i S_i^y S_i^z + S_i^z S_i^y$ and $X|Y = \sum_i S_i^x[S_{i+1}^y - aS_i^y] + S_i^y[S_{i+1}^x - aS_i^x]$, and $Y|Z = \sum_i S_i^y[S_{i+1}^z - bS_i^z] + S_i^z[S_{i+1}^y - bS_i^y]$, where the constants a and b are introduced to orthogonalize the last five terms w.r.t. the Hilbert-Schmidt norm. Here, the S_i^α are the spin-1 operators. More details can be found in Appendix D of Ref. (Yao et al., 2020b).

We emphasize that this is just one particular choice of $\mathcal{A}^{\mathrm{d}}$ for each model. In practice, the algorithm is agnostic to the discrete action space which is determined by the available controls for the system of interest: e.g., on a quantum computer, these can be a universal set of gates, etc., or, one can also consider the minimal complete pools from which recent studies construct hardware-efficient ansätze (Tang et al., 2021) for Variational Quantum Eigensolver (VQE) on NISQ devices.

Appendix B. Pseudocode and Algorithm Hyperparameters

The pseudocode for RL-QAOA is outlined in Algorithm (1). The agent samples a batch of actions from the autoregressive network. Then, the corresponding expected energy density is computed using a classical simulator for the quantum dynamics. Below, we focus on the noise-free case.

Algorithm 1 Autoregressive network based reinforcement learning: RL-QAOA

```
Input: batch size $M$, learning rate $\eta_t$, total number of iterations $T_{\text{iter}}$, exponential moving average
    coefficient $m$, entropy coefficient $\beta_S^{-1}$, PPO gradient steps $K$.
Initialize the autoregressive network and initialize the moving average $\hat{R}=0$.
for $t = 1, .., T_{\text{iter}}$ do
    Autoregrssively sample a batch of hybrid actions of size $M$, denoted by $B$:
        $\tau^{\{k\}} = (a_1^{\{k\}}, a_2^{\{k\}}, \cdots, a_q^{\{k\}}) \sim \pi_{\boldsymbol{\theta}}(a_1, a_2, \cdots, a_q),\ k = 1, 2, \cdots, M.$
    Measure the observables and use the negative energy density as the return and compute the
    moving average of the return
        $R_k = -\mathcal{E}_k = -\frac{1}{N}\langle\psi_i|U^\dagger(\{a_j^{\{k\}}\}_{j=1}^q) H U(\{a_j^{\{k\}}\}_{j=1}^q)|\psi_i\rangle, \quad \hat{R} = m\cdot\hat{R} + (1-m)\cdot\frac{1}{M}\sum_{k=1}^{M} R_k.$
    Compute the advantage estimates $A_k = R_k - \hat{R}$
    Initialize the parameter $\boldsymbol{\theta}_{t+1}^{[1]} = \boldsymbol{\theta}_t$.
    for $\kappa = 1, .., K$ do
        Evaluate the samples' likelihood using the parameter from the last iterations and current
        iterations, i.e. $\pi_{\boldsymbol{\theta}_{t+1}^{[\kappa]}}(\tau^{\nu,\{k\}})$, $\pi_{\boldsymbol{\theta}_t}(\tau^{\nu,\{k\}})$ and compute the importance weight
            $\rho_k^{[\kappa],\nu} = \pi_{\boldsymbol{\theta}_{t+1}^{[\kappa]}}(\tau^{\nu,\{k\}}) / \pi_{\boldsymbol{\theta}_t}(\tau^{\nu,\{k\}}).$
        Using the advantage estimation and importance weight to compute
        $\mathcal{G}_k^{[\kappa],\text{d}}, \mathcal{G}_k^{[\kappa],\text{c}}, \mathcal{S}_k^{[\kappa],\text{d}}, \mathcal{S}_k^{[\kappa],\text{c}}$.
        Compute the RL-QAOA objective Eq (12) and backpropagate to get the gradients.
            $\nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}_{t+1}^{[\kappa]}) = \frac{1}{M} \sum_{\{a_j^{\{k\}}\}_{j=1}^q \in B} \nabla_{\boldsymbol{\theta}} \left[ \mathcal{G}_k^{[\kappa],\text{d}} + \mathcal{G}_k^{[\kappa],\text{c}} + \beta_S^{-1} (\mathcal{S}_k^{[\kappa],\text{d}} + \mathcal{S}_k^{[\kappa],\text{c}}) \right].$
        Update weights $\boldsymbol{\theta}_{t+1}^{[\kappa+1]} \leftarrow \boldsymbol{\theta}_{t+1}^{[\kappa]} + \eta_t \nabla_{\boldsymbol{\theta}} \mathcal{J}(\boldsymbol{\theta}_{t+1}^{[\kappa]})$.
    end for
    Update the parameter $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_{t+1}^{[K+1]}$
end for
```

Dealing with noise requires a trivial modification of the reward signal following Sec. 4.3. The baseline for the reward is estimated through an exponential moving average. Finally, proximal policy optimization is applied to update the agent's policy.

We also conducted coarse hyperparameter sweeps to find the optimal values for the hyperparameters of RL-QAOA, cf. Table 2. We use a batch size of 128 to train the policy. The policy network is optimized using Adam. The initial learning rate is set to 5×10^{-4}, which is typical when training autoregressive networks; we employ a learning rate decay schedule which decreases by a factor of 0.98 every 50 iterations. The Autoregressive network is implemented using uniform masks and dense layers (Germain et al., 2015). The base layer (see Fig. 2) consists of two hidden layers with 100 neurons each and the heads contain $3|\mathcal{A}^{\text{d}}|$ neurons in total.

The agent is trained via proximal policy optimization (PPO). We use four PPO updates to the policy network parameters per iteration. The clipping parameters are set as $\epsilon^{\text{c}} = 0.1$ for the continuous policy, and $\epsilon^{\mathrm{d}} = 10^{-3}$ for the discrete policy. We include entropy bonus to increase exploration; the corresponding temperature schedule $\beta_{\mathcal{S}}^{-1}$ starts at 1×10^{-1}, and drops by a factor of 0.99 every 50 iterations.

Table 2: RL-QAOA Hyperparameters.

HYPERPARAMETER	VALUE
OPTIMIZER	ADAM (KINGMA AND BA, 2015)
LEARNING RATE ($\eta_{\{0\}}$)	5×10^{-4}
LIKELIHOOD RATIO CLIP (ϵ^{ν})	0.1 (ϵ^{c})
	0.001 (ϵ^{d})
PPO EPOCHS (K)	4
HIDDEN UNITS (MASKED DENSE LAYER)	$[100, 100]$
ACTIVATION FUNCTION	RELU
BASELINE EXPONENTIAL MOVING AVERAGE (m)	0.95
LEARNING RATE ANNEALING STEPS	50
LEARNING RATE ANNEALING FACTOR	0.98
LEARNING RATE ANNEALING STYLE	STAIRCASE
ENTROPY BONUS TEMPERATURE ($\beta_{S,\{0\}}^{-1}$)	1×10^{-1}
ENTROPY BONUS TEMPERATURE DECAY STEPS	50
ENTROPY BONUS TEMPERATURE DECAY FACTOR	0.99
ENTROPY BONUS TEMPERATURE DECAY STYLE	SMOOTH
MINIBATCH SIZE (M)	128

A typical learning curve in the noisy setting is shown in Fig. 5. Three quantities are recorded to measure the performance of the agent. In the noise setting, these quantities correspond to the ideal noise-free case. We use them only for the purpose of evaluation; during the training, the RL agent only has access to the noisy rewards. These quantities are shown in terms of the energy ratio with respect to the target ground state, so the possible maximum is upper bounded by one; since the energy of a state can be either positive or negative, while the GS has a negative value, negative ratios are possible. Figure 5 shows that the agent starts to pick up the learning signal around two thousand iterations. After that, it slightly modifies the policy in order to achieve a higher reward. Here, the mean reward stands for the sample mean of energy density at every iteration; the max reward is the maximum over the sample; the history best is the best-encountered reward during the entire training process.

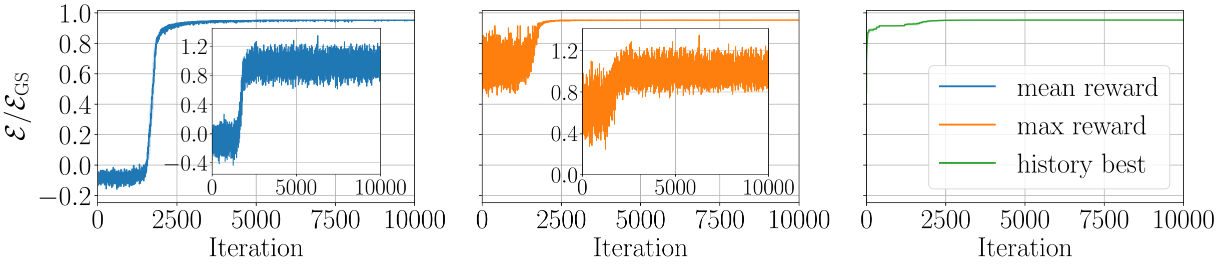


Figure 5: Spin-1/2 Ising model: training curves for RL-QAOA with energy minimization as a cost function. The quantities in the main figure are noiseless evaluation, while in the inset are noisy measurement. The noiseless quantities are only for the evaluation's purpose, and the agent can only access the noisy quantities (in the inset). The mean reward (blue curve) is the average energy ratio across the minibatch sampled from the autoregressive policy; the max reward (orange curve) is taking the maximum across the minibatch; the history best bookkeeps the best ever max reward during the training. The total duration is $T = 10$ and the number of spin-1/2 particles is $N = 8$. The discrete RL-QAOA action space is $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2; Y, X|Y, Y|Z\}$, and we use $q = 8$. Here, the noise is classic gaussian noise, with the noise level $\gamma = 0.1$.

Appendix C. A Comparison of Compactly Supported Distributions defining Continuous Actions

C.1. Sigmoid Gaussian Distribution

In order to enforce the durations sampled from the continuous Gaussian policy to be bounded, we apply the sigmoid function. Bounded actions are particularly useful in practice in order to be able to normalize the durations to match the total protocol duration T; otherwise, one would observe a large variance (see main text).

To achieve this, we can apply the sigmoid function to the Gaussian distribution. In the following formula, we have $x = f(y)$, where $f(y) = \frac{1}{1+e^{-y}}$ is the sigmoid. We denote the original distribution as $\pi_0(y; \kappa, \xi)$ and the distribution after the transformation, as $\pi(x; \kappa, \xi)$:

$$\pi(x; \kappa, \xi) = \pi_0(y; \kappa, \xi) \left| \det \left(\frac{\mathrm{d}x}{\mathrm{d}y} \right) \right|^{-1}$$

For example, if we choose π_0 to be Gaussian distribution according to $\mathcal{N}(\kappa, \xi^2)$, then

$$\log \pi(x; \kappa, \xi) = -\log \xi - \frac{1}{2}\log(2\pi) - \frac{1}{2}\left(\frac{\mathrm{logit}(x) - \kappa}{\xi}\right)^2 - \log(x\,(1-x)). \tag{17}$$

Here, the logit function, $\mathrm{logit}(x) = \log x - \log(1-x)$, is the inverse of the sigmoid function $f(x) = 1/(1 + \exp(-x))$.

Thus, the derivative with respect to the parameters (i.e., κ and ξ) can be computed analytically, and reads

$$\frac{\partial \log \pi(x;\kappa,\xi)}{\partial \kappa} = \frac{\operatorname{logit}(x)-\kappa}{\xi^2}, \tag{18}$$

$$\frac{\partial \log \pi(x;\kappa,\xi)}{\partial \xi} = -\frac{1}{\xi} + \frac{1}{\xi}\left(\frac{\operatorname{logit}(x)-\kappa}{\xi}\right)^2. \tag{19}$$

We use this log probability in the policy gradient formula to speed up training.

C.2. Beta Distribution

The probability density function of the beta distribution is defined as:

$$\pi(x;\kappa,\xi) = \frac{\Gamma(\kappa+\xi)}{\Gamma(\kappa)\Gamma(\xi)} x^{\kappa-1}(1-x)^{\xi-1},$$

where the Gamma function is $\Gamma(z) = \int_0^\infty x^{z-1}e^{-t}\mathrm{d}t$. Here, the κ and ξ are the parameters of the beta distribution, which can be learned by the autoregressive policy network. The corresponding log-probability takes the form

$$\log \pi(x;\kappa,\xi) = \log\Gamma(\kappa+\xi) - \log\Gamma(\kappa) - \log\Gamma(\xi) + (\kappa-1)\log(x) + (\xi-1)\log(1-x). \tag{20}$$

Thus, the derivative with respect to the parameters (i.e., κ and ξ) reads

$$\frac{\partial \log \pi(x;\kappa,\xi)}{\partial \kappa} = \psi(\kappa+\xi) - \psi(\kappa) + \log(x), \tag{21}$$

$$\frac{\partial \log \pi(x;\kappa,\xi)}{\partial \xi} = \psi(\kappa+\xi) - \psi(\xi) + \log(1-x), \tag{22}$$

where the digamma function is defined as the logarithmic derivative of the Gamma function:

$$\psi(x) = \frac{d}{dx}\ln\big(\Gamma(x)\big) = \frac{\Gamma'(x)}{\Gamma(x)}. \tag{23}$$

This expression for the gradient can be used to compute the policy gradient for faster training.

In the Sec. 3.3, we mainly talk about the sigmoid-Gaussian distribution. Nevertheless, switching to the beta distribution only requires several simple modifications. As to Eqn. 10, we instead parametrize $z_j^\kappa = \exp\Big(V_j^\kappa y_{\le j} + c_j^\kappa\Big) + 1$, $z_j^\xi = \exp\Big(V_j^\xi y_{\le j} + c_j^\xi\Big) + 1$, where Beta distribution has parameters bigger than one, and thus is concave and unimodal (Chou et al., 2017). Then, for example, we can sample the first duration $a_1^{\mathrm{c}} \sim \pi(a_1^{\mathrm{c}}|a_1^{\mathrm{d}}) = \mathrm{B}\left(z_1^\kappa[a_1^{\mathrm{d}}], z_1^\xi[a_1^{\mathrm{d}}]\right)$.

Another good feature of the beta distribution is an analytic formula for entropy $\mathcal{S}^{\mathrm{c}}(\mathrm{B}(\alpha,\beta)) = \ln\mathrm{B}(\alpha,\beta) - (\alpha-1)\psi(\alpha) - (\beta-1)\psi(\beta) + (\alpha+\beta-2)\psi(\alpha+\beta)$. Thus, we adopt this formula as an entropy regularization in the optimization.

Appendix D. Choosing the protocol duration T

In this appendix, we explain the choice of protocol duration $JT = 10$ used in our study. Figure 6 shows a scan of the best energy over the protocol duration T in the noise-free case for $N = 4$ qubits using the three methods: QAOA, CD-QAOA and adiabatic driving. For the adiabatic driving, we consider the driven spin-1/2 Ising model:

$$H(\lambda) = \lambda(t)H + (1 - \lambda(t))\tilde{H}, \tag{24}$$

where $\lambda(t) = \sin^2\left(\frac{\pi t}{2T}\right)$, $t \in [0, T]$, is a smooth protocol satisfying the boundary conditions: $\lambda(0) = 0$, $\lambda(T) = 1$, $\dot{\lambda}(0) = 0 = \dot{\lambda}(T)$. The initial state is the ground state at $t = 0$, i.e. $|\psi_i\rangle = |\uparrow \cdots \uparrow\rangle$, while the target state is the ground state of the Ising model at $t = T$ for $h_z/J = 0.4523$ and $h_x/J = 0.4045$. Here, H is the target Hamiltonian defined in Eq. (3), and $\tilde{H} = -\sum_{i=1}^{N} S_i^z$.

The value $JT = 10$ is selected to achieve a compromise: on the one hand, it is large enough for CD-QAOA to reach close enough to the ground state; on the other hand, it is small enough for a discrepancy between the performance of CD-QAOA and QAOA to become clearly visible. Hence, $JT = 10$ exemplifies nicely the benefits of using the generalized QAOA ansatz, compared to QAOA. Last, we emphasize that $JT = 10$ is far away from the adiabatic regime, as shown by the adiabatic curve.

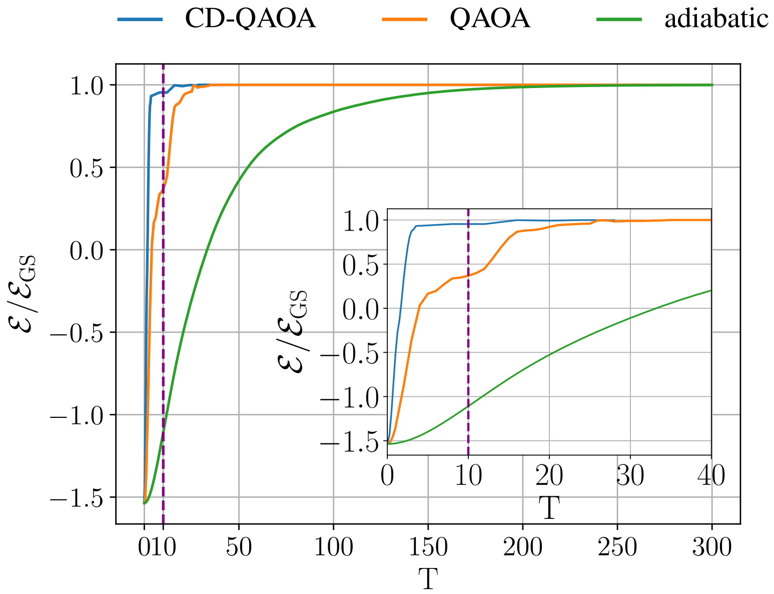
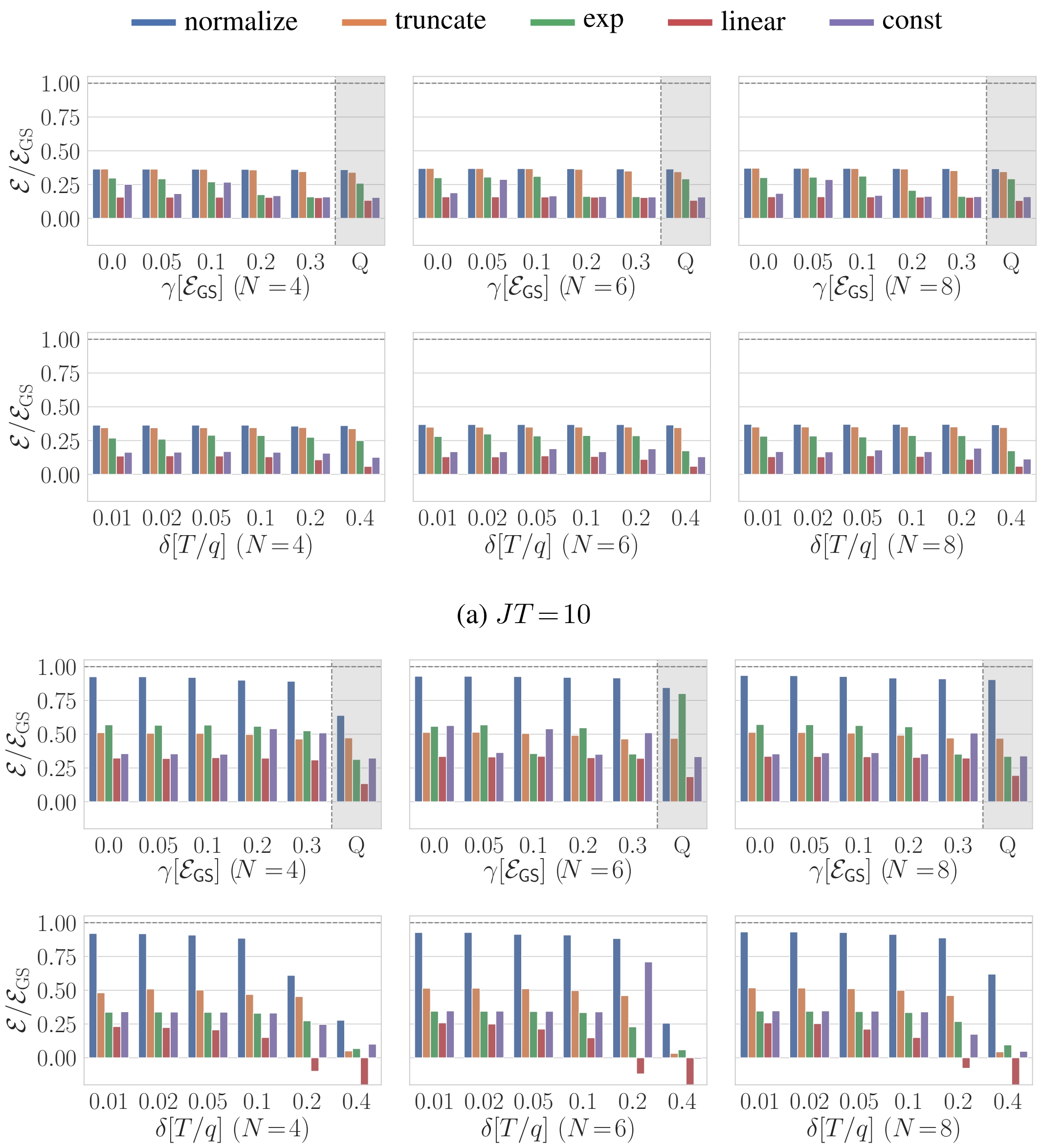


Figure 6: Spin-1/2 Ising model: energy minimization at different protocol duration T for three different methods in the nose-free setup: CD-QAOA (blue line), QAOA (red line), adiabatic evolution (green line). The physics model and the setting are the same as in Sec. 4.1. For the adiabatic driving simulation, we used the protocol function $\lambda(t) = \sin^2\left(\frac{\pi t}{2T}\right)$, $t \in [0, T]$. The quantum dynamics was solved for numerically, using a step size of $\Delta t = 1 \times 10^{-3}$. The system size is $N = 4$. The vertical purple dashed line corresponds to $JT = 10$.



(a) $JT=10$

(b) $JT=28$

Figure 7: Spin-1/2 Ising chain: Comparison among different normalizing methods for the protocol durations in the PG-QAOA algorithm. The blue bar is the normalization mentioned in Sec. 2.3; the orange one is truncating the duration up to T by modifying the last protocol duration or trimming the protocol to match the total duration; the last three correspond to three different penalty functions in Eqn. 25. The experiment setups are energy minimization in the Ising model against different noise levels (odd rows display measurement noise (including quantum), and even rows show gate noise) with circuit depths $p=q/2=4$, different system sizes $N=4,6,8$ and protocol duration $JT=10$ (first two rows) and $JT=28$ (last two rows). The initial and target states are $|\psi_i\rangle=|\uparrow\cdots\uparrow\rangle$ and $|\psi_*\rangle=|\psi_{\mathrm{GS}}(H)\rangle$ for $h_z/J=0.4523$ and $h_x/J=0.4045$. The alternating unitaries for PG-QAOA are generated by $\mathcal{A}^{\mathrm{d}}=\{H_1,H_2\}$.

Appendix E. Implementing Constrained Protocol Durations

As mentioned in Section 2.3, there exist different methods to constrain the total protocol duration of the PG-QAOA algorithm, and fix it to the value T. We implemented and compared three different methods in the investigation:

1. The first is the normalization method described in Section 2.3 that normalizes the sum of the durations;

2. The second method is the truncation method by adjusting the tail values of the protocol to ensure that the total duration matches the protocol duration constraint T, i.e. appending the last protocol duration with the difference when below the constraint, or trimming the protocol duration sharply before it exceeds the constraint;

3. The last method is a penalty method by adding extra negative penalty to the reward when the total protocol duration tops the constraint T.

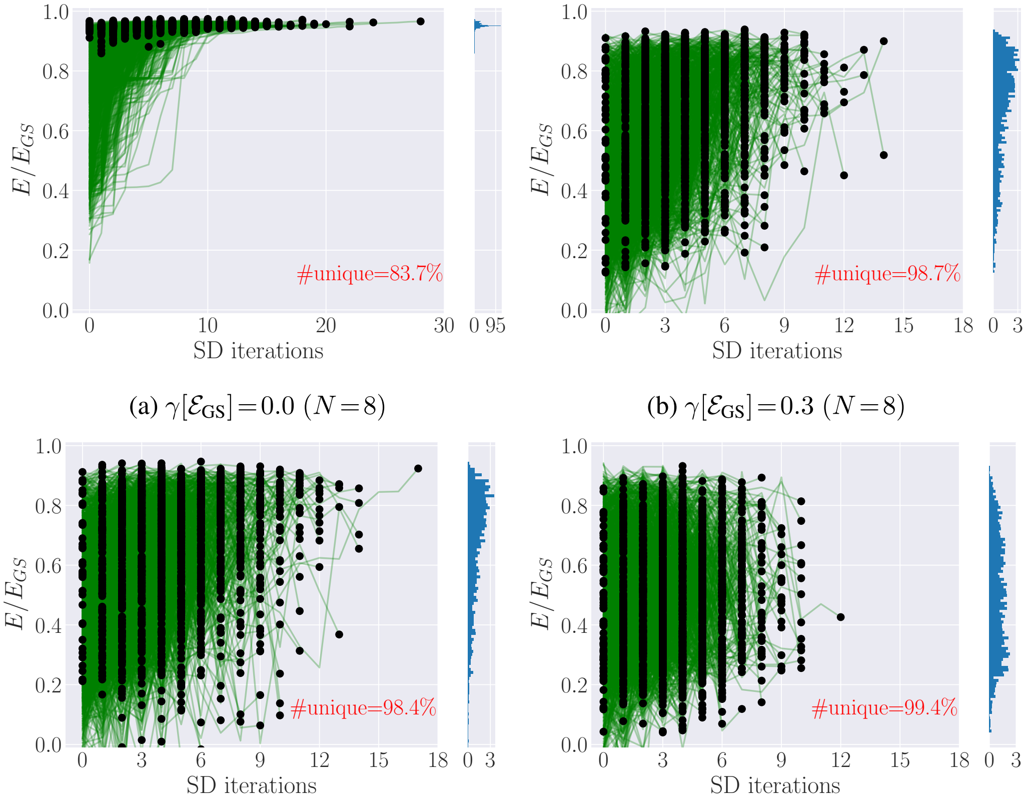
Method 3 incentivizes the RL agent to stay inside the protocol duration constraint by penalizing any violation. If the protocol duration exceeds the constraint, the reward drops, thereby encouraging the agent to stay away from this region. Nevertheless, there is no guarantee that the protocol duration exactly matches the total duration.

Method 3 also involves the design of the negative penalty function, and so we compared three different shapes [Eq. (25)]: exponential, linear, and constant. The extra formulas for the different penalty reward functions which we used in the investigations are shown below.

$$\begin{cases} q_{\text{exp}}(a^{\text{c}}, T) = \exp\left(\sum_i a_i^{\text{c}} - T\right) - 1.1 \\ q_{\text{linear}}(a^{\text{c}}, T) = -3\left(\sum_i a_i^{\text{c}} - T\right) - 0.1 \\ q_{\text{const}}(a^{\text{c}}, T) = -1 \end{cases} \tag{25}$$

Figure 7 details the comparison of the three different methods (five different values are shown since there are three separate implementations for the last method). We can see that while the truncation method is comparable to the normalization method in the $JT = 10$ case, the normalization method significantly outperforms all other methods in the $JT = 28$ case across all noise values and sources. For this reason, we chose to proceed with the normalization method in benchmark of different algorithms. That said, we provide no formal proof that the normalization method is the optimal way to implement the total duration constraint, and it is conceivable that yet more efficient methods exists.

Appendix F. Local Minima in the Discrete Optimization Landscape



(a) $\gamma[\mathcal{E}_{\mathrm{GS}}]=0.0$ $(N=8)$

(b) $\gamma[\mathcal{E}_{\mathrm{GS}}]=0.3$ $(N=8)$

(c) $\delta[\mathcal{E}_{\mathrm{GS}}]=0.4$ $(N=8)$

(d) Q $(N=8)$

Figure 8: Spin-$1/2$ Ising chain: The discrete optimization landscape visualization for the discrete protocol sequences using the stochastic descent for energy minimization against different noise levels: a) noise free setting, b) classical measurement noise, c) gate noise, d) Quantum measurement noise. The green trajectories corresponds to the path for each realization; the black dot is each ending point. The ratio of the different protocol is shown in the red in the lower right corner. The right panel shows the density histogram of the final protocol's energy ratio for the 2000 random realizations. The Ising model's system size is $N = 8$ with circuit depths $p = q/2 = 4$ and protocol duration $JT = 10$. The initial and target states are $|\psi_i\rangle = |\uparrow \cdots \uparrow\rangle$ and $|\psi_*\rangle = |\psi_{\mathrm{GS}}(H)\rangle$ for $h_z/J = 0.4523$ and $h_x/J = 0.4045$. The discrete protocol pool is of size 5: $\mathcal{A}^{\mathrm{d}} = \{H_1, H_2; Y, X|Y, Y|Z\}$.

Our aim in this section is to obtain some preliminary understanding of the control landscape, and in particular to investigate how smooth/rugged it is. For simplicity, instead of the joint continuous-discrete optimization problem, we consider two independent optimization landscapes, corresponding to the continuous and discrete problems independently. This is sufficient, since we provide numerical evidence that the discrete optimization landscape already contains a large number of unique local minima, which makes searching for the optimal protocol sequence a formidable challenge. The continuous landscape was investigated in Ref. (Yao et al., 2020b), so we focus on the discrete case here.

In order to investigate the topography of the discrete optimization landscape, we use the Stochastic Descent (SD) method, cf. Ref. (Bukov, 2018). SD starts with a random initialization of the discrete protocol sequence. At each SD step, we sample a random position in the protocol (i.e., time step) and randomly perturb the discrete action (i.e. Hamiltonian gate) at that position. Then we use an optimization solver to find the durations of the continuous actions, which ultimately allows us to compute the reward associated with a given protocol sequence. If the resulting protocol sequence can achieve a lower energy, we accept the new protocol; otherwise, we keep the previous protocol and repeat the same process to generate another protocol. We stop the procedure once it is no longer possible to find a change in the discrete actions that yields a lower-energy protocol; the resulting protocol sequence then corresponds to a local (one-flip) landscape minimum. Repeating the SD procedure a number of times starting from a different random initial protocol sequence provides us with an empirical sample of the topography of local minima in the discrete optimization landscape.

In Fig. 8, we start with 2000 randomly initialized protocol sequences of length $JT = 10$, and visualize the local minima that SD gets stuck into. In the absence of noise, we find a total of 80% unique discrete protocol sequences that coincide with local minima in the landscape. On the other hand, the landscape becomes visibly more rugged in the presence of noise, which is the main focus of this paper. The percentage of unique local minima protocols in the noisy settings is at least 98%. The sets of local minima in the three different noise types also have interesting characteristics: compared to those with the classical measurement noise and the gate noise, the energy distribution of the quantum measurement noise spreads out further in the higher energy regime. Hence, the quantum measurement noise control problem appears to possess the most difficult landscape among the three kinds of noises from the perspective of SD.

Notice that the RL agent considered in the main text consistently finds low-energy protocol sequences in the presence of noise [Fig. 3]. Hence, we conclude that, during training, the agent learns to escape the many lower-energy local minima in the landscape. However, there is no guarantee that the agent finds the global minimum (given the learning rate attenuation schedule most likely it does not).