Model-Free Quantum Control with Reinforcement Learning

V. V. Sivak,[1,*] A. Eickbusch,[1] H. Liu,[1] B. Royer,[2] I. Tsioutsios,[1] and M. H. Devoret[1,†]
[1]Department of Applied Physics, Yale University, New Haven, Connecticut 06520, USA
[2]Department of Physics, Yale University, New Haven, Connecticut 06520, USA

(Received 3 May 2021; revised 3 December 2021; accepted 28 January 2022; published 28 March 2022)

Model bias is an inherent limitation of the current dominant approach to optimal quantum control, which relies on a system simulation for optimization of control policies. To overcome this limitation, we propose a circuit-based approach for training a reinforcement learning agent on quantum control tasks in a model-free way. Given a continuously parametrized control circuit, the agent learns its parameters through trial-and-error interaction with the quantum system, using measurement outcomes as the only source of information about the quantum state. Focusing on control of a harmonic oscillator coupled to an ancilla qubit, we show how to reward the learning agent with measurements of experimentally available observables. We train the agent to prepare various nonclassical states via both unitary control and control with adaptive measurement-based quantum feedback, and to execute logical gates on encoded qubits. The agent does not rely on averaging for state tomography or fidelity estimation, and significantly outperforms widely used model-free methods in terms of sample efficiency. Our numerical work is of immediate relevance to superconducting circuits and trapped ions platforms where such training can be implemented in experiment, allowing complete elimination of model bias and the adaptation of quantum control policies to the specific system in which they are deployed.

DOI: 10.1103/PhysRevX.12.011059 Subject Areas: Quantum Information

I. INTRODUCTION

Quantum control theory addresses a problem of optimally implementing a desired quantum operation using external controls. The design of experimental control policies is currently dominated by simulation-based optimal control theory methods, with favorable convergence properties thanks to the availability of analytic gradients [1–3] or automatic differentiation [4,5]. However, it is important to acknowledge that simulation-based methods can only be as good as the underlying models used in the simulation. Empirically, model bias leads to a significant degradation of performance of the quantum control policies, when optimized in simulation and then tested in experiment [6–9]. A practical model-free alternative to simulation-based methods in quantum control is thus desirable.

The idea of using model-free optimization in quantum control can be traced back to the pioneering proposal in 1992 of laser pulse shaping for molecular control with a genetic algorithm [10]. Only in recent years has the controllability of quantum systems and the duty cycle of optimization feedback loops reached sufficient levels to allow for the experimental implementation of such ideas. The few existing demonstrations are based on model-free optimization algorithms such as the Nelder-Mead (NM) simplex search [6–8], evolutionary strategies [9], and particle swarm optimization [11].

At the same time, deep reinforcement learning (RL) [12,13] emerged as not only a powerful optimization technique but also a tool for discovering adaptive decision-making policies. In this framework, learning proceeds by trial and error, without access to the model generating the dynamics and its gradients. Being intrinsically free of model bias, it is an attractive alternative to traditional simulation-based approaches in quantum control. In a variety of domains, deep reinforcement learning has recently produced spectacular results, such as beating world champions in board games [14,15], reaching human-level performance in sophisticated computer games 16,17]], and controlling robotic locomotion [18,19].

Applying model-free RL to quantum control implies direct interaction of the learning agent with the controlled quantum system, which presents a number of unique challenges not typically encountered in classical environments. Quantum systems have large continuous state spaces that are only partially observable to the agent through measurements. For example, a pure qubit state

*vladimir.sivak@yale.edu
†michel.devoret@yale.edu

can be described as a point on a Bloch sphere, but a projective measurement of a qubit observable yields a random binary outcome. Qubits are often used as ancillary systems to control harmonic oscillators, in which case the underlying state space is formally infinite dimensional. Learning quantum control of such systems is akin to learning to drive a car with a single sensor that provides binary-valued feedback. The following question arises: Can classical model-free RL agents efficiently handle such "quantum-observable" environments?

The previous applications of RL to quantum control [20–39], which we survey in Sec. II, relied on a number of simplifying assumptions rendering the quantum control problem more tractable for the agent but severely limiting their experimental feasibility. These approaches provide the agent with the knowledge of a quantum state or rely on fidelity as a measure of optimization progress. Such requirements are at odds with the fundamental properties of quantum environments, stochasticity and minimalistic observability. Trying to meet these requirements in realistic experiments leads to a large sample size, e.g., 10^7 measurements to learn a single-qubit gate with only 16 parameters, as recently demonstrated in Ref. [40] using a quantum-state-aware agent that relied on tomography to obtain the quantum state. Other model-free approaches that view quantum control as a standard cost function optimization problem [6–10] are subject to similar limitations. Scaling such methods beyond one- or two-qubit applications is prohibitively expensive from a practical point of view.

In this paper, we develop a framework for model-free learning of quantum control policies, which is explicitly tailored to the stochasticity and minimalistic quantum observability. It does not rely on restrictive assumptions, such as a model of the system's dynamics, knowledge of a quantum state, or access to fidelity. By framing quantum control as a quantum-observable Markov decision process (QOMDP) [41], we consider each stochastic experimental realization as an episode of interaction of the learning agent with a controlled quantum system, after which the agent receives a binary-valued reward through a projective measurement. Instead of utilizing averaging, every such episode is performed with a different control policy, which is being continually updated by a small amount within a trust region with the help of the reward signal. This novel strategy of exploration of the policy space leads to excellent sample efficiency on challenging high-dimensional tasks, significantly outperforming widely used model-free methods.

To illustrate our approach with specific examples, we focus on the quantum control of a harmonic oscillator. Harmonic oscillators are ubiquitous physical systems, realized, for instance, as the motional degrees of freedom of trapped ions [42,43] or electromagnetic modes in superconducting circuits [44,45]. They are primitives for bosonic quantum error correction [46–48] and quantum sensing [49]. Universal quantum control of an oscillator is typically realized by coupling it to an ancillary nonlinear system, such as a qubit, with state-of-the-art fidelities in the 0.9–0.99 range in circuit quantum electrodynamics (QED) [50–52] and trapped ions [53]. In such a quantum environment, ancilla measurements with binary outcomes are the agent's only source of information about the quantum state in the vast unobservable Hilbert space and the only source of rewards guiding the learning algorithm.

For an oscillator-qubit system, we demonstrate learning of both unitary control and control with adaptive measurement-based quantum feedback. These types of control are special instances of a modular circuit-based framework, in which the quantum operation executed on a system is represented as a sequence of continuously parametrized control circuits, whose parameters are learned *in situ* with the help of a reward circuit. We show how to construct task-specific reward circuits that implement an experimentally feasible dichotomic positive operator-valued measure (POVM) on the oscillator and how to use its outcomes as reward bits in the classical training loop. We train the agent to prepare various nonclassical oscillator states, such as Fock states, Gottesman-Kitaev-Preskill (GKP) states [54], Schrödinger cat states, and binomial code states [55], and to execute gates on logical qubits encoded in an oscillator.

Although our demonstration is based on a simulated environment producing mock measurement outcomes, the RL agent that we developed (code available at Ref. [56]) is compatible with real-world experiments.

II. RELATED WORK

In recent years, multiple theoretical proposals have emerged around applying reinforcement learning to quantum control problems such as quantum state preparation [20–23,23–28] and feedback stabilization [29,30], the construction of quantum gates [31–33], design of quantum error correction protocols [34–37], and control-enhanced quantum sensing [38,39]. These proposals formulate the control problem in a way that avoids directly facing quantum observability and makes it more tractable for the RL agent. In simulated environments, this is possible, for example, by providing the agent with full knowledge of the system's quantum state, which supplies enough information for decision making [20,23–25,27,29,34,38,39]. Moreover, in the simulation, the distance to the target state or operation is known at every step of the quantum trajectory, and it can be used to construct a steady reward signal to guide the learning algorithm [23–25,38], thereby alleviating the well-known delayed reward assignment problem [12,13]. Taking RL a step closer towards quantum observability, some works only provide the agent with access to fidelities and expectation values of physical

observables in different parts of the training pipeline [21,26,28,57,58], which would still require a prohibitive amount of averaging in an experiment, a problem exacerbated by the iterative nature of the training process. Under these various simplifications, there are positive indications [23,31] that RL is able to match the performance of traditional gradient-based methods, albeit in situations where the agent or the learning algorithm has access to expensive or unrealistic resources. Therefore, such RL proposals are not compatible with efficient training in experiment, which is required in order to eliminate model bias from quantum control. To address this challenge, it is necessary to develop agents that learn directly from stochastic measurement outcomes or from low-sample estimators of physical observables. Initial steps towards this goal were studied in Refs. [22,30,59].

III. REINFORCEMENT LEARNING APPROACH TO QUANTUM CONTROL

A. Markov decision process

We begin by introducing several concepts from the field of artificial intelligence (AI). An intelligent agent is any device that can be viewed as perceiving its environment through sensors and acting upon that environment with actuators [60]. In RL [12,13], a subfield of AI, the interaction of the agent with its environment is usually described with a powerful framework of Markov decision process (MDP).

In the MDP framework, the agent-environment interaction proceeds in episodes consisting of a sequence of discrete time steps. At every time step t, the agent receives an observation $o_t \in \mathcal{O}$ containing some information about the current environment state $s_t \in \mathcal{S}$ and acts on the environment with an action $a_t \in \mathcal{A}$. This action induces a transition of the environment to a new state s_{t+1} according to a Markov transition function $\mathcal{T}(s_{t+1}|s_t, a_t)$. The agent selects actions according to a policy $\pi(a_t|h_t)$, which, in general, can depend on the history $h_t = o_{0:t}$ of all past observations made in the current episode. In the partially observable environment, observations are issued according to an observation function $O(o_t|s_t)$ and carry only limited information about the state. In the special case of a fully observable environment, the observation $o_t = s_t$ is a sufficient statistic of the past, which allows us to restrict the policy to a mapping from states to actions $\pi(a_t|s_t)$. Environments can be further categorized as discrete or continuous according to the structure of the state space $\mathcal{S}$, and as deterministic or stochastic according to the structure of the transition function $\mathcal{T}$. Likewise, policies can be categorized as discrete or continuous, according to the structure of the action space $\mathcal{A}$, and as deterministic or stochastic.

The agent is guided through the learning process by a reward signal $r_t \in \mathcal{R}$. The reward is issued to the agent after each action, but it cannot be used by the agent to decide on the next action. Instead, it is used by the learning algorithm to improve the policy. The reward signal is designed by a human supervisor according to the final goal, and it must indicate how good the new environment state is after the applied action. Importantly, it is possible to specify the reward signal for achieving a final goal without knowing what the optimal actions are, which is a major difference between reinforcement learning and more widely appreciated supervised learning. The goal of the learning algorithm is to find a policy π that maximizes the agent's utility function J, which in RL is taken to be the expectation $J = \mathbb{E}_\pi[R]$ of the reward accumulated during the episode, also known as a return $R = \sum_t r_t$.

Even from this brief description, it is clear that learning environments vary vastly in complexity from "simple" discrete, fully observable, deterministic environments, such as a Rubik's cube, to "difficult" continuous, partially observable, stochastic environments, such as those of self-driving cars. Where does quantum control land on this spectrum?

B. Quantum control as quantum-observable Markov decision process

To explain how quantum control can be viewed as a sequential decision problem, for concreteness we specialize the discussion to a typical circuit QED [45] experimental setup, depicted in Fig. 1, although our framework is independent of the physical platform. The agent is a program implemented in a classical computer controlling the quantum system. The quantum environment of the agent consists of a quantum harmonic oscillator, realized as an electromagnetic mode of the superconducting resonator, and an ancilla qubit, realized as the two lowest energy levels of a transmon [61]. Note the difference in the use of the term "environment," which in quantum physics refers to a dissipative bath coupled to a quantum system, while in our RL context, it refers to the quantum system itself, which is the environment of the agent.

It is convenient to abstract away the exact details of the control hardware and adopt the circuit model of quantum control. According to such an operational definition, the agent interacts with the environment by executing a parametrized control circuit in discrete steps, as illustrated in Fig. 1. On each step t, the agent receives an observation o_t and produces the action vector a_t of parameters of the control circuit to apply in the next time step. The agent-environment interaction proceeds for T steps, comprising an episode.

Compared to the typical classical, partially observable MDPs (POMDPs), there are two significant complications in the quantum case: (i) The quantum environment is minimally observable to the agent through projective ancilla measurements; i.e., the observations o_t carry no more than 1 bit of information, and (ii) the observation causes a random discontinuous jump in the underlying

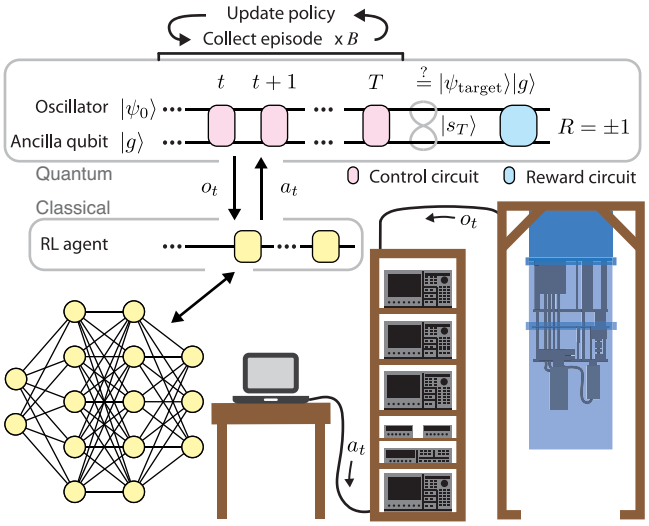

FIG. 1. Pipeline of classical reinforcement learning applied to a quantum-observable environment. The agent (yellow box), whose policy is represented with a neural network, is a program implemented in a classical computer controlling the quantum system. The quantum environment of the agent consists of a harmonic oscillator and its ancilla qubit, implemented with superconducting circuits and cryogenically cooled in the dilution refrigerator. The goal of the agent is to prepare the target state $|\psi_{\text{target}}\rangle$ of the oscillator after T time steps, starting from initial state $|\psi_0\rangle$. Importantly, the agent does not have access to the quantum state of the environment; it can only observe the environment through intermediate projective measurements of the ancilla qubit yielding binary outcomes o_t. The agent controls the environment by producing, at each time step, the action vector a_t of parameters of the control circuit (pink box). The reward R for the RL training is obtained by executing the reward circuit (blue box) on the final state $|s_T\rangle$ prepared in each episode. This circuit is designed to probabilistically answer the following question: "Is the prepared state $|s_T\rangle$ equal to $|\psi_{\text{target}}\rangle|g\rangle$?" A batch of B episodes is collected per training epoch and used in the classical optimization loop to update the policy.

environment state. While, in principle, classical POMDPs could have such properties, they arise more naturally in the quantum case. Historically, RL was sometimes benchmarked in stochastic but always richly observable environments, and it is therefore an open question whether existing RL algorithms are well suited for quantum environments with properties (i) and (ii). There is also a fundamental question of whether classical agents can efficiently, in the algorithmic complexity sense, learn compressed representations of the latent quantum states producing the observations and if such representations are necessary for learning quantum control policies. Recognizing some of these difficulties, Ref. [41] introduced the quantum-observable Markov decision process (QOMDP), a term we will adopt to describe our quantum control framework.

We use the Monte Carlo wave-function method [62] to simulate the quantum environment of the agent. For the environment consisting of an oscillator coupled to an ancilla qubit and isolated from the dissipative bath, the most general QOMDP has the following specifications:

(1) State space is the joint Hilbert space of the oscillator-qubit system, which in our simulation corresponds to $\mathcal{S} = \{|s\rangle \in \mathbb{C}^2 \otimes \mathbb{C}^N, \langle s|s\rangle = 1\}$, with $N = 100$ being the oscillator Hilbert space truncation in the photon number basis.

(2) Observation space $\mathcal{O} = \{-1, +1\}$ is a set of possible measurement outcomes of the qubit σ_z operator. If the control circuit contains a qubit measurement, the observation function is given by the Born rule of probabilities. If the control circuit does not contain a measurement, the observation is a constant, which we take to be $o_t = +1$. We refer to the former as measurement-based feedback control and the latter as unitary control.

In other approaches [20,23–25,27,29,34,38,39], an observation is a quantum state itself $o_t = |s_t\rangle$, which is not naturally compatible with real-world experiments. It could be obtained through quantum-state tomography [40], but this would result in exponential scaling of the training sample complexity with system size.

(3) Action space $\mathcal{A} = \mathbb{R}^{|\mathcal{A}|}$ is the space of parameters a of the control circuit. It generates the set $\{\mathcal{K}[a]\}$ of continuously parametrized Kraus maps. If the control circuit contains a qubit measurement, then each map $\mathcal{K}[a]$ consists of two Kraus operators $K_\pm[a]$ satisfying the completeness relation $K_+^\dagger[a]K_+[a] + K_-^\dagger[a]K_-[a] = I$ and corresponding to observations ± 1. If the control circuit does not contain a measurement, then the map consists of a single unitary operator $K_0[a]$.

(4) State transitions happen deterministically according to $|s_{t+1}\rangle = K_0[a_t]|s_t\rangle$ if the control circuit does not contain a measurement and otherwise stochastically according to $|s_{t+1}\rangle = K_\pm[a_t]|s_t\rangle/\sqrt{p_\pm}$, with probabilities $p_\pm = \langle s_t|K_\pm^\dagger[a_t]K_\pm[a_t]|s_t\rangle$.

In this paper, we do not consider the coupling of a quantum system to a dissipative bath, but it can be incorporated into the QOMDP by expanding the Kraus maps to include uncontrolled quantum jumps of the state $|s_t\rangle$ induced by the bath. This would lead to more complicated dynamics, but since the quantum state and its transitions are hidden from the agent, nothing would change in the RL framework.

In the traditional simulation-based approach to quantum control, the model for $\mathcal{K}[a]$ is specified, for example, through the system's Hamiltonian and Schrödinger equation, allowing for gradient-based optimization of the cost function [1–5]. In contrast, in our approach, the Kraus map $\mathcal{K}[a]$ is not modeled. Instead, the experimental apparatus implements $\mathcal{K}[a]$ exactly. In this case, the optimization proceeds at a higher level by trial-and-error learning of the patterns in the action-reward relationship. This ensures that the learned control sequence is free of model bias.

In practice, common contributions to model bias come from frequency- and power-dependent pulse distortions in the control lines [63,64], higher-order nonlinearities, coupling to spurious modes, etc. Simulation-based approaches often attempt to compensate for model bias by introducing additional terms in the cost function, such as penalties for pulse power and bandwidth, weighted with somewhat arbitrarily chosen coefficients, or finding policies that are first-order insensitive to deviations in system parameters [65]. In contrast, our RL agent will learn the relevant constraints automatically since it optimizes the true unbiased objective incorporated into the reward.

As shown in Fig. 1, the reward in our approach is produced by following the training episode with the reward circuit. This circuit realizes a dichotomic POVM on the oscillator, whose binary outcome probabilistically indicates whether the applied action sequence implements the desired quantum operation. Since the agent's goal is to maximize the expectation $J = \mathbb{E}[R]$, we require that in the state preparation QOMDPs, the reward circuit is designed to satisfy the condition

$$\underset{|\psi\rangle}{\operatorname{argmax}}\ \mathbb{E}[R] = |\psi_{\text{target}}\rangle, \tag{1}$$

where expectation is taken with respect to the sampling noise in reward measurements when the state $|\psi\rangle|g\rangle$ is supplied at the input to the reward circuit.

In circuit QED, dichotomic POVMs are realized through unitary operation on the oscillator-qubit system followed by a projective qubit measurement in the σ_z basis. Since the reward measurement, in general, will disrupt the quantum state, we only apply the reward circuit at the end of the episode and use the reward $r_{t<T} = 0$ at all intermediate time steps. Hence, from now on, we will omit the time-step index and refer to the reward as simply $R \equiv r_T$. Such delayed rewards are known to be particularly challenging for RL agents because they need to make multiple action decisions during the episode, while the reward only informs about whether the complete sequence of actions was successful but does not provide feedback on the individual actions.

A common choice of reward R in other approaches [20,21,23,25–28,31–33,40] is the fidelity of the executed quantum operation. The fidelity oracle, often assumed to be freely available, would translate into time-consuming averaging in experiments involving quantum systems with high-dimensional Hilbert space, and it is therefore prohibitively expensive from a practical point of view.

Clearly, quantum control is a "difficult" decision process according to a rough categorization outlined in Sec. III A. One may compare it to driving a car blind with a single sensor that provides binary-valued feedback instead of a rich visual picture of the surroundings. In the following subsection, we describe our approach to solving QOMDPs through policy gradient RL.

C. Solving quantum control through policy gradient reinforcement learning

The solution to a POMDP is a policy $\pi(a_t|h_t)$ that assigns a probability distribution over actions to each possible history $h_t = o_{0:t}$ that the agent might see. In large problems, it is unfeasible to represent the policy as a lookup table, and instead, it is convenient to parametrize it using a powerful function approximator such as a deep neural network [14,16,66]. As an additional benefit, this representation allows the learning agent to generalize via parameter sharing to histories it has never encountered during training. We refer to such neural network policies as π_θ, where θ represents the network parameters. It is advantageous to adopt recurrent network architectures, such as the long short-term memory (LSTM) [67], in problems with variable-length inputs. In this work, we use neural networks with a LSTM layer and several fully connected layers.

The output of the policy network is the mean $\mu_\theta[h_t]$ and diagonal covariance $\sigma^2_\theta[h_t]$ of the multivariate Gaussian distribution from which the action a_t is sampled on every time step, as depicted in Fig. 1. The stochasticity of the policy during the training ensures a balance between exploration of new actions and exploitation of the current best estimate μ_θ of the optimal action. Typically, as training progresses, the agent learns to reduce the entropy of the stochastic policy, eventually converging to a near-deterministic policy. After the training is finished, the deterministic policy is obtained by choosing the optimal action μ_θ.

In application to QOMDPs, such a stochastic action-space exploration strategy means that every experimental run is performed with a different policy candidate, which is evaluated with a binary reward measurement. Instead of spending the sample budget on increasing the evaluation accuracy for any given policy candidate through averaging, our strategy is to spend this budget on evaluating more policy candidates, albeit with the minimal accuracy. Such a strategy is explicitly tailored to the stochasticity and minimalistic observability of quantum environments, and is conceptually rather different from widely used model-free optimization methods that crucially rely on averaging to suppress noise in the cost function, as we further discuss in the Appendix B.

Policy gradient reinforcement learning [12,13] provides a set of tools for learning the policy parameters θ guided by the reward signal. Even though the binary-valued reward R is a nondifferentiable random variable sampled from episodic interactions with the environment, its expectation J depends on the policy parameters θ, and it is therefore differentiable. The basic working principle of the policy gradient algorithms is to construct an empirical estimator g_k of the gradient of performance measure $\nabla_\theta J(\pi_\theta)|_{\theta=\theta_k}$ based on a batch of B episodes of experience collected in the environment following the current stochastic policy π_{θ_k},

and then perform a gradient ascent step on the policy parameters $\theta_{k+1} = \theta_k + \alpha g_k$, where α is the learning rate. This data collection and the subsequent policy update comprise a single epoch of training.

Various policy gradient RL algorithms differ in their construction of the gradient estimator. In this work, we use the proximal policy optimization algorithm (PPO) [68], whose brief summary is included in the Supplemental Material [69]. PPO was developed to cure sudden performance collapses often observed when using high-dimensional neural network policies. It achieves this by discouraging large policy updates (hence "proximal"), inspired by ideas from trust region optimization. The stability of PPO is essential in stochastic environments, motivating our choice of this algorithm for solving QOMDPs.

As described above, the learning process is a guided search in the policy space, where the guiding signal is the reward assigned to each attempted action sequence. Since in the state preparation QOMDP the goal is to approach arbitrarily close to the target state that resides in a continuous state space, it is tempting to think that the guiding signal needs to be of high resolution, i.e., assign different rewards to policies of different qualities, with the reward difference being indicative of the quality difference. This condition is certainly satisfied by using fidelity as a reward [20,21,23,25–28,31–33,40]. In contrast, our reward-circuit-based approach breaks this condition but promises high experimental sample efficiency by virtue of not having to perform expensive fidelity estimation. However, it is not obvious that stochastic ± 1 outcomes of the reward circuits are sufficient to navigate a continuous policy space and converge at all, not to mention reaching a high fidelity. For example, consider that for two policies with fidelities $\mathcal{F}_1 > \mathcal{F}_2$, in our approach, it is possible to receive the rewards $R_1 = -1 < R_2 = +1$ because of the measurement sampling noise, leading to the incorrect contribution to policy gradient. By probabilistically comparing multiple policy candidates and performing small updates within the trust region, our proximal policy optimization is able to successfully cope with such a highly stochastic learning problem.

The next section is devoted to empirically proving that our approach indeed leads to stable learning convergence, i.e., that the agent's performance gradually improves to a desired level and does not collapse or stagnate. We demonstrate this by training the agent to solve challenging state preparation instances.

We also provide a simple introductory example illustrating the basic principles of our approach in Appendix A.

IV. RESULTS

Currently, direct pulse shaping with gradient ascent pulse engineering (GRAPE) is a dominant approach to quantum state preparation in circuit QED [48–50]. Nevertheless, a modular approach based on repetitive application of a parametrized control circuit has several advantages [51,52]. First, thanks to a reduced number of parameters, the modular approach is less likely to overfit and can generalize better under small environment perturbations. In addition, each gate in the module can be individually tested and calibrated. Finally, the modular approach is physically motivated and more interpretable, leading to a better understanding of the solution.

Our RL approach is compatible with any parametrized control circuit, including piecewise constant parametrization used in direct pulse shaping. In this work, for concreteness, we make the particular choice of a control circuit based on the universal gate set consisting of the selective number-dependent arbitrary phase gate $\mathrm{SNAP}(\varphi)$ and displacement $D(\alpha)$ [70]:

$$\mathrm{SNAP}(\varphi) = \sum_{n=0}^{\infty} e^{i\varphi_n} |n\rangle\langle n|, \tag{2}$$

$$D(\alpha) = \exp(\alpha a^\dagger - \alpha^* a). \tag{3}$$

In practice, this gate set has been realized in the strong dispersive limit of circuit QED [52,71]. Displacements $D(\alpha)$ are implemented with resonant driving of the oscillator, while the Berry phases φ_n in the $\mathrm{SNAP}(\varphi)$ gate are created by driving the qubit resonantly with the $|g\rangle|n\rangle \leftrightarrow |e\rangle|n\rangle$ transition. Recently, it was demonstrated that SNAP can be made first-order path independent with respect to ancilla qubit decay [72,73]. Furthermore, a linear scaling of the circuit depth T with the state size $\langle n\rangle$ can be achieved for this approach [74], while many interesting experimentally achievable states can be prepared with just $T \sim 5$. Inspired by this finding, we parametrize our unitary control circuit as $D^\dagger(\alpha)\mathrm{SNAP}(\varphi)D(\alpha)$; see Fig. 2(a).

In Secs. IV A–IV C, our aim is to demonstrate that model-free RL is feasible; i.e., the learning converges to high-fidelity protocols in a realistic number of training episodes. To isolate the learning aspect of the problem, in Secs. IV A–IV C, we use perfect gate implementations acting on the Hilbert space as intended by Eqs. (2) and (3). However, the major power of the model-free paradigm is the ability to utilize available controls even when they do not produce the expected effect, tailoring the learned actions to the unique control imperfections present in the system. We focus on this aspect in Sec. IV D by training the agent with an imperfectly implemented SNAP. Moreover, the advantage of model-free RL compared to other model-free optimization methods is that it can efficiently solve problems requiring adaptive decision making [14–19]. We leverage this advantage of RL in Sec. IV D to learn adaptive measurement-based quantum feedback strategies compensating for imperfect SNAP implementation. Finally, in Appendix E, we demonstrate learning of gates for logical qubits encoded in an oscillator.

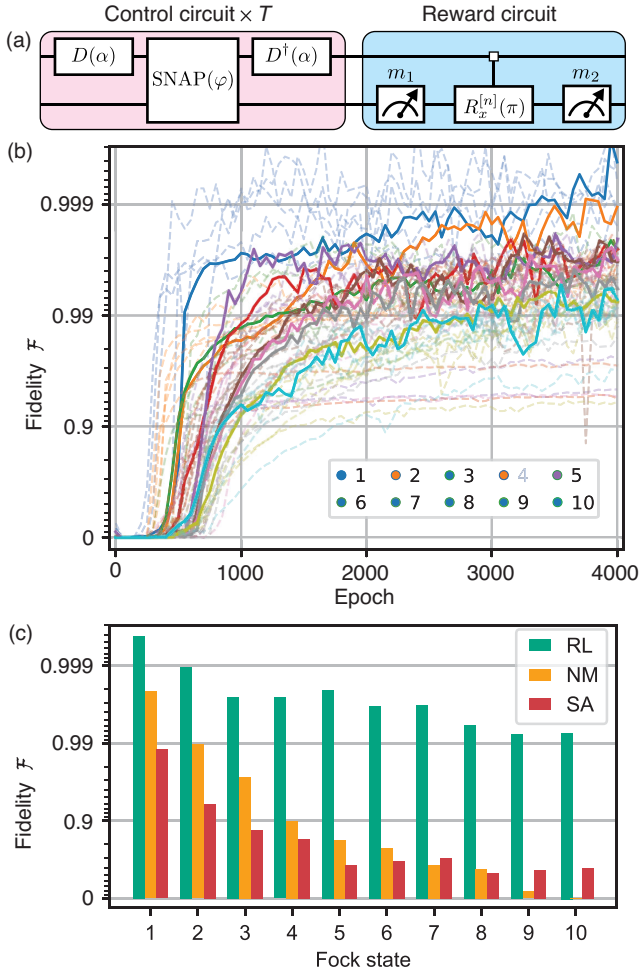

FIG. 2. Preparation of Fock states $|1\rangle, \ldots, |10\rangle$. (a) Parametrized control circuit (pink) and Fock reward circuit (blue). The reward circuit contains a selective π pulse on the qubit, conditioned on having n photons in the oscillator. (b) Evaluation of the training progress. The background trajectories correspond to six random seeds for each state; solid lines show the trajectory with the highest final fidelity. (c) Summary of comparison of different model-free approaches on the task of Fock state preparation. We perform extensive hyperparameter tuning for all three approaches, as described in Sec. IV A for RL, and in Appendix B for Nelder-Mead (NM) and simulated annealing (SA). All approaches are constrained to the same total sample size of $M_{\text{tot}} = 4 \times 10^6$. The displayed final fidelity is the highest achieved among six tested random seeds.

A. Preparation of oscillator Fock states

One central question in our RL approach is how to assign a reward R to the agent by performing a measurement on the prepared state $|s_T\rangle$. To satisfy Eq. (1), it is sufficient to design the reward circuit in such a way that $\mathbb{E}[R] = f(\mathcal{F})$, where f is any monotonously increasing function of fidelity $\mathcal{F}$ to the target state. Although this is not necessary, we find it to be a useful guiding principle. For example, the most efficient choice is to generate R as an outcome of a measurement with POVM $\{\Omega_{\text{target}}, I - \Omega_{\text{target}}\}$, where $\Omega_{\text{target}} = |\psi\rangle\langle\psi|_{\text{target}}$ is the target projector. This POVM maximizes the distinguishability of the target state from all other states [75]. We refer to such a reward as the target projector reward. If the measurement outcomes associated with this POVM are ± 1, then the reward will satisfy $\mathbb{E}[R] = 2\mathcal{F} - 1$.

In the strong dispersive limit of circuit QED [76], a dichotomic POVM measurement required for the target projector reward can be routinely realized for an important class of nonclassical states known as Fock states $|n\rangle$, which are eigenstates of the photon number operator. To learn the preparation of such states, we use the "Fock reward circuit" shown in Fig. 2(a).

All reward circuits considered in this work contain two ancilla measurements. If the SNAP is ideal as in Eq. (2), the qubit will remain in $|g\rangle$ after the control sequence, and the outcome of the first measurement will always be $m_1 = 1$, which is the case in Secs. IV A–IV C and in Appendix E. However, in a real experimental setup, residual entanglement between the qubit and oscillator can remain. Therefore, in general, the first measurement serves to disentangle them. The second measurement with outcome m_2 is used to produce the reward. In the Fock reward circuit, this is done according to the rule $R = -m_2$.

The training episodes begin with the oscillator in vacuum $|\psi_0\rangle = |0\rangle$ and the ancilla qubit in the ground state $|g\rangle$. Episodes follow the general template shown in Fig. 1, in which the control circuit is applied for $T = 5$ time steps, followed by the Fock reward circuit. The SNAP gate is truncated at $\Phi = 15$ levels, leading to the $(15 + 2)$-dimensional parametrization of the control circuit and amounting to 85 real parameters for the full control sequence. In our approach, the choice of the circuit depth T and the action-space dimension $|\mathcal{A}| = \Phi + 2$ needs to be made in advance, which requires some prior understanding of the problem complexity. In this example, we choose $T = 5$ and $\Phi = 15$ for all Fock states $|1\rangle, ..., |10\rangle$ to ensure a fair comparison of the convergence speed, but, in principle, the states with lower n can be prepared with shorter sequences [70,71]. An automated method for selecting the circuit depth was proposed in Ref. [74], and it can be utilized here to make an educated guess of T.

The action vectors are sampled from the Gaussian distribution produced by the deep neural network with one LSTM layer and two fully connected layers, representing the stochastic policy. The neural network input is only the "clock" observation (one-hot encoding of the step index t) since there are no measurement outcomes in the unitary control circuit. The agent is trained for 4×10^3 epochs with batches of $B = 10^3$ episodes per epoch. This amounts to a sample size of $M_{\text{tot}} = 4 \times 10^6$ experimental runs. The total time budget of the training is split between (i) experience collection, (ii) optimization of the neural network, and (iii) communication and instruments reinitialization. We estimate that with the help of active oscillator reset [77], the

experience collection time in experiment can be as short as 10 minutes in total for such training (assuming 150 μs duty cycle per episode). Our neural network is implemented with TensorFlow [78] on a NVIDIA Tesla V100 graphics processing unit (GPU). The total time spent updating the neural network parameters is 10 minutes in total for such training. The real experimental implementation will likely be limited by instrument reinitialization [9]. This time budget puts our proposal within the reach of current technology.

Throughout this paper, we use the fidelity $\mathcal{F}$ only as an evaluation metric to benchmark the agent, and it is not used anywhere in the training loop. If desired, in experiment, the training epochs can be periodically interleaved with evaluation epochs to perform fidelity estimation [79,80] for the deterministic version of the current stochastic policy. Other metrics can also be used to monitor the training progress without interruption, such as the return and entropy of the stochastic policy.

The agent benchmarking results for this QOMDP are shown in Fig. 2(b). They indicate that our stochastic action-space exploration strategy is not only able to converge but also yields high-fidelity solutions within a realistic number of experimental runs. The agent was able to reach $\mathcal{F} > 0.99$ for all Fock states and $\mathcal{F} > 0.999$ for Fock state $|1\rangle$.

Such stable convergence in a stochastic setting is possible with proximal policy optimization because after every epoch, the policy distribution only changes by a small amount within a trust region. This working principle is in stark contrast with popular optimization algorithms such as the NM simplex search [6–8] or SA [40], where each update of the simplex (in NM) or the state (in SA) can result in a drastically different policy. As a result, both of these approaches perform poorly on high-dimensional problems with the stochastic cost function, as shown in Appendix B and summarized in Fig. 2(c). When constrained to the same total number of experimental runs $M_{\mathrm{tot}} = 4 \times 10^6$ as in Fig. 2(b), NM is only able to find solutions with $\mathcal{F} > 0.99$ for Fock states $|1\rangle$ and $|2\rangle$ and SA only for Fock state $|1\rangle$.

Despite its low resolution, the target projector reward represents the most informative POVM from the perspective of state certification [75], and it results in efficient learning of state preparation protocols. However, for most target states, it will be unfeasible to experimentally implement such POVM in a trustworthy way. Recall that in circuit QED, any dichotomic POVM on the oscillator is implemented with a unitary operation on the oscillator-qubit system and a subsequent qubit measurement in the σ_z basis. The trustworthiness requirement implies that this unitary operation can be independently calibrated to high accuracy because errors in its implementation can bias the reward circuit and, as a result, bias the learning objective of the agent. For example, in the Fock reward circuit in Fig. 2(a), the unitary is a simple photon-number-selective qubit flip whose calibration is relatively straightforward. Therefore, we consider the Fock reward as a feasible and trustworthy instance of the target projector reward.

In a more general case, when a target projector reward is unfeasible to implement, consider the following probabilistic measurement strategy. Let $\{\Omega_k\}$ be a parametrized set of POVM elements that can be realized in a trustworthy way. To implement a reward measurement, in each episode, we first sample the parameter k from some probability distribution $P(k)$ and then implement a dichotomic POVM $\{\Omega_k, I - \Omega_k\}$ with associated reward $R = \pm R_k$. One can view such a reward scheme as probabilistically testing different properties of the prepared state, instead of testing directly whether it is equal to the target state. The scale R_k of the binary reward is chosen according to the importance of each such property. Note that in such a reward scheme, the expectation in Eq. (1) is taken with respect to both the sampling of POVMs and the sampling of measurement outcomes.

In Secs. IV B and IV C, we consider examples of such probabilistic reward measurement schemes, with further examples relevant for other physical systems included in Appendix D.

B. Preparation of stabilizer states

The class of stabilizer states is of particular interest for quantum error correction [81]. A state is a stabilizer state if it is a unique joint eigenvalue-1 eigenstate of a commutative stabilizer group. To demonstrate learning stabilizer state preparation in an oscillator, we train the agent to prepare a grid state, also known as the Gottesman-Kitaev-Preskill (GKP) state [54]. Grid states were originally introduced for encoding a 2D qubit subspace into an infinite-dimensional Hilbert space of an oscillator for bosonic quantum error correction, and they were subsequently recognized to be valuable resources for various other quantum applications. In particular, the 1D version of the grid state, which we consider here, can be used for sensing both real and imaginary parts of a displacement simultaneously [82,83].

An infinite-energy 1D grid state is a Dirac comb $|\psi_0^{\mathrm{GKP}}\rangle \propto \sum_{t\in\mathbb{Z}} D(t\sqrt{\pi})|0_x\rangle$, where $|0_x\rangle$ is a position eigenstate located at $x = 0$. The generators of a stabilizer group for such a state are $S_{x,0} = D(\sqrt{\pi})$ and $S_{p,0} = D(i\sqrt{\pi})$. The finite-energy version of this state $|\psi_\Delta^{\mathrm{GKP}}\rangle$ can be obtained with generators $S_{x,\Delta} = E_\Delta S_{x,0} E_\Delta^{-1}$ and $S_{p,\Delta} = E_\Delta S_{p,0} E_\Delta^{-1}$, where $E_\Delta = \exp(-\Delta^2 a^\dagger a)$ is the envelope operator and Δ determines the degree of squeezing in the peaks of the Dirac comb and the extent of the grid envelope.

To learn the preparation of such a GKP state, consider a probabilistic reward measurement scheme based on a set $\{\Omega_k\}$, with $k = x,\ p$ of POVM elements, which are the projectors onto the $+1$ eigenspaces of stabilizer generators $S_{x/p,\Delta}$. The direction of the stabilizer displacement (along x or p quadrature) is sampled uniformly, and the scale of reward is $R_k = 1$ for each direction. In this scheme, there is

no simple relation between $\mathbb{E}[R]$ and $\mathcal{F}$, but the condition (1) is satisfied. In contrast, for a multiqubit system with a finite stabilizer group, it is possible to construct a scheme in which the expectation of reward is a monotonous function of fidelity by sampling uniformly from the full stabilizer group (see Appendix D).

The infinite-energy stabilizers $S_{x/p,0}$ are unitary and can be measured in the oscillator-qubit system with the standard phase estimation circuit [84], as was experimentally demonstrated with trapped ions [85] and superconducting circuits [86]. On the other hand, the finite-energy stabilizers $S_{x/p,\Delta}$ are not unitary nor Hermitian. Recently, an approximate circuit for generalized measurement of $S_{x/p,\Delta}$ was proposed [87,88] and realized with trapped ions [88]. Our stabilizer reward circuit, shown in Fig. 3(a), is based on these proposals. The measurement outcome m_2, obtained in this circuit, is administered as a reward $R = m_2$. Since this circuit only approximates the desired POVM, such a reward will only approximately satisfy $\mathbb{E}[R] = (\langle S_{x,\Delta}\rangle + \langle S_{p,\Delta}\rangle)/2$ and fulfill the condition (1). Nevertheless, the agent that strives to maximize such a reward will learn to prepare an approximate $|\psi_\Delta^{\mathrm{GKP}}\rangle$ state.

After choosing the reward circuit, we need to properly constrain the control circuit. Grid states have a large photon number variance $\sqrt{\mathrm{var}(n)} \approx \langle n\rangle \approx 1/(2\Delta^2)$; hence, preparation of such states requires a large SNAP truncation Φ. However, increasing the action-space dimension $|\mathcal{A}| = \Phi + 2$ can result in less stable and efficient learning. As a compromise, we choose $\Phi = 30$ and $T = 9$, amounting to 288 real parameters for the full control sequence.

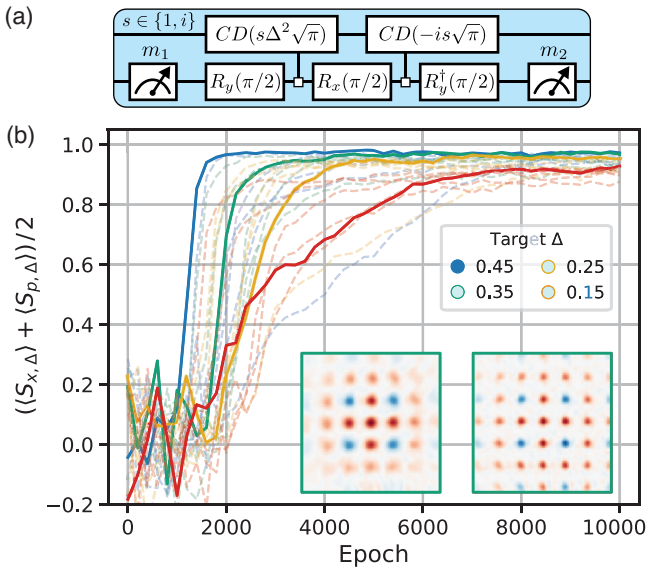

FIG. 3. Preparation of grid states. (a) Stabilizer reward circuit for the target state $|\psi_\Delta^{\mathrm{GKP}}\rangle$. The circuit makes use of the conditional displacement gate $CD(\alpha) = D(\sigma_z\alpha/2)$. The control circuit is the same as in Fig. 2(a). (b) Evaluation of the training progress. The background trajectories correspond to six random seeds for each state; solid lines show the trajectory with the highest final stabilizer value. Inset: example Wigner functions of the states prepared by the agent after 10,000 epochs of training.

The agent benchmarking results for this QOMDP are shown in Fig. 3(b), with the average stabilizer value as the evaluation metric [measured with the approximate circuit from Fig. 3(a)]. For a perfect policy, the stabilizers would saturate to $+1$, but it is increasingly difficult to satisfy this requirement for target states with smaller Δ because of a limited SNAP truncation and circuit depth. Nevertheless, our agent successfully copes with this task. Example Wigner functions of the states prepared by the agent after 10,000 epochs of training are shown as insets.

Learning state preparation with a probabilistic reward measurement scheme is generally less efficient than with a target projector reward because individual reward bits carry only partial information about the state. However, in principle, if stabilizer measurements can be realized in a quantum nondemolition way, this opens a possibility of acquiring the values of multiple commuting stabilizers after every episode, thereby increasing the signal-to-noise ratio (SNR) of the reward signal.

Reward circuits in Secs. IV A and IV B are designed for special classes of states. Next, we consider how to construct a reward circuit applicable to arbitrary states.

C. Preparation of arbitrary states

In the general case, we aim to construct an unbiased estimator of fidelity $\mathcal{F}$ based on a measurement scheme that is (i) tomographically complete, (ii) feasible to implement in a given experimental platform, and (iii) trustworthy. The requirement (i), in combination with universality of the control circuit, is necessary to guarantee that arbitrary states can, in principle, be prepared with our approach. However, it is not sufficient by itself and needs to be supplemented with requirements (ii) and (iii) to ensure practical feasibility.

In the strong dispersive limit of circuit QED, the Wigner tomography is a canonical example satisfying all three requirements above [89]. The Wigner function is defined on the oscillator phase space with coordinates $\alpha \in \mathbb{C}$, and it is given as the expectation value of the "displaced parity" operator $W(\alpha) = (2/\pi)\langle\Pi_\alpha\rangle$, where $\Pi_\alpha = D(\alpha)\Pi D^\dagger(\alpha)$, and $\Pi = e^{i\pi a^\dagger a}$ is the photon number parity. Hence, for the probabilistic reward measurement scheme based on the Wigner function, we consider a continuously parametrized set of POVM elements $\{\Omega_\alpha\}$, where $\Omega_\alpha = (I + \Pi_\alpha)/2$ is a projector onto $+1$ (even) eigenspace of the displaced parity operator.

Next, we need to determine the probability distribution $P(\alpha)$ according to which the POVMs are samples from the set $\{\Omega_\alpha\}$ for reward evaluation. To this end, we derive the estimator of fidelity based on the Monte Carlo importance sampling of the phase space:

$$\mathcal{F} = \pi \int d^2\alpha W(\alpha) W_{\text{target}}(\alpha) \tag{4}$$

$$= 2 \mathop{\mathbb{E}}_{\alpha\sim P} \mathbb{E}_{\psi} \left[\frac{1}{P(\alpha)} \Pi_\alpha W_{\text{target}}(\alpha) \right], \tag{5}$$

where points α are sampled according to an arbitrary probability distribution $P(\alpha)$, which is nonzero where $W_{\text{target}}(\alpha) \neq 0$. The estimator (5) leads to the following scheme, dubbed the "Wigner reward": First, the phase-space point α is generated with rejection sampling, as illustrated in Fig. 4(b), and then the displaced parity Π_α is measured, corresponding to the reward circuit shown in Fig. 4(a). The reward is then assigned according to the rule $R = R_\alpha m_2$, where $R_\alpha = \{2c/P(\alpha)\} W_{\text{target}}(\alpha)$ is chosen to reflect the importance of a sampled phase-space point, and $c > 0$ is an arbitrary scaling factor. Such a reward satisfies $\mathbb{E}[R] = c\mathcal{F}$ according to Eq. (5) but only requires a single binary tomography measurement per policy candidate.

The estimator (5) is unbiased for any $P(\alpha)$, but its variance can be reduced by choosing $P(\alpha)$ optimally. The lowest variance is achieved with $P(\alpha) \propto |W_{\text{target}}(\alpha)|$, as shown in Appendix C. Such a choice also helps to stabilize the learning algorithm since it conveniently leads to rewards $R = m_2 \operatorname{sgn} W_{\text{target}}(\alpha)$ of equal magnitude $|R| = 1$, where we made a proper choice of the scaling factor c.

We investigate the agent's performance with Wigner reward circuit for (i) preparation of the Schrödinger cat state $|\psi_{\text{target}}\rangle \propto |\beta\rangle + |-\beta\rangle$ with $\beta = 2$ in $T = 5$ steps, shown in Fig. 4(c), and (ii) preparation of the binomial code state $|\psi_{\text{target}}\rangle \propto \sqrt{3}|3\rangle + |9\rangle$ [55] in $T = 8$ steps, shown in Fig. 4(d). In contrast to target projector and stabilizer rewards that asymptotically lead to a reward of $+1$ for optimal policy, the Wigner reward remains stochastic even under the optimal policy. Since in this case it is impossible to find the policy that would systematically produce a reward of $+1$, for some states, the agent converges to policies of intermediate fidelity (green line). To increase the SNR of the Wigner reward, we evaluate every policy candidate with reward circuits corresponding to 1, 10, and 100 different phase-space points, doing a single measurement per point and averaging the obtained measurement outcomes to generate the reward R. The results show that the increased reward SNR allows us to reach higher fidelity, albeit at the expense of increased sample size. We expect that in the limit of infinite averaging, the training would proceed as if the fidelity $\mathcal{F}$ was directly available to be used as a reward (blue line).

We observe notable variations in convergence speed and saturation fidelity depending on the choice of hyperparameters, which is typical of reinforcement learning. A lot of progress has been made in developing robust RL algorithms applicable to a variety of tasks without extensive problem-specific hyperparameter tuning [15,16], but this

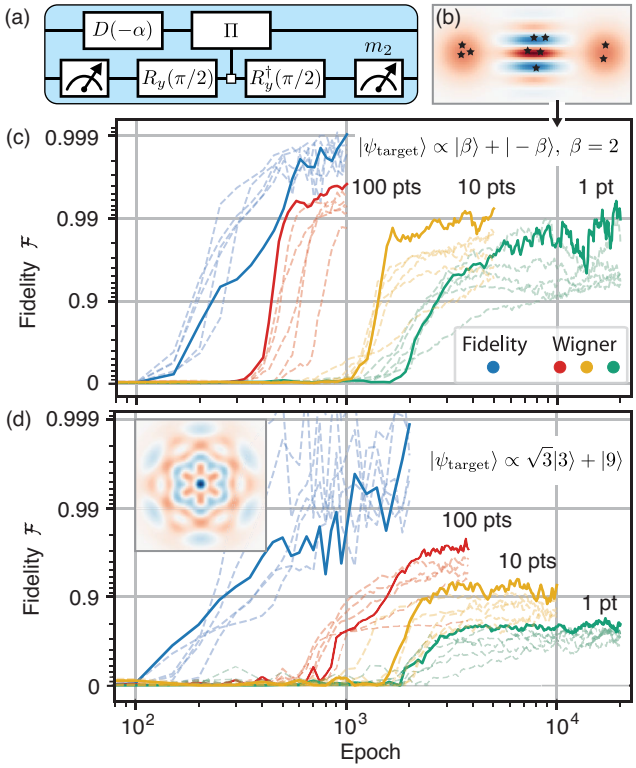

FIG. 4. Preparation of arbitrary states. (a) Wigner reward circuit based on the measurement of the photon number parity. In this circuit, the conditional parity gate corresponds to $|g\rangle\langle g| \otimes I + |e\rangle\langle e| \otimes \Pi$. (b) Wigner function of the cat state $|\psi_{\text{target}}\rangle \propto |\beta\rangle + |-\beta\rangle$, with $\beta = 2$. Scattered stars illustrate phase-space sampling of points α for the Wigner reward. (c) Evaluation of the training progress for the cat state. The background trajectories correspond to six random seeds for each setting; solid lines show the trajectory with the highest final fidelity. The Wigner reward is obtained by sampling 1, 10, and 100 different phase-space points, doing a single measurement per point, and averaging the obtained measurement outcomes to improve the resolution and achieve a higher convergence ceiling. For the blue curves, the fidelity $\mathcal{F}$ is used as a reward, representing the expected performance in the limit of infinite averaging. (d) Evaluation of the training progress for the binomial code state $|\psi_{\text{target}}\rangle \propto \sqrt{3}|3\rangle + |9\rangle$, whose Wigner function is shown in the inset.

still remains a major open problem in the field. The list of hyperparameters used in all our training examples can be found in the Supplemental Material [69]. Even with the optimal choice of hyperparameters, there is no rigorous guarantee of convergence—a problem plaguing all heuristic optimization methods in nonconvex spaces. In the presented examples, we plot learning trajectories corresponding to several random seeds to demonstrate that the probability of getting stuck with a suboptimal solution is small.

This demonstration shows that arbitrary-state preparation is, in principle, possible with our approach, as long as a tomographically complete reward measurement scheme is

available in a given physical system. In Appendix D, we provide fidelity estimators based on the characteristic function, enabling training for arbitrary-state preparation in trapped ions and multiqubit systems.

Examples considered in Secs. IV A–IV C already demonstrate the model-free aspect of our approach despite the perfect gate implementations in the underlying simulation of the quantum-state evolution. In the following example, we demonstrate this aspect more explicitly by training the agent on a system with imperfect SNAP. In addition, the next example highlights the potential of RL for measurement-based feedback control.

D. Learning adaptive quantum feedback with imperfect controls

Many quantum control experiments with circuit QED systems claim decoherence-limited fidelity [50,71]. The effect of decoherence on the quantum operation can be decreased by reducing the execution time. However, this would involve controls with a wider spectrum and larger amplitude, pushing the system to the limits where model assumptions are no longer valid. Therefore, such experiments are decoherence limited instead of model-bias limited only by choice. Recent experiments that push quantum control towards faster implementation [51,52] reveal that significant parts of the error budget cannot be accounted for by common and well-understood theoretical models, making the problem of model bias explicit. Model-free optimization will become an indispensable tool to achieve higher experimental fidelity despite the inability to capture the full complexity of a quantum system with a simple model.

To provide an example of this effect, we consider again a SNAP-displacement control sequence. In the oscillator-qubit system with dispersive coupling $H_c/h = \frac{1}{2}\chi a^\dagger a\sigma_z$, the Berry phases φ_n in Eq. (2) are created through photon-number-selective qubit rotations:

$$\mathrm{SNAP}(\varphi) = \sum_n |n\rangle\langle n| \otimes R_{\pi-\varphi_n}(\pi)R_0(\pi), \qquad (6)$$

where $R_\phi(\vartheta) = \exp\{-i(\vartheta/2)[\cos\phi\sigma_x + \sin\phi\sigma_y]\}$. Note that this operation, if implemented perfectly, would return the qubit to the ground state, and hence it can be considered as an operation on the oscillator alone, as defined in Eq. (2). Such an implementation relies on the ability to selectively address number-split qubit transitions, which requires pulses of long duration $\tau \gg 1/\chi$. In practice, it is desirable to keep the pulses short to reduce the probability of ancilla relaxation during the gate. However, shorter pulses of wider bandwidth would drive unintended transitions, as illustrated in Fig. 5(b), leading to imperfect implementation of the SNAP gate: In addition to accumulating incorrect Berry phases for different levels, this will generally leave the qubit and oscillator entangled. Such imperfections are notoriously difficult to calibrate out or precisely account for at the pulse or sequence construction level, which presents a good test bed for our model-free learning paradigm. We demonstrate that our approach leads to high-fidelity protocols even in the case $\tau < 1/\chi$ far from the theoretically optimal regime, where the sequences produced assuming ideal SNAP yield poor fidelity because of severe model bias.

We begin by illustrating in Fig. 5(a) the degradation of performance of the policies optimized for preparation of Fock state $|3\rangle$ using the unitary control circuit from Fig. 2(a) with an ideal SNAP (blue line), when tested with a finite-duration gate SNAP_τ (red and pink lines) whose details are included in the Supplemental Material [69]. Achieving extremely high fidelity (blue line) requires delicate adjustment of the control parameters, but this fine-tuning is futile when the remaining infidelity is smaller than the performance gap due to model bias, shown with arrows in Fig. 2(a) and *a priori* unknown. As seen by testing on the $\chi\tau = 3.4$ case (red line), any progress that the optimizer made after 300 epochs was due to overfitting to the model of the ideal SNAP. As depicted with a spectrum in Fig. 5(b), the qubit pulse of such duration is still reasonably selective (and is close to the experimental choice $\chi\tau \approx 4$ in Ref. [71]), but it already requires a much more sophisticated modeling of the SNAP implementation in order to not limit the experimental performance. In the partially selective case $\chi\tau = 0.4$ (pink line), the performance is drastically worse. Note that sequences optimized with any other simulation-based approach assuming ideal SNAP, such as Refs. [70,74], would exhibit a similar degradation.

One way to recover higher fidelity is through a detailed modeling of the composite qubit pulse in the SNAP [52], although such an approach will still contain residual model bias. An alternative approach, which comes at the expense of reduced success rate, is to perform a verification ancilla measurement and postselection, leading to a control circuit, shown in Fig. 5(c). Postselecting on a qubit measured in $|g\rangle$ in all time steps (history $h_T = 11111$) significantly boosts the fidelity of a biased policy from 0.9 to 0.97 in the case $\chi\tau = 3.4$, but it does not lead to any improvement in the extreme case $\chi\tau = 0.4$. The postselected fidelity is still lower than with the ideal SNAP because such a scheme only compensates for qubit under- or over-rotation, and not for the incorrect Berry phases. Additionally, the trajectories corresponding to other measurement histories have extremely poor fidelities because only the history $h_T = 11111$ was observed during the optimization with an ideal SNAP.

However, in principle, if the qubit is projected to $|e\rangle$ by the measurement, the desired state evolution can still be recovered using adaptive quantum feedback. Experimental Fock state preparation with quantum feedback was demonstrated in the pioneering work in cavity QED [90]. In our context, a general policy in the adaptive setting is a binary

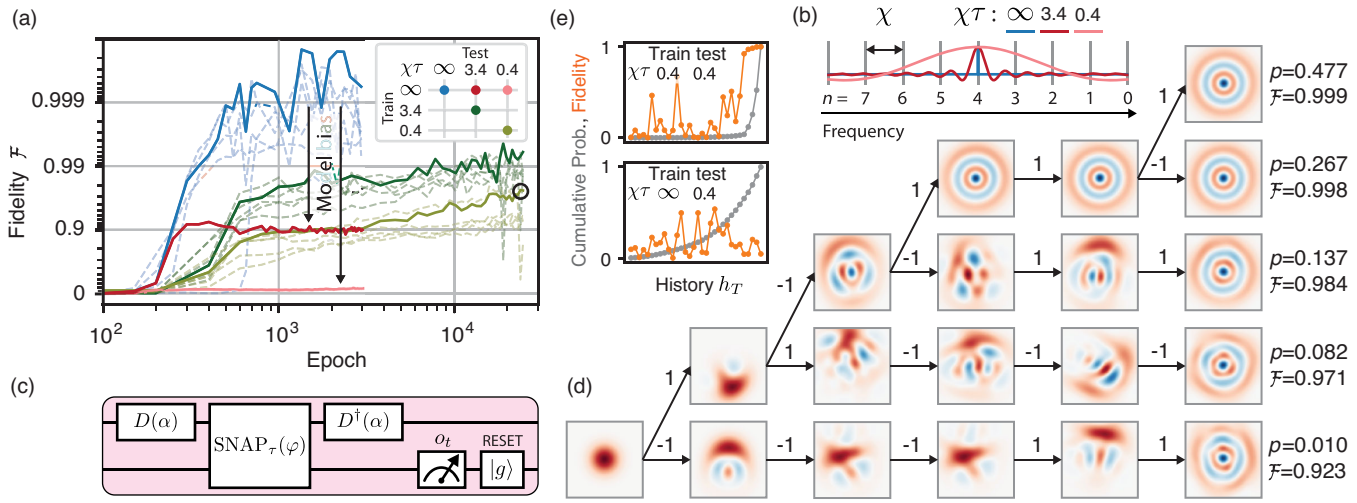

FIG. 5. Learning adaptive measurement-based quantum feedback for preparation of Fock state $|3\rangle$ with imperfect controls. (a) Evaluation of the training progress. Blue lines: training the agent with the unitary control circuit, shown in Fig. 2(a), that uses an ideal SNAP. The background trajectories correspond to six random seeds. The protocols of the best-performing seed are then tested using the same control circuit but with a finite-duration gate SNAP$_\tau$ substituted instead of an ideal SNAP. Such a test reveals the degradation of performance (red and pink lines) due to the model bias. (b) Spectrum of partially selective qubit pulses used in the gate SNAP$_\tau$. The degradation of performance in panel (a) occurs because the pulse overlaps in the frequency domain with unintended number-split qubit transitions, leaving the qubit and oscillator entangled after the gate. (c) Feedback-based control circuit containing a finite-duration gate SNAP$_\tau$ and a verification measurement that produces an observation o_t and disentangles the qubit and oscillator. The qubit is always reset to $|g\rangle$ after the measurement. This control circuit requires either postselection or adaptive control. The agent successfully learns measurement-based feedback control (a, green) even in the extreme case $\chi\tau = 0.4$ far from the theoretically optimal regime $\chi\tau \gg 1$. (d) Example state evolution under the policy obtained after 25,000 epochs of training, shown with a black circle in panel (a). The agent chooses to focus on a small number of branches and to ensure that they lead to high-fidelity states. (e) Cumulative probability and fidelity of the observed histories quantifying this trend (top panel). The policy trained with ideal SNAP and tested with SNAP$_\tau$ (bottom panel) has relatively uniform probability of all histories and poor fidelity.

decision tree, equivalent to 2^{T-1} distinct parameter settings for every possible measurement history. There exist model-based methods for construction of such a tree [91], but they are not applicable in the cases dominated by *a priori* unknown control errors. A RL agent, on the other hand, can discover such a tree in a model-free way. Even though our policies are represented with neural networks, they can be easily converted to a decision-tree representation, which is more advantageous for low-latency inference in real-world experimental implementation.

To this end, we train a new agent with a feedback-based control circuit that directly incorporates a finite-duration imperfect gate SNAP$_\tau$, shown in Fig. 5(c), mimicking training in an experiment. We use a Fock reward circuit, shown in Fig. 2(a), in which $m_1 = 1$ in all episodes, despite the imperfect SNAP, because of the qubit reset operation. Since the control circuit contains a measurement, the agent will be able to dynamically adapt its actions during the episode depending on the received outcomes o_t. As shown with the green curves in Fig. 5(a), the agent successfully learns adaptive strategies of high fidelity even in the extreme case $\chi\tau = 0.4$. This indicates that RL is not only good for fine-tuning or "last-mile" optimization, but it is also a valuable tool for the domains where model-based quantum control is not applicable, e.g., because of the absence of reliable models or prohibitive memory requirements for simulation of a large Hilbert space.

To further analyze the agent's strategy, we select the best-performing random seed for the case $\chi\tau = 0.4$ after 25,000 epochs of training and visualize the resulting state evolution in Fig. 5(d). The average fidelity of such a policy is $\mathcal{F} = 0.974$. There are five high-probability branches, all of which yield $\mathcal{F} > 0.9$, and further postselection of history $h_T = 1\bar{1}111$ will boost the fidelity to $\mathcal{F} > 0.999$. We observe that fidelity reduces in the branches with more "−1" measurement outcomes (top to bottom) because, being less probable, such branches receive less attention from the agent during the training. As shown in Fig. 5(e), top panel, the agent chooses to focus only on a small number of branches (5 out of 2^5) and ensure that they lead to high-fidelity states. This is in contrast to the protocol optimized with the ideal SNAP and tested with SNAP$_\tau$ (bottom panel), which, as a result of model bias, performs poorly and has relatively uniform probability of all histories (of course, such protocol would produce only history 11111 if it was applied with an ideal SNAP).

It is noteworthy that in the two most probable branches in Fig. 5(e), the agent actually finishes preparing the state in just three steps and, in the remaining time, chooses to simply idle instead of further entangling the qubit with the

oscillator and subjecting itself to additional measurement uncertainty. In the other branches, this extra time is used to catch up after previously receiving undesired measurement outcomes. This indeed seems to be an intelligent strategy for such a problem, which serves as a positive indication that this agent will be able to cope with incoherent errors by shortening the effective sequence length.

We emphasize that even though for this numerical demonstration of model-free learning we had to build a specific model of the finite-duration SNAP, the agent is completely agnostic to it by construction. The only input that the agent receives is binary measurement outcomes, whose source is a black box to the agent. Effectively, in this demonstration, the model bias comes from the mismatch between ideal and finite-duration SNAP. We also tested the agent against other types of model bias: We added independent random static offsets to the Berry phases and qubit rotation angles, and found that the agent performs equally well in this situation.

V. DISCUSSION

As empirically demonstrated in Sec. IV, our stochastic policy optimization is stable and leads to high sample efficiency. Starting from a random initial policy, learning the preparation of high-fidelity Fock states (with the target projector reward) and GKP states (with the stabilizer reward) required 10^6–10^7 experimental runs, and learning with the Wigner reward required 10^7–10^8 runs. Although seemingly large, this sample size compares favorably with the number of measurements required to merely tomographically verify the states of similar quality in experiments, e.g., 3×10^6 for Fock states [50] and 2×10^7 for GKP states [86].

Exactly quantifying the sample complexity of heuristic learning algorithms remains difficult. However, we can qualitatively establish the general trends. A natural question to ask is whether our approach will scale favorably with increased (i) target state complexity, (ii) action space, and (iii) sequence length.

(i) Target state complexity: Sample efficiency of learning the control policy is affected by multiple interacting factors, but among the most important is the variance of the fidelity estimator used for the reward assignment. Variance of the estimator in Eq. (5) with $P(\alpha) \propto |W_{\text{target}}(\alpha)|$ is given by $\text{Var} = 4(1+\delta_{\text{target}})^2 - \mathcal{F}^2$, where $\delta_{\text{target}} = \int |W_{\text{target}}(\alpha)| d\alpha - 1$ is one measure of the state nonclassicality known as the Wigner negativity [92] (see Appendix C for the derivation). This result leads to a simple lower bound on the sample complexity of learning the state preparation policy that reaches the fidelity $\mathcal{F}$ to the desired target state

$$M > \frac{4(1+\delta_{\text{target}})^2 - \mathcal{F}^2}{(1-\mathcal{F})^2}. \tag{7}$$

This expression bounds the number of measurements M required for resolving the fidelity $\mathcal{F}$ of a fixed policy with standard error of the mean comparable to the infidelity. The task of the RL agent is more complicated since it needs to not only resolve the fidelity of the current policy but, at the same time, learn how to improve it. Therefore, this bound is not tight, and the practical overhead depends on the choice of control parametrization, the learning algorithm, and its hyperparameters. However, the bound (7) clearly indicates that learning the preparation of larger nonclassical states is increasingly difficult, as one would expect, and the difficulty can be quantified according to the Wigner negativity of the state. This is a fundamental limitation on the learning efficiency with the Wigner reward, which can only be overcome by designing a reward scheme that takes advantage of the special structure of the target state and available trustworthy state manipulation tools, as we did, for instance, for Fock states and GKP states. The Wigner negativity of Fock states grows as $\sqrt{n}$ [92], where n is the photon number, which would result in $O(n)$ scaling of the bound (7). In contrast, the target projector reward, of which the Fock reward is a special case, has target-state-independent variance $\text{Var} = \mathcal{F}(1-\mathcal{F})$ leading to a bound $M > \mathcal{F}/(1-\mathcal{F})$, which does not increase with the photon number. How such a reward design can be optimized in general, is a matter that we leave for further investigation.

(ii) Action space: The overhead on top of Eq. (7) is determined, among other factors, by the choice of the control circuit. In the case of SNAP and displacement, the action-space dimension $|\mathcal{A}| = \Phi + 2$ has to grow with the target state size to ensure individual control of the phases of involved oscillator levels. This might be problematic since the performance of RL (or any other approach) usually declines on high-dimensional tasks, as evidenced, for instance, by studies of robotic locomotion with different numbers of controllable joints [93,94]. However, the sample complexity is not a simple function of $|\mathcal{A}|$, as can be inferred from Fig. 2(b), where we use the same $|\mathcal{A}| = 17$ for all Fock states. For lower Fock states, the agent quickly learns to disregard the irrelevant action dimensions because their contribution to policy gradient averages to zero. In contrast, for higher Fock states, it needs to discover the pattern of relations between all action dimensions across different time steps, and thus the learning is slower. Note that on the same problem, a much stronger degradation is observed when using the Nelder-Mead approach or simulated annealing [see Fig. 2(c)].

(iii) Sequence length: Tackling decision-making problems with long-term dependencies (i.e., $T \gg 1$) is what made RL popular in the first place,

as exemplified by various game-playing agents [14–17]. In quantum control, the temporal structure of the control sequences can be exploited by adopting recurrent neural network architectures, such as the LSTM used in our work. Recently, machine learning for sequential data has significantly advanced with the invention of the transformer models [95], which use attention mechanisms to ensure that the gradients do not decay with the sequence depth T. Machine-learning innovations such as this will undoubtedly find applications in quantum control.

As can be seen above, there are some aspects of scalability that are not specific to quantum control but are common in any control task. The generality of the model-free reinforcement learning framework makes it possible to transfer the solutions to such challenges, found in other domains, to quantum control problems.

Let us now return to the discussion of other factors influencing the sample efficiency. As we briefly alluded to previously, the overhead on top of Eq. (7) depends on the learning algorithm and its hyperparameters. Model-free RL is known to be less sample efficient than gradient-based methods, typically requiring millions of training episodes [13]. This is especially true for on-policy RL algorithms, such as PPO, since they discard the training data after each policy update. In contrast, off-policy methods keep old experiences in the replay buffer and learn from them even after the current policy has long diverged from the old policy under which the data were collected, typically resulting in better sample efficiency. Our pick of PPO was motivated by its simplicity and stability in the stochastic setting, but it is worth exploring an actively expanding collection of RL algorithms [13] and understanding which are most suitable for quantum-observable environments.

The sample efficiency of model-free RL in the quantum control setting can be further improved by utilizing the strength of conventional simulation-based methods. A straightforward way to achieve this would be through supervised pretraining of the agent's policy in the simulation. Such pretraining would provide a better initial point for the agent subsequently retrained in the real-world setting. Our preliminary numerical experiments show that this indeed provides significant speedups.

The proposals discussed above resolve the bias-variance trade-off in favor of complete bias elimination, necessarily sacrificing sample efficiency. In this respect, model-free learning is a swing in the opposite direction from the traditional approach in physics of constructing sparse physically interpretable models with very few parameters which can be calibrated in experiment. Building on the insights from the machine-learning community, model bias can, in principle, be strongly reduced (not eliminated) by learning a richly parametrized model, either physically motivated [96,97] or neural-network based [98,99], from direct interaction with a quantum system. The learned model can then be used to optimize the control policy with simulation-based (not necessarily RL) methods. Another promising alternative is to use model-based reinforcement learning techniques [100], where the agent can plan the actions by virtually interacting with its learned model of the environment while refining both the model and the policy using real-world interactions. Finally, in addition to adopting existing RL algorithms, a worthwhile direction is to design new algorithms tailored to the specifics of quantum-observable environments.

VI. CONCLUSION

Addressing the problem of model bias as an inherent limitation of the dominant simulation-based approach to quantum control, we claim that end-to-end model-free reinforcement learning is not only a feasible alternative, but it is also a powerful tool that will extend the capabilities of quantum control to domains where simulation-based methods are not applicable. By focusing on control of a harmonic oscillator in the circuit QED architecture, we explored various aspects of learning under the conditions of quantum uncertainty and scarce observability. Our policy exploration strategy is explicitly tailored to these features of the quantum learning environments. We demonstrated stable learning directly from stochastic binary measurement outcomes, instead of relying on averaging to eliminate stochasticity as is done in other model-free quantum control optimization methods. With multiple numerical experiments, we confirmed that such a strategy leads to high fidelity and sample efficiency on challenging control tasks that include both the unitary control and control with adaptive measurement-based quantum feedback. The RL agent that we developed can be directly applied in real-world experiments with various physical systems.

ACKNOWLEDGMENTS

We acknowledge a helpful discussion with Thomas Fösel. We thank the anonymous reviewers for their comments, which encouraged us to make several additions. We thank Yale Center for Research Computing for providing compute resources. This research is supported by ARO under Grant No. W911NF-18-1-0212. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office (ARO), or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

APPENDIX A: EDUCATIONAL EXAMPLE

In this Appendix, we analyze a deliberately simple problem with the purpose of illustrating in detail various components and stages of the learning process.

Problem setting.—Consider a qubit state preparation problem in which the initial state is $|g\rangle$ and the target state is $|e\rangle$. Such state preparation can be achieved with a unitary rotation gate parametrized as $U(a) = \exp(-i\pi a\sigma_x)$, where the optimal solution $a = 0.5$ is known in advance. We let the agent discover this solution in a model-free way, without knowing which unitary is actually applied. The training episodes consist of a single time step in which the agent produces an action $a \in \mathbb{R}$, leading to execution of control circuit $U(a)$; the agent then collects a reward with a simple reward circuit consisting of a σ_z measurement, as shown in the inset of Fig. 6(a). The resulting measurement outcome $m \in \{-1, 1\}$ is used to issue a reward $R = -m$, which is maximized in the target state $|e\rangle$, hence satisfying Eq. (1).

Actor and critic.—In every training episode, the action a is sampled according to the probability distribution specified by the policy. Policy $\pi_\theta(a)$ is parametrized with learnable parameters θ. In this problem, it is convenient to choose a simple Gaussian policy

$$\pi_\theta(a) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(a-\mu)^2}{2\sigma^2}\right], \tag{A1}$$

whose learnable parameters are $\theta = \{\mu, \sigma^2\}$. The policy defines how the agent interacts with the environment, and it is often referred to as the actor. Another important component of PPO is the value function $V_{\theta'}$, or critic, which helps the agent assess the value of the environment state (see Supplemental Material [69]). In this example, the value function can be chosen as a simple baseline $V_{\theta'} = b$ with learnable parameters $\theta' = \{b\}$. During the training process, parameters $\{\mu, \sigma^2, b\}$ are iteratively updated according to the PPO algorithm.

Training process.—The training process, illustrated in Fig. 6, is split into 50 epochs. Within each epoch k, the parameters of the policy remain fixed, and the agent collects a batch of $B = 30$ episodes of experience, behaving stochastically according to the current policy $\pi_{\theta_k}(a)$. Figure 6(a) shows the policy distribution for a selected set of epochs and the actions that the agent tried in the episodes of the corresponding epoch. The initial policy is widely distributed to ensure that the agent can adequately explore the action space. Since initially most of the actions do not lead to high-fidelity states, the agent is very likely to receive negative rewards, as shown in Fig. 6(b). After every epoch, the parameters of the stochastic policy (A1) are updated $\theta_k \to \theta_{k+1}$ in a way that utilizes the information contained in the reward signal. Controlled by the learning rate, these updates result in gradually shifting the probability density of the stochastic policy towards more promising actions, as seen in Fig. 6(a). After iterating in this manner for several epochs, the policy becomes localized near the correct value of the action, which leads to a significantly increased fraction of positive rewards. In the initial stage, the best progress is achieved by rapidly learning the parameter μ. However, to achieve high fidelity, it is necessary to localize μ more finely, and thus, in the later stages, the agent shrinks the variance σ^2 of the policy. Eventually, there are almost no episodes with a negative reward, meaning that the agent has achieved good performance.

Complications.—This simple example illustrates how learning proceeds in our approach. More realistic examples contained in Sec. IV follow the same basic principles. Additional complications arise from the following considerations.

(i) Typically, the action space $\mathcal{A}$ is high dimensional. In such a case, the Gaussian policy distribution is defined on $\mathbb{R}^{|\mathcal{A}|}$ instead of $\mathbb{R}$.

(ii) The agent can receive a nontrivial observation o, for instance, a qubit measurement outcome, which requires incorporating adaptive measurement-based feedback into the policy. In such a case, the policy distribution $\pi_\theta(a|o)$ is conditioned on the observation. In the case of a Gaussian policy, this is achieved by making the mean and variance para-

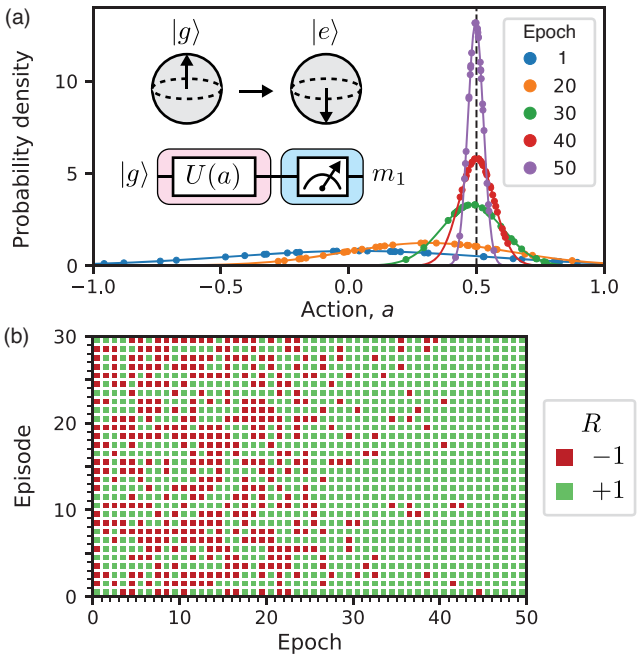

FIG. 6. Educational example of model-free learning. (a) Inset: The task is to prepare qubit state $|e\rangle$ starting from state $|g\rangle$. Episodes consist of a single time step; the control circuit contains a rotation unitary $U(a)$, and the reward circuit contains a measurement of σ_z. Main panel: policy distribution (solid lines) for a selected set of epochs, and actions that the agent tried in the episodes of corresponding epoch (dots). (b) Rewards received by the agent in the episodes of every epoch.

metrized functions of the observation $\{\mu, \sigma^2\} = \{\mu_\theta(o), \sigma_\theta^2(o)\}$. In our work, these functions are chosen to be neural networks.

(iii) The episodes typically consist of multiple time steps. In such cases, the policy distribution $\pi_\theta(a|t; h_t)$ is conditioned on the time-step index t and on the history of observations $h_t = o_{0:t}$ received up to the current time step. For notational simplicity, we usually treat the time dependence as implicit and denote the policy as $\pi_\theta(a|h_t)$.

APPENDIX B: ALTERNATIVE MODEL-FREE APPROACHES

1. Qualitative comparison of action-space exploration strategies

It is instructive to compare the action-space exploration strategy of our RL agent to widely used model-free methods. For this comparison, we focus on the NM simplex search used in many quantum control experiments [6–8].

NM and other model-free methods that view quantum control as a standard cost function optimization problem explore the action space by evaluating the cost function for a set of policy candidates and by using this evaluation to inform the selection of the next candidate. In NM, the latter step is done by choosing a new vertex of the simplex, as illustrated in Fig. 7(a). The effectiveness of such an approach relies on the ability to reliably approximate the cost function landscape by only sampling it at a small subset of points. In general, this is difficult to achieve in high-dimensional action spaces or when the cost function is stochastic. Therefore, such an approach requires spending a large part of the sample budget on averaging, which limits the number of policies that it can explore under the constraint of a fixed total sample size of M_{tot} experimental runs.

On the other hand, in our RL approach, every experimental run (episode) is performed with a slightly different policy. These random policy candidates are assigned a stochastic score of ± 1, resulting from the reward measurement outcome. Even though the value of the "cost function" is not known to any satisfying accuracy for any of the policy candidates, the acquired information is sufficient to stochastically move the Gaussian distribution of policy candidates towards a more promising region of the action space, as illustrated in Fig. 7(b). In contrast to NM that crucially relies on averaging, our RL agent spends the sample budget to effectively explore a much larger part of the action space.

To confirm this intuition, we quantitatively compare the RL agent to widely used model-free approaches, the NM simplex search and SA, on the task of Fock state preparation when constrained to the same total sample size of $M_{\text{tot}} = 4 \times 10^6$. The results of this comparison are shown in Fig. 2(c), revealing that RL indeed significantly outperforms its model-free alternatives in terms of sample efficiency, especially when the effective problem dimension increases, i.e., for higher photon numbers n. In the following sections, we describe the numerical experiments with NM and SA, performed using their SciPy 1.4.1 implementation [101].

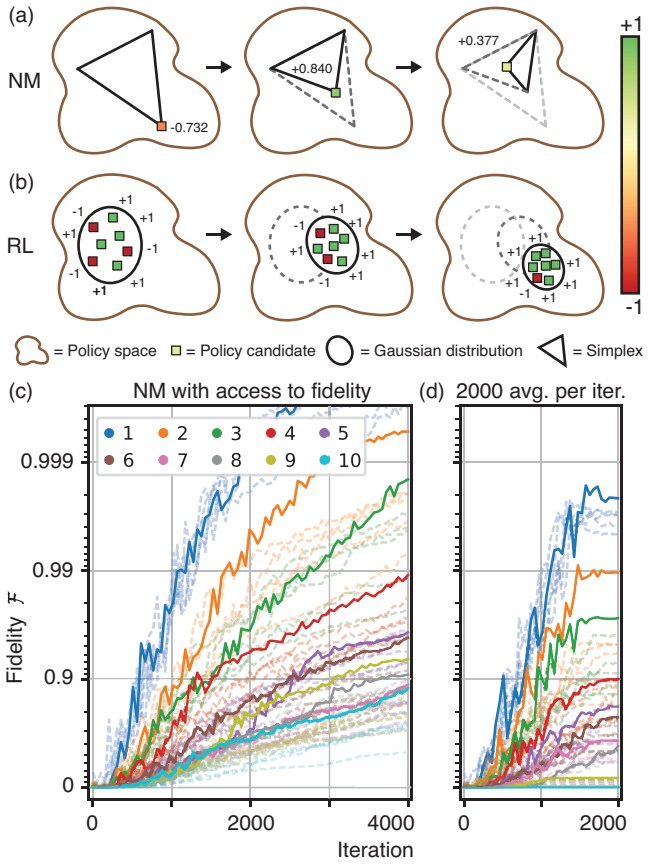

FIG. 7. Preparation of Fock states $|1\rangle, \ldots, |10\rangle$ with the NM simplex search. (a) Cartoon depiction of the NM simplex search. One algorithm iteration corresponds to an update of one vertex of a simplex. The color-coded values of the cost function have high resolution, achieved through averaging of many ± 1 measurement outcomes. (b) Cartoon depiction of our RL approach. Every training epoch consists of several episodes executed with different policy candidates that are sampled from a Gaussian distribution. Policy candidates are assigned a low-resolution reward of ± 1 based on a single measurement outcome instead of averaging. (c) NM optimization progress with infidelity used as a cost function. The background trajectories correspond to six random seeds for each state, and solid lines show the trajectory with the highest final fidelity. (d) NM optimization progress with a stochastic cost function obtained by averaging 2000 outcomes of the Fock reward circuit shown in Fig. 2(a).

2. Nelder-Mead simplex search

To ensure a fair comparison of NM with RL, we perform hyperparameter tuning for NM and display the best of the six independent optimization runs for each problem setting. Given the simplicity of the NM heuristic with its small

number of hyperparameters, we believe that the performed tuning is exhaustive and that no further significant improvements are possible.

First, we study the performance of NM when it is given direct access to fidelity on the task of Fock state preparation. We initialize the control circuits with random parameters whose magnitude is swept to optimize the NM performance, as it is known to be sensitive to the simplex initialization. We find that the optimal initialization is similar to that in RL and corresponds to random initial circuits that do not significantly deviate the oscillator state from vacuum. With this choice, the convergence of NM is shown in Fig. 7(c). It exhibits fast degradation with increasing photon number n. Next, we study the performance of NM in the presence of measurement sampling noise. We constrain NM to the same total sample size $M_{\mathrm{tot}} = 4 \times 10^6$ as used for RL, and we optimally split the sample budget between algorithm iterations and averages per iteration to maximize the final performance. The convergence of NM with 2000 averages per iteration is shown in Fig. 7(d), and it can be directly compared to the RL results in Fig. 2(b), clearly showing the advantage of RL in the stochastic setting.

3. Simulated annealing

We use simulated annealing with the Cauchy-Lorentz visiting distribution and without a local search on accepted locations, which is a similar version to the recent experiment [40]. We performed extensive tuning of hyperparameters, including the magnitude of the randomly initialized control circuit parameters, parameters of the visiting distribution, as well as initial and final temperatures. The optimization results with the best choice of hyperparameters are shown in Fig. 8, where for each optimization trajectory, we only display the best fidelity of every 100 consecutive iterations to reduce the plot clutter resulting from the periodic restarts of the annealing.

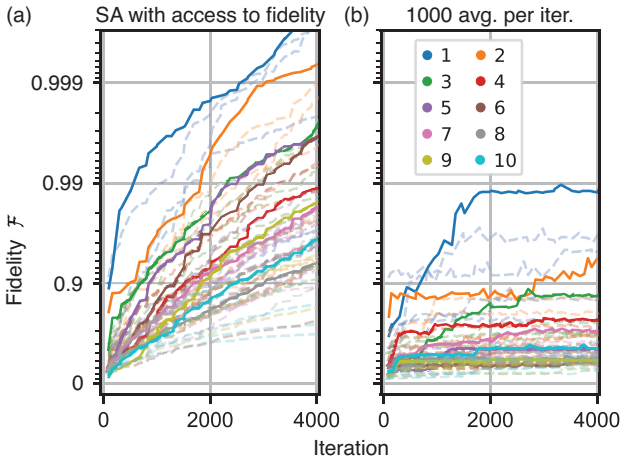

FIG. 8. Preparation of Fock states $|1\rangle, \ldots, |10\rangle$ with simulated annealing. (a) SA optimization progress with infidelity used as a cost function. The background trajectories correspond to six random seeds for each state, and solid lines show the trajectory with the highest final fidelity. (b) SA optimization progress with a stochastic cost function obtained by averaging 1000 outcomes of the Fock reward circuit shown in Fig. 2(a).

With direct access to fidelity, as shown in Fig. 8(a), the convergence of SA is similar to NM, and is significantly slower than the RL agent even when the agent does not have access to fidelity. Next, we replace the fidelity with its estimator based on 1000 runs of the Fock reward circuit. This number of runs per cost function evaluation is tuned to achieve the highest performance under the constrained total sample size of $M_{\mathrm{tot}} = 4 \times 10^6$. In such stochastic settings, the performance of SA drops significantly, as shown in Fig. 8(b), and is worse than that of both NM and RL.

APPENDIX C: VARIANCE OF THE FIDELITY ESTIMATOR

Variance of the estimator (5) is given by

$$\mathrm{Var} = \mathop{\mathbb{E}}_{\alpha\sim P}\mathop{\mathbb{E}}_{\psi}\left[\left(\frac{2}{P(\alpha)}\Pi_\alpha W_{\mathrm{target}}(\alpha)\right)^2\right] - \left(\mathop{\mathbb{E}}_{\alpha\sim P}\mathop{\mathbb{E}}_{\psi}\left[\frac{2}{P(\alpha)}\Pi_\alpha W_{\mathrm{target}}(\alpha)\right]\right)^2 \tag{C1}$$

$$= \int \frac{4}{P(\alpha)} W^2_{\mathrm{target}}(\alpha) d\alpha - \mathcal{F}^2, \tag{C2}$$

where we made the simplifications $\Pi_\alpha^2 = 1$ and $\mathop{\mathbb{E}}_{\alpha\sim P}[\ldots] = \int [\ldots] P(\alpha) d\alpha$.

We now use variational calculus to find the $P(\alpha)$ that minimizes Eq. (C2) with the constraint $\int P(\alpha) d\alpha = 1$. The variational derivative is given by

$$\delta(\mathrm{Var}) = \int \left[c - \frac{4}{P^2(\alpha)} W^2_{\mathrm{target}}(\alpha)\right] \delta P(\alpha) d\alpha, \tag{C3}$$

where c is the Lagrange multiplier for the constraint. From this, we find that the optimal sampling distribution satisfies $P(\alpha) \propto |W_{\mathrm{target}}(\alpha)|$, and the minimal variance is

$$\min\{\mathrm{Var}\} = 4\left(\int |W_{\mathrm{target}}(\alpha)| d\alpha\right)^2 - \mathcal{F}^2. \tag{C4}$$

We consider the sampling problem in which $N_m = 1$ parity measurement is done per phase-space point, and in such setting, we find an optimal sampling distribution independent of the state that is being characterized—a rather convenient property for the online training since the actual prepared state is not known (only the target state is known). We can consider a different problem, in which both $W(\alpha)$ and $W_{\mathrm{target}}(\alpha)$ are known, and where the goal is to compute the fidelity integral (4) through Monte Carlo phase-space sampling. This can be relevant, for instance, in a simulation, as an

alternative to computing the integral through the Riemann sum. In such setting, the optimal condition for the variance is modified to $P(\alpha) \propto |W(\alpha)W_{\text{target}}(\alpha)|$. If, in addition, the fidelity is known in advance to be close to 1, i.e., $W(\alpha) \approx W_{\text{target}}(\alpha)$, then the optimal sampling distribution becomes $P(\alpha) \propto W^2_{\text{target}}(\alpha)$. The latter does not depend on the state that is being characterized, and therefore, it can also be used in the online setting, as was proposed in Refs. [79,80]. However, such a sampling distribution is optimal only in the limit $N_m \gg 1$.

In general, consider fidelity estimation based on N_α phase-space points and N_m parity measurements per point, such that the total number of measurements $N = N_\alpha N_m$ is fixed. Under this condition, the optimal choice is $N_\alpha = N$, $N_m = 1$ (adopted in this work), in which case the distribution $P(\alpha) \propto |W_{\text{target}}(\alpha)|$ is optimal. However, because of various hardware constraints (e.g., small memory of the FPGA controller), in some experiments, it might be preferred to limit $N_\alpha = C$ and compensate for it by accumulating multiple measurements in each phase-space point, i.e., $N_m = N/C \gg 1$. Under such constraints, the optimal sampling corresponds to $P(\alpha) \propto W^2_{\text{target}}(\alpha)$.

APPENDIX D: OTHER REWARD MEASUREMENT SCHEMES

In this Appendix, we describe how our approach can be adapted to the control of other physical systems, focusing specifically on the design of probabilistic reward measurement schemes.

1. State preparation in trapped ions

Universal control of a motional state of a trapped ion can be achieved by utilizing the ion's internal electronic levels as ancilla qubit [42,43]. Control policies are typically produced with GRAPE, but modular constructions also exist [102]. Regardless of the control circuit parametrization, our RL approach can be used for model-free learning of its parameters. Here, we propose a reward circuit that can be used for such learning in trapped ions, based on the characteristic function.

The symmetric characteristic function of a continuous-variable system is defined as $C(\alpha) = \langle D(\alpha) \rangle$ [103]. It is equal to the 2D Fourier transform of the Wigner function, and it is therefore tomographically complete and can be used to construct the fidelity estimator similar to Eq. (5):

$$\mathcal{F} = \frac{1}{\pi} \int d^2\alpha C(\alpha) C^*_{\text{target}}(\alpha) \tag{D1}$$

$$= \frac{1}{\pi} \mathop{\mathbb{E}}_{\alpha\sim P} \mathop{\mathbb{E}}_{\psi} \left[\frac{1}{P(\alpha)} D(\alpha) C^*_{\text{target}}(\alpha) \right], \tag{D2}$$

where $P(\alpha)$ is the phase-space sampling distribution. In trapped ions, the characteristic function can be measured with phase estimation of the unitary displacement operator [53,85].

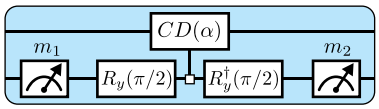

FIG. 9. Reward circuit for learning preparation of arbitrary symmetric states of a continuous-variable system, based on the characteristic function.

For simplicity, we focus on symmetric states whose characteristic function is real (e.g., Fock states and GKP states), although the procedure can be generalized to asymmetric states. In this case, the reward circuit is similar to the Wigner reward, and it is shown in Fig. 9. The conditional displacement gate $CD(\alpha)$, required for such a reward circuit, is typically called the "internal-state-dependent force" in the trapped ions community. Note that it was also recently realized in circuit QED [51,86].

2. Multiqubit systems

Universal control of a system of n qubits with Hilbert space of dimension $d = 2^n$ can be achieved with various choices of control circuits that can be tailored to the specific physical layout of the device. We refer to the literature on variational quantum algorithms for more details [104]. Here, we focus instead on the reward measurement schemes. There exists a large body of work on quantum state certification in the multiqubit systems [75]. Our RL approach greatly benefits from this work since state certification protocols can be directly converted into probabilistic reward measurement schemes for state preparation control problems. Moreover, some state certification protocols are directly linked to fidelity estimation, which allows us to construct reward measurement schemes satisfying the condition $\mathbb{E}[R] = f(\mathcal{F})$, where f is a monotonously increasing function of fidelity. Here, we propose a stabilizer reward built on the stabilizer state certification protocol [75] and a reward for preparation of arbitrary n-qubit states based on the characteristic function.

a. Stabilizer states

Consider a stabilizer group $\mathcal{S} = \{I, S_1, \ldots, S_{d-1}\}$ and a corresponding parametrized set of POVM elements $\{\Omega_k\}$, which consists of projectors $\Omega_k = \frac{1}{2}(I + S_k)$ onto the $+1$ eigenspace of each stabilizer, except for the trivial stabilizer I. We sample the parameter $k = 1, \ldots, d-1$ uniformly with probabilities $P(k) = 1/(d-1)$ and with the associated identical reward scale $R_k = 1$. The reward of ± 1 is issued based on the stabilizer measurement outcome. A straightforward calculation shows that, in this case, the expectation of reward satisfies $\mathbb{E}[R] = (2^n\mathcal{F} - 1)/(2^n - 1)$, and therefore, it also automatically satisfies the condition (1). Note the difference from the GKP state preparation example

considered in Sec. IV B, where the stabilizer group was infinite and we considered sampling of only the generators of this group, which does not lead to a simple connection between $\mathbb{E}[R]$ and $\mathcal{F}$.

b. Arbitrary states

The stabilizer reward is only applicable to a restricted family of states. To construct a reward measurement scheme applicable to arbitrary states, we need to choose a tomographically complete set of POVM elements. The simplest such scheme is based on the Pauli group, where the fidelity estimator can be constructed based on the measurements of d^2 possible n-fold tensor products G_k of single-qubit Pauli operators [79]. Instead of sampling points α in the continuous phase space, in this case, we sample indices k of the Pauli operators from a discrete set $\{k = 1, \ldots, d^2\}$ with probability distribution $P(k)$. Denoting the characteristic function as $C(k) = \langle G_k \rangle$, we obtain an estimator

$$\mathcal{F} = \frac{1}{d}\sum_k C(k) C_{\text{target}}(k) \tag{D3}$$

$$= \frac{1}{d}\,\mathop{\mathbb{E}}_{k\sim P}\,\mathop{\mathbb{E}}_{\psi}\left[\frac{1}{P(k)} G_k C_{\text{target}}(k)\right]. \tag{D4}$$

Given the estimator above, the reward circuit simply consists of measurement of the sampled Pauli operator.

APPENDIX E: LEARNING GATES FOR ENCODED QUBITS

The tools demonstrated for quantum state preparation in Sec. IV are applicable for learning more general quantum operations that map an input subspace of the state space to the target output subspace. For example, consider a qubit encoded in oscillator states $\{|\pm Z_L\rangle\}$, which serve as logical Z eigenstates. Learning a gate U_{target} on this logical qubit amounts to finding an operation that simultaneously implements the state transfers $|\pm Z_L\rangle \to U_{\text{target}}|\pm Z_L\rangle$ and that extends to logical qubit subspace by linearity. However, the reward circuits introduced in Sec. IV will result in a final state equal to the target up to an arbitrary phase factor; hence, it is insufficient to only use the set $\{|\pm Z_L\rangle\}$ during the training. To constrain the phase factor, we extend this set to include all cardinal points $\{|\pm X_L\rangle, |\pm Y_L\rangle, |\pm Z_L\rangle\}$ on the logical Bloch sphere.

The training process for a gate is a straightforward generalization of the training for state preparation depicted in Fig. 1, as summarized below:

(1) Sample initial state $|\psi_0\rangle \in \{|\pm X_L\rangle, |\pm Y_L\rangle, |\pm Z_L\rangle\}$. Start the episode by preparing this state.
(2) Run the episode by applying T steps of the control circuit, resulting in a state $|\psi_T\rangle$.
(3) Apply a reward circuit to state $|\psi_T\rangle$, with the target state given by $|\psi_{\text{target}}\rangle = U_{\text{target}}|\psi_0\rangle$.

Here, we demonstrate learning of logical gates for the Fock encoding with $|+Z\rangle = |0\rangle$ and $|-Z\rangle = |1\rangle$, and for the GKP encoding with $\Delta = 0.3$. In these numerical experiments, we sample a new initial state every epoch, and we use the same state for all batch members within the epoch (preparation of the initial states can be learned beforehand). We use an ideal SNAP-displacement control circuit, as shown in Fig. 2(a), and a Wigner reward circuit, as shown in Fig. 4(a), with a single phase-space point and a single measurement per policy candidate. The choice of training hyperparameters is summarized in the Supplemental Material [69].

The training results are displayed in Fig. 10 for the Hadamard H and Pauli X gates on the Fock qubit, and a non-Clifford $\sqrt{H}$ gate on the GKP qubit. We use average gate fidelity [105] as an evaluation metric. These results show that stable convergence is achieved in such QOMDP despite an additional source of randomness due to the sampling of initial states. The total number of experimental realizations used by the agent is 10^6, 2×10^6, and 4×10^6 for the H, X, and $\sqrt{H}$ gates, respectively.

In future work, an error amplification technique based on gate repetitions, such as randomized benchmarking, can be incorporated to increase the SNR of the reward, similarly to how it is done in other quantum control demonstrations [6,9]. However, this technique could be modified, in the spirit of our approach, to use a single experimental realization of a randomized benchmarking sequence as one episode, instead of averaging them to suppress the stochasticity of the cost function.

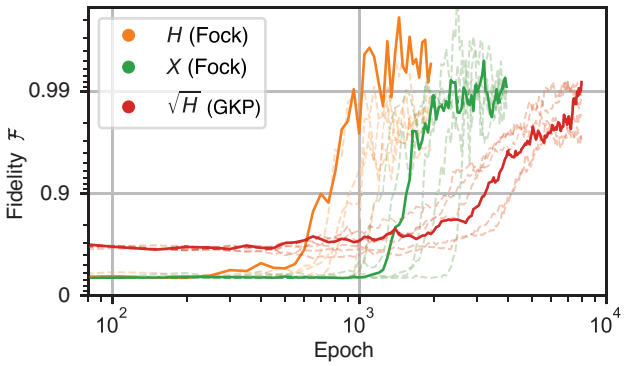

FIG. 10. Learning gates for logical qubits encoded in an oscillator. The agent is trained to produce Hadamard H and Pauli X gates on the Fock qubit, and a non-Clifford $\sqrt{H}$ gate on the GKP qubit. The average gate fidelity is used as an evaluation metric. The background trajectories correspond to six random seeds for each gate, and solid lines show the trajectory with the highest final fidelity.

[1] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *Optimal Control of Coupled Spin Dynamics: Design of NMR Pulse Sequences by Gradient Ascent Algorithms*, J. Magn. Reson. **172**, 296 (2005).

[2] T. Caneva, T. Calarco, and S. Montangero, *Chopped Random-Basis Quantum Optimization*, Phys. Rev. A **84**, 022326 (2011).

[3] P. De Fouquieres, S. G. Schirmer, S. J. Glaser, and I. Kuprov, *Second Order Gradient Ascent Pulse Engineering*, J. Magn. Reson. **212**, 412 (2011).

[4] N. Leung, M. Abdelhafez, J. Koch, and D. Schuster, *Speedup for Quantum Optimal Control from Automatic Differentiation Based on Graphics Processing Units*, Phys. Rev. A **95**, 042318 (2017).

[5] M. Abdelhafez, D. I. Schuster, and J. Koch, *Gradient-Based Optimal Control of Open Quantum Systems Using Quantum Trajectories and Automatic Differentiation*, Phys. Rev. A **99**, 052327 (2019).

[6] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I. C. Hoi, E. Jeffrey *et al.*, *Optimal Quantum Control Using Randomized Benchmarking*, Phys. Rev. Lett. **112**, 240504 (2014).

[7] Z. Chen, J. Kelly, C. Quintana, R. Barends, B. Campbell, Y. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Lucero *et al.*, *Measuring and Suppressing Quantum State Leakage in a Superconducting Qubit*, Phys. Rev. Lett. **116**, 020501 (2016).

[8] M. A. Rol, C. C. Bultink, T. E. O'Brien, S. R. de Jong, L. S. Theis, X. Fu, F. Luthi, R. F. L. Vermeulen, J. C. de Sterke, A. Bruno *et al.*, *Restless Tuneup of High-Fidelity Qubit Gates*, Phys. Rev. Applied **7**, 041001(R) (2017).

[9] M. Werninghaus, D. J. Egger, F. Roy, S. Machnes, F. K. Wilhelm, and S. Filipp, *Leakage Reduction in Fast Superconducting Qubit Gates via Optimal Control*, npj Quantum Inf. **7**, 14 (2021).

[10] R. S. Judson and H. Rabitz, *Teaching Lasers to Control Molecules*, Phys. Rev. Lett. **68**, 1500 (1992).

[11] A. Lumino, E. Polino, A. S. Rab, G. Milani, N. Spagnolo, N. Wiebe, and F. Sciarrino, *Experimental Phase Estimation Enhanced by Machine Learning*, Phys. Rev. Applied **10**, 044033 (2018).

[12] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (A Bradford Book, 2018).

[13] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, *An Introduction to Deep Reinforcement Learning*, Found. Trends Mach. Learn. **11**, 219 (2018).

[14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, *Mastering the Game of Go with Deep Neural Networks and Tree Search*, Nature (London) **529**, 484 (2016).

[15] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, *A General Reinforcement Learning Algorithm that Masters Chess, Shogi, and Go through Self-Play*, Science **362**, 1140 (2018).

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, *Human-Level Control through Deep Reinforcement Learning*, Nature (London) **518**, 529 (2015).

[17] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev *et al.*, *Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning*, Nature (London) **575**, 350 (2019).

[18] S. Levine, C. Finn, T. Darrell, and P. Abbeel, *End-to-End Training of Deep Visuomotor Policies*, J. Mach. Learn. Research **17**, 1 (2015), https://www.jmlr.org/papers/volume17/15-522/15-522.pdf.

[19] T. Haarnoja, S. Ha, A. Zhou, J. Tan, G. Tucker, and S. Levine, *Learning to Walk via Deep Reinforcement Learning*, *Proceedings of Robotics: Science and Systems* (2019), 10.15607/RSS.2019.XV.011.

[20] C. Chen, D. Dong, H. X. Li, J. Chu, and T. J. Tarn, *Fidelity-Based Probabilistic Q-Learning for Control of Quantum Systems*, IEEE Trans. Neural Netw. Learn. Syst. **25**, 920 (2014).

[21] M. Bukov, A. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, *Reinforcement Learning in Different Phases of Quantum Control*, Phys. Rev. X **8**, 031086 (2018).

[22] M. Bukov, *Reinforcement Learning for Autonomous Preparation of Floquet-Engineered States: Inverting the Quantum Kapitza Oscillator*, Phys. Rev. B **98**, 224305 (2018).

[23] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, *When Does Reinforcement Learning Stand out in Quantum Control? A Comparative Study on State Preparation*, npj Quantum Inf. **5**, 85 (2019).

[24] R. Porotti, D. Tamascelli, M. Restelli, and E. Prati, *Coherent Transport of Quantum States by Deep Reinforcement Learning*, Commun. Phys. **2**, 61 (2019).

[25] Z. An, H.-J. Song, Q.-K. He, and D. L. Zhou, *Quantum Optimal Control of Multilevel Dissipative Quantum Systems with Reinforcement Learning*, Phys. Rev. A **103**, 012404 (2021).

[26] M. August and J. M. Hernández-Lobato, in *Lecture Notes in Computer Science* (2018), Vol. 11203, pp. 591–613, 10.1007/978-3-030-02465-9_43.

[27] T. Haug, W.-K. Mok, J.-B. You, W. Zhang, C. E. Png, and L.-C. Kwek, *Classifying Global State Preparation via Deep Reinforcement Learning*, Mach. Learn. **2**, 01LT02 (2021).

[28] E.-J. Kuo, Y.-L. L. Fang, and S. Y.-C. Chen, *Quantum Architecture Search via Deep Reinforcement Learning*, arXiv:2104.07715.

[29] Z. T. Wang, Y. Ashida, and M. Ueda, *Deep Reinforcement Learning Control of Quantum Cartpoles*, Phys. Rev. Lett. **125**, 100401 (2020).

[30] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, *Measurement Based Feedback Quantum Control with Deep Reinforcement Learning*, Phys. Rev. Lett. **127**, 190403 (2021).

[31] M. Dalgaard, F. Motzoi, J. J. Sørensen, and J. Sherson, *Global Optimization of Quantum Dynamics with AlphaZero Deep Exploration*, npj Quantum Inf. **6**, 6 (2020).

[32] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, *Universal Quantum Control through Deep Reinforcement Learning*, npj Quantum Inf. **5**, 33 (2019).

[33] Z. An and D. L. Zhou, *Deep Reinforcement Learning for Quantum Gate Control*, Europhys. Lett. **126**, 60002 (2019).
[34] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, *Reinforcement Learning with Neural Networks for Quantum Feedback*, Phys. Rev. X **8**, 031084 (2018).
[35] P. Andreasson, J. Johansson, S. Liljestrand, and M. Granath, *Quantum Error Correction for the Toric Code Using Deep Reinforcement Learning*, Quantum **3**, 183 (2019).
[36] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, *Optimizing Quantum Error Correction Codes with Reinforcement Learning*, Quantum **3**, 215 (2019).
[37] L. D. Colomer, M. Skotiniotis, and R. Muñoz-Tapia, *Reinforcement Learning for Optimal Error Correction of Toric Codes*, Phys. Lett. A **384**, 126353 (2020).
[38] H. Xu, J. Li, L. Liu, Y. Wang, H. Yuan, and X. Wang, *Generalizable Control for Quantum Parameter Estimation through Reinforcement Learning*, npj Quantum Inf. **5**, 82 (2019).
[39] J. Schuff, L. J. Fiderer, and D. Braun, *Improving the Dynamics of Quantum Sensors with Reinforcement Learning*, New J. Phys. **22**, 035001 (2020).
[40] Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, P. Mundada, T. Merkh, A. R. Carvalho, and M. J. Biercuk, *Experimental Deep Reinforcement Learning for Error-Robust Gate-Set Design on a Superconducting Quantum Computer*, PRX Quantum **2**, 040324 (2021).
[41] J. Barry, D. T. Barry, and S. Aaronson, *Quantum Partially Observable Markov Decision Processes*, Phys. Rev. A **90**, 032311 (2014).
[42] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, *Quantum Dynamics of Single Trapped Ions*, Rev. Mod. Phys. **75**, 281 (2003).
[43] C. D. Bruzewicz, J. Chiaverini, R. McConnell, and J. M. Sage, *Trapped-Ion Quantum Computing: Progress and Challenges*, Appl. Phys. Rev. **6**, 021314 (2019).
[44] P. Krantz, M. Kjaergaard, F. Yan, T. P. Orlando, S. Gustavsson, and W. D. Oliver, *A Quantum Engineer's Guide to Superconducting Qubits*, Appl. Phys. Rev. **6**, 021318 (2019).
[45] A. Blais, A. L. Grimsmo, S. M. Girvin, and A. Wallraff, *Circuit Quantum Electrodynamics*, Rev. Mod. Phys. **93**, 025005 (2021).
[46] N. Ofek, A. Petrenko, R. Heeres, P. Reinhold, Z. Leghtas, B. Vlastakis, Y. Liu, L. Frunzio, S. M. Girvin, L. Jiang *et al.*, *Extending the Lifetime of a Quantum Bit with Error Correction in Superconducting Circuits*, Nature (London) **536**, 441 (2016).
[47] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf *et al.*, *Quantum Error Correction of a Qubit Encoded in Grid States of an Oscillator*, Nature (London) **584**, 368 (2020).
[48] L. Hu, Y. Ma, W. Cai, X. Mu, Y. Xu, W. Wang, Y. Wu, H. Wang, Y. P. Song, C.-L. Zou *et al.*, *Quantum Error Correction and Universal Gate Set Operation on a Binomial Bosonic Logical Qubit*, Nat. Phys. **15**, 503 (2019).
[49] W. Wang, Y. Wu, Y. Ma, W. Cai, L. Hu, X. Mu, Y. Xu, Z.-J. Chen, H. Wang, Y. P. Song *et al.*, *Heisenberg-Limited Single-Mode Quantum Metrology in a Superconducting Circuit*, Nat. Commun. **10**, 4382 (2019).
[50] R. W. Heeres, P. Reinhold, N. Ofek, L. Frunzio, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, *Implementing a Universal Gate Set on a Logical Qubit Encoded in an Oscillator*, Nat. Commun. **8**, 94 (2017).
[51] A. Eickbusch, V. Sivak, A. Z. Ding, S. S. Elder, S. R. Jha, J. Venkatraman, B. Royer, S. M. Girvin, R. J. Schoelkopf, and M. H. Devoret, *Fast Universal Control of an Oscillator with Weak Dispersive Coupling to a Qubit*, arXiv:2111.06414.
[52] M. Kudra, M. Kervinen, I. Strandberg, S. Ahmed, M. Scigliuzzo, A. Osman, D. P. Lozano, G. Ferrini, J. Bylander, A. F. Kockum *et al.*, *Robust Preparation of Wigner-Negative States with Optimized SNAP-Displacement Sequences*, arXiv:2111.07965.
[53] C. Flühmann and J. P. Home, *Direct Characteristic-Function Tomography of Quantum States of the Trapped-Ion Motional Oscillator*, Phys. Rev. Lett. **125**, 043602 (2020).
[54] D. Gottesman, A. Kitaev, and J. Preskill, *Encoding a Qubit in an Oscillator*, Phys. Rev. A **64**, 012310 (2001).
[55] M. H. Michael, M. Silveri, R. T. Brierley, V. V. Albert, J. Salmilehto, L. Jiang, and S. M. Girvin, *New Class of Quantum Error-Correcting Codes for a Bosonic Mode*, Phys. Rev. X **6**, 031006 (2016).
[56] See https://github.com/v-sivak/quantum-control-rl.
[57] A. Garcia-Saez and J. Riu, *Quantum Observables for continuous control of the Quantum Approximate Optimization Algorithm via Reinforcement Learning*, arXiv:1911.09682.
[58] M. M. Wauters, E. Panizon, G. B. Mbeng, and G. E. Santoro, *Reinforcement-Learning-Assisted Quantum Optimization*, Phys. Rev. Research **2**, 033446 (2020).
[59] M. Bilkis, M. Rosati, R. M. Yepes, and J. Calsamiglia, *Real-Time Calibration of Coherent-State Receivers: Learning by Trial and Error*, Phys. Rev. Research **2**, 033295 (2020).
[60] S. Russel and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. (Pearson, 2020), https://www.amazon.com/Artificial-Intelligence-A-Modern-Approach/dp/0134610997.
[61] J. Koch, T. M. Yu, J. Gambetta, A. A. Houck, D. I. Schuster, J. Majer, A. Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf, *Charge-Insensitive Qubit Design Derived from the Cooper Pair Box*, Phys. Rev. A **76**, 042319 (2007).
[62] K. Mølmer, Y. Castin, and J. Dalibard, *Monte Carlo Wave-Function Method in Quantum Optics*, J. Opt. Soc. Am. B **10**, 524 (1993).
[63] M. Jerger, A. Kulikov, Z. Vasselin, and A. Fedorov, *In Situ Characterization of Qubit Control Lines: A Qubit as a Vector Network Analyzer*, Phys. Rev. Lett. **123**, 150501 (2019).
[64] M. A. Rol, L. Ciorciaro, F. K. Malinowski, B. M. Tarasinski, R. E. Sagastizabal, C. C. Bultink, Y. Salathe, N. Haandbaek, J. Sedivy, and L. Dicarlo, *Time-Domain Characterization and Correction of On-Chip Distortion of*

Control Pulses in a Quantum Processor, Appl. Phys. Lett. **116**, 054001 (2020).

[65] T. Propson, B. E. Jackson, J. Koch, Z. Manchester, and D. I. Schuster, *Robust Quantum Optimal Control with Trajectory Optimization*, Phys. Rev. Applied **17**, 014036 (2022).

[66] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, *Playing Atari with Deep Reinforcement Learning*, arXiv:1312.5602.

[67] S. Hochreiter and J. Schmidhuber, *Long Short-Term Memory*, Neural Comput. **9**, 1735 (1997).

[68] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, *Proximal Policy Optimization Algorithms*, arXiv:1707.06347.

[69] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevX.12.011059 for introduction to PPO algorithm, implementational details, and list of hyperparameters.

[70] S. Krastanov, V. V. Albert, C. Shen, C.-L. Zou, R. W. Heeres, B. Vlastakis, R. J. Schoelkopf, and L. Jiang, *Universal Control of an Oscillator with Dispersive Coupling to a Qubit*, Phys. Rev. A **92**, 040303(R) (2015).

[71] R. W. Heeres, B. Vlastakis, E. Holland, S. Krastanov, V. V. Albert, L. Frunzio, L. Jiang, and R. J. Schoelkopf, *Cavity State Manipulation Using Photon-Number Selective Phase Gates*, Phys. Rev. Lett. **115**, 137002 (2015).

[72] P. Reinhold, S. Rosenblum, W.-L. Ma, L. Frunzio, L. Jiang, and R. J. Schoelkopf, *Error-Corrected Gates on an Encoded Qubit*, Nat. Phys. **16**, 822 (2020).

[73] W.-L. Ma, M. Zhang, Y. Wong, K. Noh, S. Rosenblum, P. Reinhold, R. J. Schoelkopf, and L. Jiang, *Path-Independent Quantum Gates with Noisy Ancilla*, Phys. Rev. Lett. **125**, 110503 (2020).

[74] T. Fösel, S. Krastanov, F. Marquardt, and L. Jiang, *Efficient Cavity Control with SNAP Gates*, arXiv:2004.14256.

[75] M. Kliesch and I. Roth, *Theory of Quantum System Certification*, PRX Quantum **2**, 010201 (2021).

[76] D. I. Schuster, A. A. Houck, J. A. Schreier, A. Wallraff, J. M. Gambetta, A. Blais, L. Frunzio, J. Majer, B. Johnson, M. H. Devoret *et al.*, *Resolving Photon Number States in a Superconducting Circuit*, Nature (London) **445**, 515 (2007).

[77] W. Pfaff, C. J. Axline, L. D. Burkhart, U. Vool, P. Reinhold, L. Frunzio, L. Jiang, M. H. Devoret, and R. J. Schoelkopf, *Controlled Release of Multiphoton Quantum States from a Microwave Cavity Memory*, Nat. Phys. **13**, 882 (2017).

[78] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, *TensorFlow: A System for Large-Scale Machine Learning*, arXiv:1605.08695.

[79] S. T. Flammia and Y.-K. Liu, *Direct Fidelity Estimation from Few Pauli Measurements*, Phys. Rev. Lett. **106**, 230501 (2011).

[80] M. P. da Silva, O. Landon-Cardinal, and D. Poulin, *Practical Characterization of Quantum Devices without Tomography*, Phys. Rev. Lett. **107**, 210404 (2011).

[81] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2010).

[82] K. Duivenvoorden, B. M. Terhal, and D. Weigand, *Single-Mode Displacement Sensor*, Phys. Rev. A **95**, 012305 (2017).

[83] K. Noh, S. M. Girvin, and L. Jiang, *Encoding an Oscillator into Many Oscillators*, Phys. Rev. Lett. **125**, 080503 (2020).

[84] B. M. Terhal and D. Weigand, *Encoding a Qubit into a Cavity Mode in Circuit QED Using Phase Estimation*, Phys. Rev. A **93**, 012315 (2016).

[85] C. Flühmann, T. L. Nguyen, M. Marinelli, V. Negnevitsky, K. Mehta, and J. P. Home, *Encoding a Qubit in a Trapped-Ion Mechanical Oscillator*, Nature (London) **566**, 513 (2019).

[86] P. Campagne-Ibarcq, A. Eickbusch, S. Touzard, E. Zalys-Geller, N. E. Frattini, V. V. Sivak, P. Reinhold, S. Puri, S. Shankar, R. J. Schoelkopf *et al.*, *Quantum Error Correction of a Qubit Encoded in Grid States of an Oscillator*, Nature (London) **584**, 368 (2020).

[87] B. Royer, S. Singh, and S. M. Girvin, *Stabilization of Finite-Energy Gottesman-Kitaev-Preskill States*, Phys. Rev. Lett. **125**, 260509 (2020).

[88] B. de Neeve, T. L. Nguyen, T. Behrle, and J. Home, *Error Correction of a Logical Grid State Qubit by Dissipative Pumping* (2022) 10.1038/s41567-021-01487-7.

[89] B. Vlastakis, G. Kirchmair, Z. Leghtas, S. E. Nigg, L. Frunzio, S. M. Girvin, M. Mirrahimi, M. H. Devoret, and R. J. Schoelkopf, *Deterministically Encoding Quantum Information Using 100-Photon Schrodinger Cat States*, Science **342**, 607 (2013).

[90] C. Sayrin, I. Dotsenko, X. Zhou, B. Peaudecerf, T. Rybarczyk, S. Gleyzes, P. Rouchon, M. Mirrahimi, H. Amini, M. Brune *et al.*, *Real-Time Quantum Feedback Prepares and Stabilizes Photon Number States*, Nature (London) **477**, 73 (2011).

[91] C. Shen, K. Noh, V. V. Albert, S. Krastanov, M. H. Devoret, R. J. Schoelkopf, S. M. Girvin, and L. Jiang, *Quantum Channel Construction with Circuit Quantum Electrodynamics*, Phys. Rev. B **95**, 134501 (2017).

[92] A. Kenfack and K. Zyczkowski, *Negativity of the Wigner Function as an Indicator of Non-classicality*, J. Opt. B **6**, 396 (2004).

[93] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, *High-Dimensional Continuous Control Using Generalized Advantage Estimation*, arXiv:1506.02438.

[94] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, *Benchmarking Deep Reinforcement Learning for Continuous Control*, *33rd International Conference on Machine Learning* 3, 2001 (2016), https://arxiv.org/abs/1604.06778.

[95] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need*, arXiv:1706.03762.

[96] S. Krastanov, S. Zhou, S. T. Flammia, and L. Jiang, *Stochastic Estimation of Dynamical Variables*, Quantum Sci. Technol. **4**, 035003 (2019).

[97] S. Krastanov, K. Head-Marsden, S. Zhou, S. T. Flammia, L. Jiang, and P. Narang, *Unboxing Quantum Black Box Models: Learning Non-Markovian Dynamics*, arXiv:2009.03902.

[98] E. Flurin, L. S. Martin, S. Hacohen-Gourgy, and I. Siddiqi, *Using a Recurrent Neural Network to Reconstruct Quantum Dynamics of a Superconducting Qubit from Physical Observations*, Phys. Rev. X **10**, 011006 (2020).

[99] L. Banchi, E. Grant, A. Rocchetto, and S. Severini, *Modelling Non-Markovian Quantum Processes with Recurrent Neural Networks*, New J. Phys. **20**, 123030 (2018).

[100] A. Plaat, W. Kosters, and M. Preuss, *Deep Model-Based Reinforcement Learning for High-Dimensional Problems, A Survey*, arXiv:2008.05598.

[101] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, *SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python*, Nat. Methods **17**, 261 (2020).

[102] B. Kneer and C. K. Law, *Preparation of Arbitrary Entangled Quantum States of a Trapped Ion*, Phys. Rev. A **57**, 2096 (1998).

[103] S. Haroche and J.-M. Raimond, *Exploring the Quantum* (Oxford University Press, New York, 2006).

[104] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, *Variational Quantum Algorithms*, Nat. Rev. Phys. **3**, 625 (2021).

[105] M. A. Nielsen, *A Simple Formula for the Average Gate Fidelity of a Quantum Dynamical Operation*, Phys. Lett. A **303**, 249 (2002).