Sample-efficient Model-based Reinforcement Learning for Quantum Control

Irtaza Khalid,[1,*] Carrie A. Weidner,[2,†] Edmond A. Jonckheere,[3,‡] Sophie G. Schirmer,[4,§] and Frank C. Langbein[1,¶]

[1] *School of Computer Science and Informatics, Cardiff University, Cardiff, CF24 4AG, UK*
[2] *Quantum Engineering Technology Laboratories, H. H. Wills Physics Laboratory and Department of Electrical and Electronic Engineering, University of Bristol, Bristol BS8 1FD, UK*
[3] *Department of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA 90007, USA*
[4] *Department of Physics, Swansea University, Swansea, SA2 8PP, UK*

We propose a model-based reinforcement learning (RL) approach for noisy time-dependent gate optimization with reduced sample complexity over model-free RL. Sample complexity is defined as the number of controller interactions with the physical system. Leveraging an inductive bias, inspired by recent advances in neural ordinary differential equations (ODEs), we use an auto-differentiable ODE, parametrized by a learnable Hamiltonian ansatz, to represent the model approximating the environment, whose time-dependent part, including the control, is fully known. Control alongside Hamiltonian learning of continuous time-independent parameters is addressed through interactions with the system. We demonstrate an order of magnitude advantage in sample complexity of our method over standard model-free RL in preparing some standard unitary gates with closed and open system dynamics, in realistic computational experiments incorporating single shot measurements, arbitrary Hilbert space truncations, and uncertainty in Hamiltonian parameters. Also, the learned Hamiltonian can be leveraged by existing control methods like GRAPE for further gradient-based optimization with the controllers found by RL as initializations. Our algorithm, which we apply to nitrogen vacancy (NV) centers and transmons, is well suited for controlling partially characterized one- and two-qubit systems.

I. INTRODUCTION

Control of quantum devices for practical applications requires overcoming a unique set of challenges [1]. One is to find robust controls for noisy systems, where typical noise sources include control and feedback noise, system parameter mischaracterization, measurement and state preparation errors, decoherence and cross-talk [2]. To achieve scalable, fault-tolerant quantum devices [3–5], control algorithms must produce controls resilient to such noise. Reinforcement learning (RL) approaches appear more likely to find robust controls for certain applications [6] at the cost of requiring a large number of measurements from the quantum device (samples). We propose a model-based RL approach to address this problem.

Typically, a quantum control problem is formulated as an open-loop optimization problem based on a model [1, 7–9], which may be constructed *ab initio* or obtained via a process tomography approach. During optimization there is no interaction between the physical system to be controlled and the control algorithm. The underlying assumption is that the model represents the system sufficiently accurately. This class of control algorithms has low sample complexity (high sample efficiency) represented by the number of optimization function calls until successful termination. The reason for this is, generally, that an analytical model, in particular gradient information, can be leveraged. This is a strong assumption, at least in the noisy intermediate scale quantum era, where noise impedes perfect characterization of quantum devices. However, the approach has merit, since significant thought goes into modelling and engineering quantum devices [10].

Alternatively, RL seeks an optimal control via interaction with the physical system, building models to various degrees. It successfully addresses challenging, noisy quantum control problems with the promise of inherent robustness [6, 11–15]. There are also gradient-free approaches [16] and methods that estimate gradients using variations of automatic differentiation [10, 17–20].

RL approaches utilizing only measurements without prior information do not suffer from model bias. Moreover, they usually optimize an average controller performance over the noise in the system, yielding inherently robust controllers [12]. However, this means the number of optimization function calls becomes prohibitively large, and RL's high sample complexity is a core problem limiting its practical applicability [21]. This is not surprising as without a prior model little or no information is available to the optimization algorithm and all information must be obtained via measurements.

Despite this inherent restriction, in recent times, RL has been deployed on real quantum devices for parametrized pulse-level gate optimization [22], improving the performance of quantum error correcting codes [23] and fluxonium gate parameter optimization [24]. In line with forthcoming analysis, the sample complexity of these RL experiments is estimated to be around 10^4, 10^3 and 10^4, respectively, excluding the

* khalidmi@cardiff.ac.uk
† c.weidner@bristol.ac.uk
‡ jonckhee@usc.edu
§ lw1660@gmail.com
¶ frank@langbein.org

cost of estimating observables using single-shot measurements. These costs are smaller than direct or ab initio applications of RL, which consume around 10^6 samples [21], as the aforementioned works, to differing extent, exploit specific knowledge of the quantum system to frame the problem to be easier to optimize for the RL agent. More specifically, these works use custom RL adaptations for each problem, e.g., fine-tuning solutions already found by other optimization algorithms as the final step during control preparation in Ref. [24], or exploiting some experimental structure that simplifies finding optimal controls in Ref. [22]. In the present paper, we remain generic in our ignorance of the system Hamiltonian during acquisition of optimal controls to demonstrate the general utility of our approach without inducing constraining (and potentially incorrect if not confidently known) biases on the learning problem. We note, however, that there is significant scope for sample-efficiency reductions. For example, the use of our model-based RL algorithm would make RL, in general, extensible to a wider class of quantum control experiments.

In classical RL, high sample complexity is typically addressed using model-based methods, which construct a model from scratch using information obtained from measurements. Such methods result in reduced sample complexity for benchmark problems [25]. They are successful if the model and the measurements (samples) obtained during training possess some generalizability [26, 27] that is captured by a function approximator (usually a neural network). However, methods involving universal function approximation of dynamic trajectories are unstable. This is because learning can be hindered by the very large space of trajectories, and interpolating from insufficient sample trajectories can be shallow or incorrect [28]. More importantly, for quantum data, it is known that a time-independent Hamiltonian can generate many unitary propagators, so estimating the model may imply learning the entire Hilbert space of propagators for a particular control problem which is often intractable. This motivates learning the dynamical generator, i.e., the Hamiltonian, instead of the propagators.

In this paper, we propose a model-based RL method for time-dependent, noisy gate preparation where the model is given by an ordinary differential equation (ODE), differentiable with respect to model parameters [29]. ODE trajectories do not intersect [30–32], which constrains the space of potential models for learning and makes learning robust to noise. We parameterize the Hamiltonian by known time-dependent controls and unknown time-independent (system) parameters, which, in addition, makes the model interpretable.

We show that combining the inductive bias from this ODE model with partially correct knowledge (assuming the controls are known but not the time-independent system Hamiltonian) reduces the sample complexity compared to model-free RL by at least an order of magnitude.

It has recently been shown that inductive biases, i.e., encoding the symmetries of the problem into the architecture of the model space, such as the translation equivariance of images in the convolution operation [33], leads to stronger out-of-distribution generalization by the learned model. This is because inductive biases impose strong priors on the space of models such that training involves exploring a smaller subset of the space to find an approximately correct model.

We demonstrate improvement over the sample-efficient soft-actor critic (SAC) model-free RL algorithm [34] for performing noisy gate control in leading quantum computing architectures: nitrogen vacancy (NV) centers (one and two qubits) [35], and transmons (two qubits) [36], subject to dissipation and single-shot measurement noise. We also show that the learned Hamiltonian can be leveraged to optimize the controllers found by our RL method further using GRAPE [7, 9].

Our approach is similar in spirit to Ref. [37] where a novel Hamiltonian learning protocol via quantum process tomography is proposed for the purpose of model-predictive control. The complete Hamiltonian (including the control and system parts) is identified term by term via a Zero-Order Hold (ZOH) method, where only one term is turned on at a time, e.g., by setting the control parameters to zero, and learned individually using optimization over the Stiefel manifold. As a side remark, a sample complexity advantage between learning the Hamiltonian with quantum control than without it has recently been shown [38]. The learned Hamiltonian is then used to obtain a viable control sequence for a variety of state and gate preparation problems for closed (unitary) systems under the influence of initial state preparation errors. While it is possible for our Hamiltonian learning protocol to also learn the full Hamiltonian using the ZOH method, we focus on the problem of improving the sample complexity of RL in this paper through the incorporation of a partially known physics-inspired model. Furthermore, our focus is also directed on the interplay of concurrently learning the model and controlling the system in noisy closed and open system settings.

This paper is organized as follows: in Sec. II we define the open and closed system control problems including our setup to simulate single-shot measurements and the RL control framework; Sec. III describes the model-based version of the RL control framework and Sec. IV presents numerical studies for some realistic example control problems on the system architectures described above in noisy and ideal settings and how to leverage the learned system Hamiltonian using GRAPE.

II. THE QUANTUM CONTROL PROBLEM

We briefly introduce the quantum control problem for open and closed quantum systems and describe how we estimate the propagators from measurements, needed for our RL approach.

A. Closed System Dynamics

Consider a quantum system that is represented by an effective Hamiltonian $H(t)$ in the space of complex Hermitian $n \times n$ matrices

$$H(\mathbf{u}(t), t) = H_0 + H_c(\mathbf{u}(t), t), \tag{1}$$

where H_0 is the time-independent system Hamiltonian and H_c is the control Hamiltonian parametrized by time-dependent controls $\mathbf{u}(t)$. Its closed-system dynamics are governed by the Schrödinger equation,

$$\frac{\mathrm{d}U(\mathbf{u}(t), t)}{\mathrm{d}t} = -\frac{i}{\hbar} H(\mathbf{u}(t), t) U(\mathbf{u}(t), t), \quad U(t=0) = \mathbb{1}, \tag{2}$$

where $U(\mathbf{u}(t), t)$ is the unitary propagator representing the state evolution. Its fidelity to realize a target gate U_{target} is

$$F(U_{\text{target}}, U(\mathbf{u}(t), t)) = \frac{1}{n^2} \left| \mathrm{Tr}\left[U_{\text{target}}^\dagger U(\mathbf{u}(t), t) \right] \right|^2. \tag{3}$$

The control problem to implement U_{target} is

$$\mathbf{u}^*(t^*) = \underset{\mathbf{u}(t),\ t \leqslant T}{\arg\max}\, F(U_{\text{target}}, U(\mathbf{u}(t), t)), \tag{4}$$

where $\mathbf{u}^*(t^*)$ are the optimized control parameters for an optimized final time $t^* \leqslant T$.

B. Open System Dynamics

For open system dynamics consider an arbitrary state with density matrix ρ for $\log_d n$ qudits evolving according to the master equation [39, 40]

$$\frac{\mathrm{d}\rho(t)}{\mathrm{d}t} = -\frac{i}{\hbar}[H(\mathbf{u}(t), t), \rho] + \mathfrak{L}(\rho(t)), \tag{5}$$

where $\mathfrak{L}(t)$ describes the Markovian decoherence and dephasing dynamics (i.e., the environment),

$$\mathfrak{L}(\rho(t)) = \sum_d \gamma_d \left(l_d \rho l_d^\dagger - \frac{1}{2}\{l_d^\dagger l_d, \rho\} \right), \tag{6}$$

and l_d is a decoherence operator that can be non-unitary.

To characterize the gate implemented by $\mathbf{u}(t)$, we need to consider the evolution of a complete orthonormal basis of states, $\{\rho_k\}_{k=1}^{n^2}$. For this we introduce the Liouville superoperator matrix $\mathbf{X}$ that acts on an arbitrary vectorized state $\boldsymbol{\rho}$ (e.g., obtained by stacking the matrix columns) to produce the evolution

$$\boldsymbol{\rho}(t) = \mathbf{X}(t)\boldsymbol{\rho}(t=0). \tag{7}$$

This is equivalent to the tensor-matrix evolution [41]

$$\rho(t)_{mn} = \sum_{\mu,\nu} X_{nm,\nu\mu}(t) \rho_{\mu\nu}(t=0). \tag{8}$$

$X_{nm,\nu\mu}(t)$ is a fourth order tensor (used to refer to multi-dimensional arrays in this context) form of $\mathbf{X}(t)$ that encodes the evolution of the state element $\rho_{\mu\nu}$.

Thus, similar to Eq. (2), we define a superoperator $X(\mathbf{u}(t), t)$ which encodes the evolution of $\{\rho_k\}_{k=1}^{n^2}$ and follows the linear ODE

$$\frac{\mathrm{d}\mathbf{X}(\mathbf{u}(t), t)}{\mathrm{d}t} = -\frac{i}{\hbar}(\mathbf{L}_0 + i\mathbf{L}_1)\mathbf{X}(\mathbf{u}(t), t), \quad \mathbf{X}(t=0) = \mathbb{1} \tag{9}$$

where $\mathbf{L}_0, \mathbf{L}_1$ represent the superoperator version of the commutator map $[H(\mathbf{u}(t), t), \cdot]$ and $\mathfrak{L}(\cdot)$ the Markovian decoherence and dephasing dynamics.

We factorize out an imaginary prefactor i to the left in Eq. (9) to unify the ODE for open and closed system dynamics. For $\mathfrak{L} \equiv \mathbf{0}$, the above reduces to the closed system dynamics of Eq. (2). For open dynamics, to be faithful to experimental limitations, we implement single-shot noise when estimating the gate, i.e., process tomography. We transform the superoperator $X_{nm,\nu\mu}$ to the Choi matrix $\Phi/\mathrm{Tr}[\Phi]$ that is given by index reshuffling or partial transpose (and more formally a contravariant-covariant change of coordinates) [41, 42],

$$\Phi_{nm,\mu\nu} = X_{\nu m,\mu n}. \tag{10}$$

In Sec. IV, we use this for open and closed dynamics. Estimating Φ is possible using ancilla-assisted quantum process tomography (AAPT) and the Choi-Jamiolkowski isomorphism [43–45] for $2\log_d n$-qudit states and $\log_d n$-qudit gates. Analogously to the above, Φ has a matrix version $\boldsymbol{\Phi}$. In this paper, we decompose $\boldsymbol{\Phi}$ over a generalized $\mathfrak{su}(n^2)$'s algebra basis $\{P_k\}_{k=1}^{n^4-1}$, e.g., Gell-Mann matrices [46],

$$\frac{\boldsymbol{\Phi}}{\mathrm{Tr}[\boldsymbol{\Phi}]} = \frac{\mathbb{1}}{n^2} + \sum_{k=2}^{n^4-1} q_k P_k \tag{11}$$

whose coefficients are

$$q_k = \frac{\mathrm{Tr}[P_k \boldsymbol{\Phi}]}{\mathrm{Tr}[\boldsymbol{\Phi}]} \in [-1, 1]. \tag{12}$$

q_k can be modelled as a binomial random variable $\mathrm{Bin}(M, p_k)$ with probability $p_k = \frac{1}{2}(1 + q_k)$ where M is the number of single-shot (Bernoulli) measurements [47]. The Gell-Mann matrices are a generalization of the Pauli matrices and the corresponding physical measurement operations are akin to measuring qudit energy levels in an informationally complete basis.

We measure the faithfulness of the implemented gate $\boldsymbol{\Phi}(\mathbf{u}(t), t)$ w.r.t. the target gate (as another Choi state) $\boldsymbol{\Phi}_{\text{target}}$ using the generalized state-fidelity [48],

$$\begin{aligned} F(\boldsymbol{\Phi}(\mathbf{u}(t), t), \boldsymbol{\Phi}_{\text{target}}) &= \frac{\mathrm{Tr}[\boldsymbol{\Phi}(\mathbf{u}(t), t)\boldsymbol{\Phi}_{\text{target}}]}{\mathrm{Tr}[\boldsymbol{\Phi}(\mathbf{u}(t), t)]\,\mathrm{Tr}[\boldsymbol{\Phi}_{\text{target}}]} \\ &= \frac{1}{n^4} + \sum_{k=2}^{n^4-1} q_k^{\text{target}} q_k. \end{aligned} \tag{13}$$

Analogously to the closed case, the open control problem is to find an optimal control $\mathbf{u}^*(t^*)$ for an optimal final time $t^* \leqslant T$ (with T being the fixed upper bound), such that

$$\mathbf{u}^*(t^*) = \underset{\mathbf{u}(t),\ t\leqslant T}{\arg\max}\, F(\mathbf{\Phi}(\mathbf{u}(t), t), \mathbf{\Phi}_{\text{target}}). \tag{14}$$

C. Discretization

The exact solution of the time-dependent general dynamics discussed in Eq. (14) is given by the time-ordered operator

$$\mathbf{E}(t^*, \mathbf{u}^*(t^*)) = \mathcal{T} \exp\left(\int_0^{t^*} dt' \ -\frac{i}{\hbar}\mathbf{G}(t', \mathbf{u}^*(t'))\right)$$

for a unitary or Lindbladian generator $\mathbf{G}$. In practice, we solve for a piece-wise constant version of the dynamics represented by N fixed steps of $\Delta t = T/N$ of the final time T. Thus, $\mathbf{E}(\mathbf{u}(t), t)$ is discretized, which amounts to fixing $\mathbf{u}(t) = \mathbf{u}_m$ to be constant for each timestep such that $\mathbf{u}_m \in \mathbb{C}^{m\times C}$ is a finite dimensional array where C is the number of controls per timestep in the vector u_l parametrizing $H_c(u_l, t_l)$ and m is the number of total timesteps in the pulse, with $m \leqslant N$ for a maximum number of pulse segments N. The propagator is

$$\mathbf{E}(t, \mathbf{u}(t)) := \mathbf{E}(\mathbf{u}_m) = \prod_{l=1}^{m} \exp\left(-\frac{i}{\hbar}\Delta t \mathbf{G}(t_l, \mathbf{u}(t_l))\right). \tag{15}$$

The control problems in Eqs. (4) and (14) are equivalent to

$$\mathbf{u}^*_m = \underset{\mathbf{u}_m=[u_1,\ldots,u_m]\in\mathbb{X}, m\leqslant N}{\arg\max} \mathcal{F}(\mathbf{\Phi}(\mathbf{E}(\mathbf{u}_m)), \mathbf{\Phi}(\mathbf{E}_{\text{target}})) \tag{16}$$

for a fidelity $\mathcal{F}$ and the time. $\mathbf{u}_m$ is constrained to some maximum and minimum values given by $\mathbb{X} = \{\mathbf{u}_m : \forall c, l\ u_{\min} \leqslant u_{cl} \leqslant u_{\max} \in \mathbb{C}\}$. The constraints are applied separately to the real and imaginary parts of the components of $\mathbf{u}_m$.

III. MODEL-BASED REINFORCEMENT LEARNING CONTROL

We give a brief overview of RL, followed by explaining our model-based RL approach. An excellent introduction can be found in Ref. [21].

A. Reinforcement Learning for Quantum Control

The RL problem is usually treated as a sequential Markov decision problem (MDP) on the space of states, actions, transition probabilities and rewards:

Algorithm 1: Reinforcement learning loop

```
Initialize empty dataset D, parametrized random
  policy π_θ, k ← 0
Observe initial state s_0
while k < T/Δt do
    Execute a_k ← π_θ(· | s_k)
    Observe s_{k+1}, r_k ← E(s_k, a_k)
    Store D ← D ∪ {(s_k, s_{k+1}, a_k, r_k)}
    k ← k + 1
  // if require update: perform model-free
  update of parameters (e.g. policy π_θ )
```

$(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R})$. This describes an environment for consecutive one-step transitions, indexed by $k = 1, 2, \ldots$, from current state $\mathbf{s}_k \in \mathcal{S}$ to next state $\mathbf{s}_{k+1} \in \mathcal{S}$ if an RL agent executes action $\mathbf{a}_k \in \mathcal{A}$, yielding immediate scalar reward $\mathrm{r}_k \in \mathcal{R}$. The environment is generally probabilistic, so $\mathcal{P}(\mathbf{s}_{k+1} \,|\, \mathbf{s}_k, \mathbf{a}_k)$ is the probability that the agent is in state $\mathbf{s}_{k+1}$ after executing $\mathbf{a}_k$ in state $\mathbf{s}_k$. An RL agent follows a policy function that is represented by a conditional probability distribution $\pi(\mathbf{a}_k \,|\, \mathbf{s}_k)$: the probability of taking action $\mathbf{a}_k$ after observing the state $\mathbf{s}_k$.

The quantum control problem can be represented as an RL problem by sequentially constructing the control amplitudes as actions, using the unitary propagator the control implements as the state with the reward as the fidelity:

$$\mathbf{a}_k = u_k, \tag{17a}$$

$$\mathbf{s}_k = \prod_{l=1}^{k} \exp\left(-\frac{i}{\hbar}\Delta t \mathbf{G}(t_l, u_l)\right), \tag{17b}$$

$$\mathrm{r}_k = \mathcal{F}(\mathbf{\Phi}(\mathbf{E}(\mathbf{u}_k)), \mathbf{\Phi}(\mathbf{E}_{\text{target}})). \tag{17c}$$

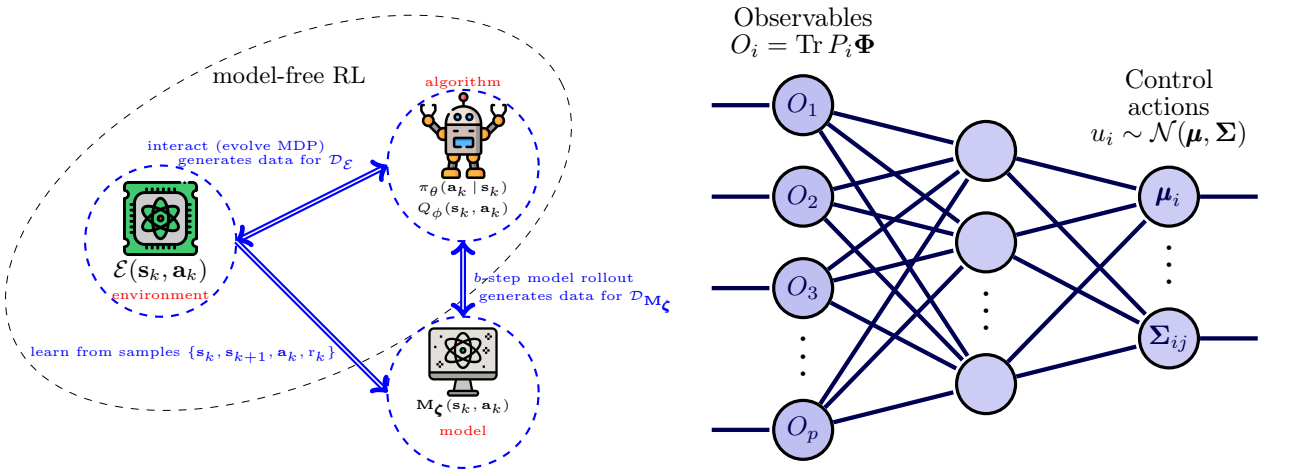
As this is deterministic the probabilities $\mathcal{P}$ are trivial, and we have a simple environment function $\mathcal{E} : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathcal{R}$, mapping the current state and action (s, a) to the next state and reward (s', r). In model-free RL (see Algorithm 1), a discounted sum of expected rewards, called the returns,

$$\eta(\pi) := \mathbb{E}_{\mathbf{a}_t\sim\pi}\left[\sum_{k=0}^{\infty} \gamma^k \mathrm{r}_k\right] \tag{18}$$

is maximized, where $\mathbb{E}_{x\sim P}[\cdot] = \int_{\mathcal{X}} dx\ P(x)[\cdot]$ is the expectation operator and $0 \leqslant \gamma \leqslant 1$ is a discount factor.

However, Refs. [34, 49] observe that adding an entropy maximizing term for the policy $\pi(\mathbf{a}_k \,|\, \mathbf{s}_k)$ to the optimization objective encourages exploration of the state space $\mathcal{S}$, improves the learning rate of the agent and reduces the relative number of samples needed, compared to other standard RL algorithms. The maximum entropy objective or the entropy-regularized cumulative reward function J for N steps is

$$J(\pi) = \sum_{k=0}^{N} \gamma^k \mathbb{E}_{(\mathbf{s}_k, \mathrm{r}_k)\sim\mathcal{E}_\pi}\left[r_k + \alpha J_1(\mathbf{s}_k)\right] \tag{19}$$



(a) **Model-based RL** (b) **Policy function** $\pi_\theta(\mathbf{a}_k \mid \mathbf{s}_k)$

Figure 1. A schematic of model-based RL is given in (a). The arrow-head implies direction of affect of the edge between a source and a sink node. The agent or policy function π_θ interacts with the RL environment modelled as MDP to collect data $\{\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k\}$. This encompasses model-free RL. The data is then used to train the model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_k, \mathbf{a}_k)$. The model is trained until some quality measure like the validation prediction error on some untrained-upon data from the environment plateaus indicating that the training is complete. Then, it is used to generate synthetic data through a b-step rollout in which the policy interacts with the model b times. The policy parameters θ (and the state-action value function parameters ϕ) are optimized using the real and model generated data. In (b), we visualize the policy inputs as the gate-characterizing observables (unitary or Lindblad) about the Choi matrix $\mathbf{\Phi}$ given by Eq. (12) and the tunable outputs are the parameters of a multivariate Gaussian distribution, i.e., the mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$. The controls u_i are drawn from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

where $\mathcal{E}_\pi$ represents the environment's state-action probability distribution induced by the policy π, α is an optimizable temperature parameter (signifying the importance of exploration in the objective), and $J_1(\mathbf{s}_k)$ is the entropy of the policy function $\pi(\cdot \mid \mathbf{s}_k)$ conditional on the kth state $\mathbf{s}_k$,

$$J_1(\mathbf{s}_k) = -\mathbb{E}_{x \sim \pi(\cdot \mid \mathbf{s}_k)} \left[\log(\pi(x \mid \mathbf{s}_k))\right]. \tag{20}$$

Thus, the RL control problem becomes a problem of finding the optimal control policy π^* given by

$$\pi^* = \arg\max_\pi J(\pi). \tag{21}$$

This is exactly solvable for tabular MDPs using dynamic programming and heuristically with neural network function approximation for continuous MDPs.

B. Model-Based Reinforcement Learning

In this paper, we use the soft actor-critic (SAC) algorithm [34] as our base (model-free) RL algorithm. For brevity, we only highlight parts of SAC relevant to us. A detailed description can be found in the original paper [34]. We use a neural network policy function $\pi_\theta(\mathbf{a}_k \mid \mathbf{s}_k)$, with the optimizable parameters θ, as the actor and the state-action value function $Q_\phi(\mathbf{s}_k, \mathbf{a}_k) = \mathbb{E}_{(\mathbf{s}_k, \mathbf{a}_k) \sim \mathcal{E}_\pi} \left[\sum_{k=0}^\infty \gamma^k (\mathrm{r}(\mathbf{s}_k, \mathbf{a}_k) + \alpha J_1(\mathbf{s}_k))\right]$ as the neural network critic with parameters ϕ. Both π and Q are simple multilayer perceptrons. In essence, the critic is used to reduce the high variance in the reward function due to the non-stationary nature of the MDP. It is trained by having its predictions match the estimated $\hat{Q}$ values obtained for some data $\{\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k\}_{k=1}^b$ obtained from a b-length rollout (number of interactions) with $\mathcal{E}$. The actor is trained by minimizing the loss function

$$J'(\pi_\theta) = \mathbb{E}_{(\mathbf{s}_k, \mathbf{a}_k) \sim \mathcal{E}_{\pi_\theta}} \left[\alpha \log \pi_\theta(\mathbf{a}_k \mid \mathbf{s}_k) - Q_\phi(\mathbf{s}_k, \mathbf{a}_k)\right], \tag{22}$$

which is equivalent to maximizing J in Eq. (19). For SAC, this policy optimization is carried out heuristically using neural networks to approximate the policy function π_θ. We define the number of agent-environment interactions needed to find an approximately optimal policy π^* as the *sample complexity*. Moreover, the policy outputs parametrize the mean and covariance $\boldsymbol{\mu}, \boldsymbol{\Sigma}$ of a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ from which the control vector $\mathbf{u}$ is drawn. For the quantum control problem in Eq. (16), we are usually just concerned with finding an optimal action sequence $\mathbf{u}^*$ producing the maximum intermediate reward $\boldsymbol{r}_k$ rather than the optimal policy function π^* which can be produced by a suboptimal policy, too.

SAC can be augmented to incorporate a model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_k, \mathbf{a}_k)$ that approximates the dynamics of $\mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$ using the policy's interaction data $\mathcal{D}$ [27] where ζ are the model's learnable parameters. The model acts as a proxy for the environment and allows the policy to do MDP rollouts (steps) to augment the interaction data. For this to work, the dynamics obtained from interacting

Algorithm 2: Learnable Hamiltonian model-based soft actor critic (LH-MBSAC)

Input :

H_c	control Hamiltonian (time-dependent part of $H(t)$ in Eq. (1))
$T, \Delta t, M$	max time, timestep size, number of single shot measurements (if open system to estimate $\mathbf{\Phi}$ using Eq. (12))
$\mathbf{E}_{\text{target}}$	target gate
$W, C, b,$ `tol`	Epochs, timesteps, rollout length, validation loss tolerance (which is a problem-specific hyperparameter)

Output:

$\mathbf{u}^*$	Approximately optimal 2D array of controls that solves Eq. (16)
$\theta,\ \phi,\ \boldsymbol{\zeta}$	Optimized parameters of the policy, critic and learned model

Initialize empty environment dataset $\mathcal{D}_{\mathcal{E}}$, model dataset $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}}$, random policy π_θ

`// collect random model training data`

Populate $\mathcal{D}_{\mathcal{E}}$ using uniform random policy π_θ with Algorithm 1 without updates ▷ `randomly explore the environment` $\mathcal{E}$ `state space`

for W *epochs* **do**

  `// Train model`

  Sample a batch of training and validation data $D_{\text{train}}, D_{\text{val}} \sim \mathcal{D}_{\mathcal{E}}$ and minimize $L_{\text{model}}(D_{\text{train}})$ in Eq. (24)

  for C *timesteps* **do**

    `// agent-environment interaction`

    Execute $\mathbf{a}_k \leftarrow \pi_\theta(\cdot|\,\mathbf{s}_k)$, observe $\mathbf{s}_{k+1}, \mathrm{r}_k \leftarrow \mathcal{E}(\mathbf{s}_k, \mathbf{a}_k)$ and store data $\mathcal{D}_{\mathcal{E}} \cup \{(\mathbf{s}_k, \mathbf{s}_{k+1}, \mathbf{a}_k, \mathrm{r}_k)\}$

    if $L_{model}(D_{val}) <$ `tol` **then**

      `// agent-model interaction`

      Sample uniformly a batch of initial states $\{\mathbf{s}_k\} \sim \mathcal{D}_{\mathcal{E}}$, $k \leftarrow 0$

      for k' *in* $\{1, \cdots, b\}$ **do**

        Execute $\mathbf{a}_{k'} \leftarrow \pi_\theta(\cdot|\,\mathbf{s}_{k'})$ and observe $\mathbf{s}_{k'+1}, \mathrm{r}_{k'} \leftarrow \mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k'}, \mathbf{a}_{k'})$ ▷ b-`length model rollout`

        Store $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \leftarrow \mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \cup \{(\mathbf{s}_{k'}, \mathbf{s}_{k'+1}, \mathbf{a}_{k'}, \mathrm{r}_{k'})\}$

        $k' \leftarrow k' + 1$,

    Train policy by minimizing $J'(\pi_\theta)$ in Eq. (22) using $\mathcal{D}_{\mathbf{M}_{\boldsymbol{\zeta}}} \cup \mathcal{D}_{\mathcal{E}}$

with $\mathbf{M}_{\boldsymbol{\zeta}}$ must be close enough to the true dynamics of $\mathcal{E}$ to allow the policy to maximize J. By improving the returns $\hat{\eta}(\pi)$ on the model $\mathbf{M}_{\boldsymbol{\zeta}}$ by at least a tolerance factor that depends on this dynamical modelling error, the policy's true returns $\eta(\pi)$ on the environment are guaranteed to improve ([27], see App. C for a detailed mathematical discussion). See Fig. 1 for an illustration of model-based RL. A good choice of the model function class, therefore, can impose strong and beneficial constraints on the space of possible predicted dynamics and thus lead to a smaller modelling error and returns' tolerance factor or allow the model to reduce the tolerance factor greatly after consuming an appropriate amount of training data.

Our choice of the model's functional form is motivated by the two ideas presented in the introduction: (a) incorporating correct partial knowledge about the physical system in the model ansatz parameters; (b) encoding the problem's symmetries and structure into model predictions as function space constraints. For the system in Eq. (1) we assume that the controls are partially characterized to address (a). Specifically, its time-dependent control structure H_c is known. We achieve (b) by parametrizing the system Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ with learnable parameters $\boldsymbol{\zeta}$, where L is the number of qubits. We make the model $\mathbf{M}_{\boldsymbol{\zeta}}$ a differentiable ODE whose generator is interpretable and has the form

$$\begin{aligned} H_{\boldsymbol{\zeta}}(\mathbf{u}(t), t) &= H_0^{(L)}(\boldsymbol{\zeta}) + H_c(\mathbf{u}(t), t) \\ &= \sum_{l=1}^{n^2} \zeta_l P_l + H_c(\mathbf{u}(t), t) \end{aligned} \tag{23}$$

where $\zeta_l = \mathrm{Tr}[P_l H_0(t)] \in [-1, 1]$ are real. Generally, like the Choi state, $H_0/\,\mathrm{Tr}[H_0]$ admits an arbitrary decomposition in terms of a basis $\{P_l\}_{l=1}^{n^2-1}$ of the SUn's Lie algebra. Analogously, for an open system, we parametrize the time-independent part of any dissipation dynamics in addition to the system Hamiltonian using an SU(n^2) algebra parametrization: $\mathbf{G}_0^{(L)}(\boldsymbol{\zeta}^{\text{diss}}) = \sum_l \zeta_l^{\text{diss}} P_l$ in the full generator $\mathbf{G}_{\boldsymbol{\zeta}}$.

The model is trained by minimizing the regression loss for single timestep predictions using data uniformly sampled, $D \sim \mathcal{D}$, where $\mathcal{D}$ represents the entire dataset,

$$L_{\text{model}}(D) = \sum_D \left(\mathbf{M}_{\boldsymbol{\zeta}}\left(\mathbf{s}_k, \mathbf{a}_k\right) - \mathbf{s}_{k+1}\right)^2. \tag{24}$$

To understand why a differentiable ODE ansatz is a good choice for the model, we need to define an ODE path that is given by $\phi_t : \mathbf{E}(0) \xrightarrow{H_{\boldsymbol{\zeta}}} \mathbf{E}(T)$ generated by $H_{\boldsymbol{\zeta}}$ for some time $t \in [0, T]$ and propagator $\mathbf{E}$. The ansatz is a good choice because of the following two properties of ODE paths: (a) they do not intersect and (b) if paths $\phi_0^{(A)}$,

$\phi_0^{(B)}$ start close compared to path $\phi_0^{(C)}$, then paths $\phi_t^{(A)}$, $\phi_t^{(B)}$ remain close compared to path $\phi_t^{(C)}$.

Both properties are well known [50, 51] for ODEs and become very useful when we try to predict the trajectories from noisy quantum data by imposing strong priors on the space of learnable Hamiltonians. Property (b) is a consequence of Gronwall's inequality [51] and essentially can be interpreted as: ODE flows that start off closer (w.r.t. the initial condition) stay closer (w.r.t. the final condition). Both (a) and (b) essentially imply a sort of intrinsic robustness of the ODE flow $\phi_t(\mathbf{z}_0)$ to perturbations on $\mathbf{z}_0$ [32]. They constrain the trajectories predicted by the model $\mathbf{M}_{\boldsymbol{\zeta}}$ to be intrinsically robust (over a finite time interval) to small noise in the states $\mathbf{s}_k$ and inaccuracies in the learned system Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$.

We call the SAC equipped with this differentiable ODE model the learnable Hamiltonian model-based SAC (LH-MBSAC) as listed in Algorithm 2. Crucially, LH-MBSAC generalizes the SAC by allowing the policy to interact with the ODE model and the physical system. LH-MBSAC gracefully falls back to the model-free SAC in the absence of a model with low prediction error that is measured from the performance of the model's predictions on an unseen validation set of interaction data. The threshold or tolerance level for switching to the agent-model interaction part of the algorithm is likely problem-dependent and thus needs to be selected along with other hyperparameters in RL. However, this allows us to improve the sample complexity of model-free reinforcement learning, when possible, by leveraging knowledge about the controllable quantum system, yet we are still able to control the system in a model-free manner if this is not possible.

IV. EXPERIMENTS

We demonstrate the performance of LH-MBSAC on three quantum systems of current interest in open and closed settings with shot noise. Measurements in this section are made using Pauli instead of the generalized Gell-Mann operators mentioned in Sec. II B and the simulated systems are all qubit systems.

To warm up, the first system $\tilde{H}_{\text{NV}}^{(1)}$ is a single-qubit NV center with microwave pulse control [52],

$$\frac{H_{\text{NV}}^{(1)}(t)}{\hbar} = 2\pi\Delta\sigma_z + \underbrace{2\pi\Omega\left(u_1(t)\sigma_x + u_2(t)\sigma_y\right)}_{H_c(t)}, \tag{25}$$

where $\Delta = 1$ MHz is the microwave frequency detuning, $\Omega = 1.4$ MHz is the Rabi frequency and the control field parameters are $u_j(t)$ in the range $\mathbb{X}_{\text{NV}}^{(1)} = \{-1 \leqslant u_j \leqslant 1\}$. In this and subsequent examples terms not covered by $H_c(t)$ are learned, parametrized by the learnable model parameters $\boldsymbol{\zeta}$. The gate operation time is 20 µs.

The second system $H_{\text{NV}}^{(2)}$ is a two-qubit NV center system [35], driven by microwave pulses of approximately 0.5 MHz, modelled as follows

$$\begin{aligned}\frac{H_{\text{NV}}^{(2)}(t)}{\hbar} &= |1\rangle\langle 1| \otimes \left(-\left(\nu_z + a_{zz}\right)\sigma_z - a_{zx}\sigma_x\right) \\ &\quad + |0\rangle\langle 0| \otimes \nu_z\sigma_z + \underbrace{\sum_{l=x,y}\sum_{k=1}^{2}\sigma_k^{(l)}u_{lk}(t)}_{H_c(t)},\end{aligned} \tag{26}$$

where $\nu_z = 0.158$ MHz, $a_{zz} = -0.152$ MHz and $a_{zx} = -0.11$ MHz, $\sigma_k^{(l)}$ is the lth Pauli operator on qubit k, and $u_{lk}(t)$ is a time-dependent control field. The range of control is $\mathbb{X}_{\text{NV}}^{(2)} = \{-1\,\text{MHz} \leqslant u_{lk} \leqslant 1\,\text{MHz}\}$ and the final gate time is $T = 2$ µs.

The third system $\tilde{H}_{\text{tra}}^{(L)}$ is an effective Hamiltonian model for cavity quantum electrodynamics (cQED) [36] for two transmons or qubits as a proxy for the IBM quantum circuits [53],

$$\begin{aligned}\frac{H_{\text{tra}}^{(2)}(t)}{\hbar} &= \sum_{l=1}^{2}\omega_l\hat{b}_l^\dagger\hat{b}_l + \frac{\eta_l}{2}\hat{b}_l^\dagger\hat{b}_l(\hat{b}_l^\dagger\hat{b}_l - \mathbb{1}) \\ &\quad + J\sum_{l=1}^{2}(\hat{b}_l^\dagger\hat{b}_{l+1} + \hat{b}_l\hat{b}_{l+1}^\dagger) + \underbrace{\sum_{l=1}^{2}u_l(t)(\hat{b}_l + \hat{b}_l^\dagger)}_{H_c(t)}.\end{aligned} \tag{27}$$

This model consists of Duffing oscillators with frequency $\omega_l = 5$ GHz representing the qubits with an anharmonicity $\eta_l = 0.2$ GHz, qubit coupling J, and a control field u_l per qubit. This is a special case of the Bose-Hubbard model [54] with $\hat{b}_l$ representing the boson annihilation operator on the jth qubit. The control field $u_l(t)$ is real by construction in addition to extra constraints imposed on the space of possible controls $\mathbb{X}$. The range of control is given by $\mathbb{X}_{\text{tra}}^{(2)} = \{-0.2\,\text{GHz} \leqslant u_l \leqslant 0.2\,\text{GHz}\}$ and the final gate time is $T = 20$ µs.

For the two-qubit system, the target gate is CNOT and for the one-qubit system, it is the Hadamard gate. Pulses are discretized in accordance with the scheme introduced in Sec. II C for the number of timesteps, $N = 20$. We follow the parameter restrictions for all systems introduced in Refs. [10, 35, 36, 52]. Moreover, due to limited support in our auto-differentiation library [55], we simulate the complex dynamics by mapping the complex ODE to two real coupled ODEs [56] (see App. A for more details on our ODE solver).

The following sections are organized as follows. In Sec. IV A, we demonstrate a sample complexity improvement for the different control problems discussed above in a noisy closed setting. For the subsequent sections, we study the two-qubit transmon control problem in more detail. The results were similar for other systems that we studied. In Sec. IV B, we study the effect of increasing the estimated Hamiltonian error from its true value on the sample complexity of control. Sec. IV C discusses how the learned Hamiltonian in LH-MBSAC can be further utilized for model-based control using gradient-based methods like GRAPE. Sec. IV D extends results from the

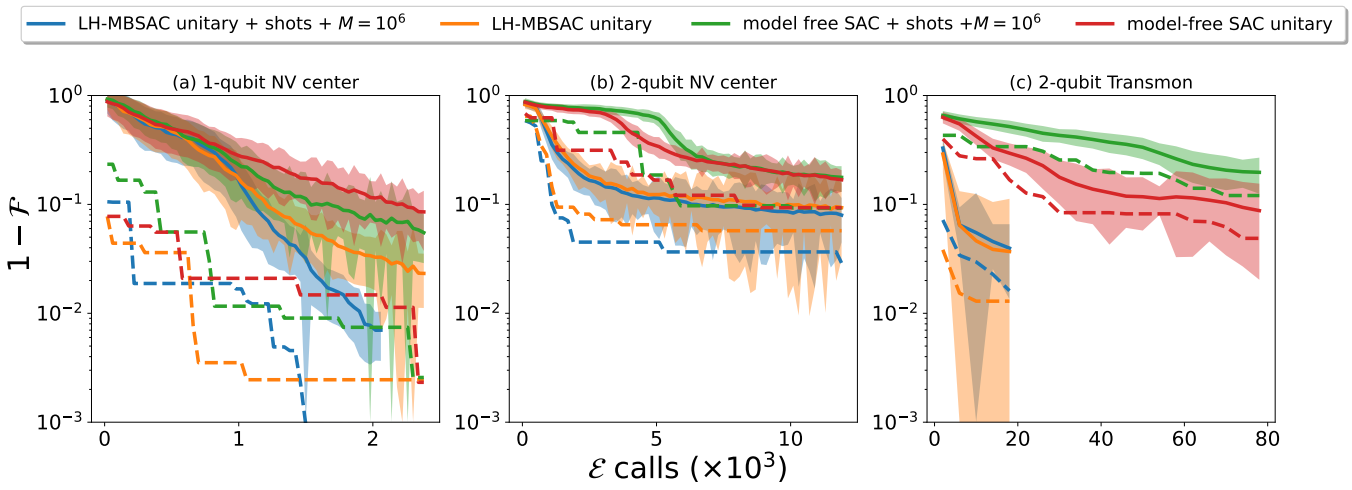


Figure 2. The closed system fidelity $\mathcal{F}$ of the Hadamard gate for (a) $H_{\text{NV}}^{(1)}$, and of the CNOT gate for (b) $H_{\text{NV}}^{(2)}$ and (c) $H_{\text{tra}}^{(2)}$ as a function of the number of environment $\mathcal{E}$ calls. The mean fidelity over 100 controllers is plotted as a solid line with the shading indicating two standard deviations, and the maximum fidelity is indicated by the dashed line. LH-MBSAC or model-free SAC with the unitary tag indicates the shot-noise-free closed system problem in Eq. (4) and single shot measurements are indicated likewise. We terminate the algorithm early at $\mathcal{F} > 0.98$ for LH-MBSAC with and without single shot measurements since the model simulations are expensive and the learned model at this point can be used to further optimize the moderately high fidelity RL pulses further as shown in Sec. IV C. The sample complexity of LH-MBSAC is significantly improved for the two-qubit transmon and the NV center over model-free SAC for the closed system control problem and with single shot measurements (of size $M = 10^6$), using AAPT. We average these results over three seeds of each algorithm run where a seed refers to a single algorithm run from scratch with a fresh set of randomly initialized parameters.

closed setting to the noisy open system setting. Finally, in Sec. IV E, we highlight some limitations and silver linings of the LH-MBSAC and the RL-for-control approach for our specific MDP (Eq. (17)) in this paper and provide promising ideas to circumvent some of the issues.

A. Sample Efficiency for Closed System Control

In this section, we only consider closed or unitary system control with and without single shot measurements defined in Sec. II A. From here on, we refer to single shot measurements as just "shots".

Unitary control (with closed system dynamics) is implemented for shots as a special case of open system control where the dissipation operator $\mathfrak{L}$ is 0. The Choi operator $\mathbf{\Phi}$ corresponding to the gate realized by the controls is obtained by sampling from the binomial distribution in Eq. (12) with $M = 10^6$ shots per measurement operator. By Hoeffding's inequality [57], we know that with probability $1 - 0.01$ the error in the estimator of q_l is 10^{-3}. Or generally, with probability $1 - \delta$, for ϵ error, we require $O(\log \frac{1}{\delta}/\epsilon^2)$ measurements. The AAPT method [45] (see Sec. II B) uses $M \times 3^L$ shots in total for 3^L possible measurement operators for an L-qubit system, which is quite expensive.

Further sparsity restrictions on the structure of $\mathbf{\Phi}$ imposed by a k-local Hamiltonian, where qubit interactions up to only the nearest $k \leqslant L$ qubits are assumed, can allow the shot cost to go down to $O(4^k(\log M)/\epsilon^2)$ for M observables due to a reduction in the number of observables that need to be measured or tracked which is asymptotically optimal in the number of measurements [58]. However, since the goal of this paper is gate control, these costs are generally unavoidable to completely verify gate performance. In practice, such gates are only limited to a few qubits and operations on many qubits are achieved in the circuit formalism through gate composition [53, 59].

We randomly initialize the learnable system Hamiltonian using the Pauli basis parametrization in Eq. (23) with coefficients $\zeta_i \sim \text{Uniform}(-1, 1)$. The environment's data buffer $D_{\mathcal{E}}$ that stores the model's training data, i.e., the initial exploration dataset (see Algorithm 2), consists of 1, 20, and 100 pulse sequences for the one-qubit NV, two-qubit NV and two-qubit transmon systems respectively. A more detailed discussion of the amount of training data needed for Hamiltonian learning is presented in Appendix D. These data are collected using random uniform policy actions during the first run of the LH-MBSAC algorithm.

The exploration dataset is then used to learn the system Hamiltonians $H_{0_{\text{NV}}}^{(1)}$, $H_{0_{\text{NV}}}^{(2)F}$, $H_{0_{\text{tra}}}^{(2)}$ via supervised learning of $\mathbf{M}_{\boldsymbol{\zeta}}$ using the dynamics prediction loss function (Eq. (24)) until a validation loss of around $10^{-3} \times 2^{2q} \times$ `batch_size` is reached, where `batch_size` is the number of samples used for a single training policy update. Here q is the number of qubits and $q = 2$ for the theoretical unitary and $q = 4$ for the Choi state (due to the Choi-Jamiolkowski isomorphism in AAPT).

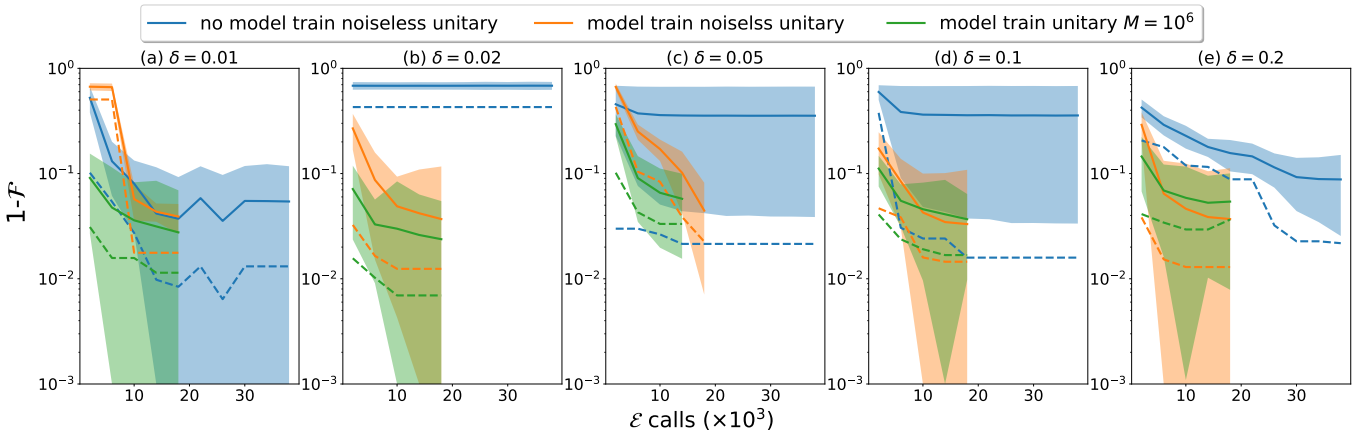


Figure 3. Sample complexity or $\mathcal{E}$ calls of LH-MBSAC for the two-qubit transmon control problem as a function of spectral norm error δ, quantifying closeness of the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ and the true system Hamiltonian H_0. The cases for $\delta = 0.01, 0.02, 0.05, 0.1, 0.2$ are plotted in (a)–(e). The mean fidelity over 100 controllers is plotted as a solid line with the shading indicating two standard deviations and the maximum fidelity is indicated by the dashed line. The 'noiseless unitary' is the no shot noise setting where the exact unitary is seen by the algorithm while alternatively the unitary is estimated using AAPT with $M = 10^6$ shots per observable characterizing the Choi state. The 'no model train' line indicates the setting where no learning of $H_0(\boldsymbol{\zeta})$ occurs and δ is fixed while the 'model train' lines denote the setting where δ is reduced through model training. In general, we see that there are some instances where the RL agent is able to optimize the objectively wrong model $\delta = 0.2, 0.01$ and there is a non-linear dependence of $\mathcal{E}$ calls on δ, i.e., a large δ can produce better model-predictive trajectories with a smaller unitary prediction error. This points us to consider the idea of learning Hamiltonians that are only 'locally consistent'. Once learning $H_0(\boldsymbol{\zeta})$ is enabled, algorithmic performance is restored in both the noiseless (with no shot noise) and shot-noise unitary settings. The number of measurements is $M = 10^6$ per observable.

After this, we switch to the model $\mathbf{M}_{\boldsymbol{\zeta}}$ to generate synthetic samples to train the policy π. Whilst concurrently maintaining policy interactions and attempting control of the system via the policy π, the model is successively trained in periods with fresh data to reduce the model error even further. Once the policy starts producing pulses with nearly optimal fidelities of around 0.98, we terminate the algorithm and use the learned Hamiltonian to further optimize the pulses using gradient-based methods like GRAPE to (a) reduce sample complexity costs and (b) improve runtime of LH-MBSAC, since the model simulations are computationally expensive. We found that terminating around 0.98 ensures that the application of further gradient-based methods doesn't cause the control parameters to diverge too much from their initial values thereby retaining, at least partially, their favourable robustness properties [12]. Step (b) is discussed in detail in Sec. IV C.

The results for LH-MBSAC and model-free SAC for the one- and two-qubit control problems are shown in Fig. 2. We consider LH-MBSAC's performance with shots by estimating the gate using its corresponding estimated Choi state $\boldsymbol{\Phi}$ using AAPT with 10^6 shots per observable. The sample complexity of LH-MBSAC to achieve a maximum fidelity significantly improves, by at least an order of magnitude, upon the model-free baseline in both cases, although it is more significant for the two-qubit transmon.

B. Sample Complexity as a Function of Hamiltonian Error

Continuing with the closed system control problem, in this section, we study the relationship between sample complexity and error in the estimated model Hamiltonian $H_0(\boldsymbol{\zeta})$ compared to the true system Hamiltonian H_0 as the error is increased. This relationship is highly nonlinear or irregular and is discussed in detail later in the section. On a high level, the purpose of this section is to understand the interplay between control and model learning especially if the model is inaccurate. Can we still learn a near optimal control policy even if the model is incorrect? To an extent, yes: we show that when the model error is small, LH-MBSAC is able to successfully find a near optimal control pulse, even with an incorrect model.

We define the model error δ as in Ref. [60]:

$$\delta = \|H_0(\boldsymbol{\zeta}) - H_0\| \tag{28}$$

where $\|\cdot\|$ is the spectral norm (the largest singular value) of $H_0(\boldsymbol{\zeta}) - H_0$. For this study, we compare two settings for some value of δ in each experimental run: (i) *learning the system Hamiltonian*, i.e., δ is decreased from its initial value; (ii) *not learning the system Hamiltonian*, i.e., δ remains fixed throughout the experiment. Case (ii) effectively corresponds to Algorithm 2 without any model training, i.e., we do not attempt to minimize $L_{\text{model}}(D_{\text{train}})$ to update the model and instead set the model to have a fixed constant Hamiltonian error δ. The

range of Hamiltonians corresponding to different δ values are chosen by randomly sampling the true Hamiltonian with rejection using Gaussian perturbations. The non-linear dependence on the sample complexity of LH-MBSAC as a function of δ for the two-qubit transmon control problem for both cases is shown in Fig. 3(a)–(e) for $\delta =\in \{0.01, 0.02, 0.05, 0.1, 0.2\}$.

For the two-qubit transmon problem, the $\delta = 0.02, 0.05, 0.1$ results show worse performance compared to the $\delta = 0.2$ results for the theoretical unitary control problem (without measurement noise). This indicates that some model system Hamiltonians $H_0(\boldsymbol{\zeta})$ with a larger δ predict dynamics more consistent with the true system Hamiltonian H_0 dynamics than $H_0(\boldsymbol{\zeta})$ with a smaller δ. However, learning $H^{(2)}_{0_{\text{tra}}}$ for all shown cases restores performance for both the noiseless unitary and shots-based closed system control problems.

To explain these empirical results and make them more intuitive, we now make use of the integration by parts lemma of Ref. [60] that bounds δ by the unitary prediction error of the ODE model w.r.t. the environment for the unitary control problem Eq. (4).

Proposition 1. *The following bound holds for the difference between the unitary model's predicted state $U_{\mathbf{M}_{\boldsymbol{\zeta}}}$ and the environment's unitary state $U_{\mathcal{E}}$,*

$$\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|_{\infty,t} \leqslant t^2\delta\left(\frac{1}{t} + \frac{2}{t}\|H_c\|_{1,t} + \|H_{\boldsymbol{\zeta}}\| + \|H_{\mathcal{E}}\|\right) \quad (29)$$

where $\|\cdot\|$ is the spectral norm and for some linear operator A, we have $\|A\|_{\infty,t} = \sup_{s\in[0,t]}\|A(s)\|$ and $\|A\|_{1,t} = \int_0^t ds\|A(s)\|$.

Proof. See proof of Prop. 2 in App. B. □

Proposition 1 hints at the intuition for why the Hamiltonian error is generally not linearly related to the propagator error.

Although there are some works with better relational bounds on the Hamiltonian error in terms of the observable error, these hinge on the ability to maintain a privileged basis and/or access to special probe states such as the Gibbs state basis [61, 62]. These bounds crucially do not include the propagator error, thanks to previous assumptions, which is a more general approach to bounding the quantum dynamical evolution error. Of course, there is always a price to be paid for generality and in this case, it is that the error bounds are less constrained and the link between the Hamiltonian and the unitary error becomes non-linear for the general case of the bound.

From Prop. 1, we infer that the unitary model prediction error or the supervised learning regression loss $L_{\text{model}}(D_{\text{train}})$ in Eq. (24) being small does not imply closeness between learned and true system Hamiltonian, i.e., $\delta \to 0$. However, in the converse case, δ being very small necessarily implies small propagator error. This is illustrated for the two-qubit transmon Hamiltonian in Fig. 4(a). The Hamiltonians are again sampled using Gaussian perturbations to the transmon Hamiltonian. There is also significant variation in the unitary model prediction error, even for the same value of δ for different repetitions of the random Hamiltonian. However, we see that with decreasing δ, the variation decreases, which is also explained by the above bound. Finally, the same pattern can also be observed if we take δ to be the mean squared difference between the Pauli coefficients of the true and learned Hamiltonian. Thus, this behaviour is general and not limited to the choice of δ.

The main takeaway of this section, that will be taken further in the next section, is that for the control problems considered here it is only necessary to learn models that are 'locally consistent' in terms of the unitary trajectories they generate, and small unitary prediction errors can be achieved by models with non-negligibly small δ.

C. Leveraging the Learned Hamiltonian with GRAPE

Proposition 1 paves the way to learning system Hamiltonians that are locally consistent with the unitary trajectories they generate. By local we mean that the learned Hamiltonian is consistent with the true Hamiltonian on only a subset of all possible generatable trajectories relevant to the control problem. In this section, we delve deeper into the learned model errors and also show that these local models can be leveraged to further optimize the fidelities of LH-MBSAC's controllers using gradient-based methods like GRAPE [7, 9].

During the model's $\mathbf{M}_{\boldsymbol{\zeta}}$ training phase, $H_0(\boldsymbol{\zeta})$ is made consistent with trajectories uniform randomly drawn from the data buffer $D_{\mathcal{E}}$ by minimizing the regression loss $L_{\text{model}}(D_{\mathcal{E}})$. This allows us to learn a model of the environment that can predict locally consistent unitary trajectories (i.e., at the scale of the control problem). In other words, the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ does not have to coincide with the true system Hamiltonian H_0 for it to be useful for the optimal control task. Indeed, we take the Hamiltonian learned for the two-qubit transmon in Fig. 2(c) and find that it has $\delta = 0.91509$. Diving deeper, the matrix difference between the true H_0 and learned Hamiltonian $H_0(\boldsymbol{\zeta})$ is,

$$H - H_0(\boldsymbol{\zeta}) = \begin{bmatrix} -0.912 & 0.001 & -0.001 & 0.001 \\ 0.001 & -0.914 & 0.001 - 0.001i & 0.001 + 0.001i \\ -0.001 & 0.001 + 0.001i & -0.913 & -0.001 \\ 0.001 & -0.001 - 0.001i & -0.001 & -0.914 \end{bmatrix}.$$

Notably, we can see that most of the error is actually in $\text{Tr}[H - H_0(\boldsymbol{\zeta})]$ with the true Hamiltonian being learned up to a scale factor of around 0.9 with the rest of the parameter error being small. This is precisely the global phase error that cannot be learned [63].

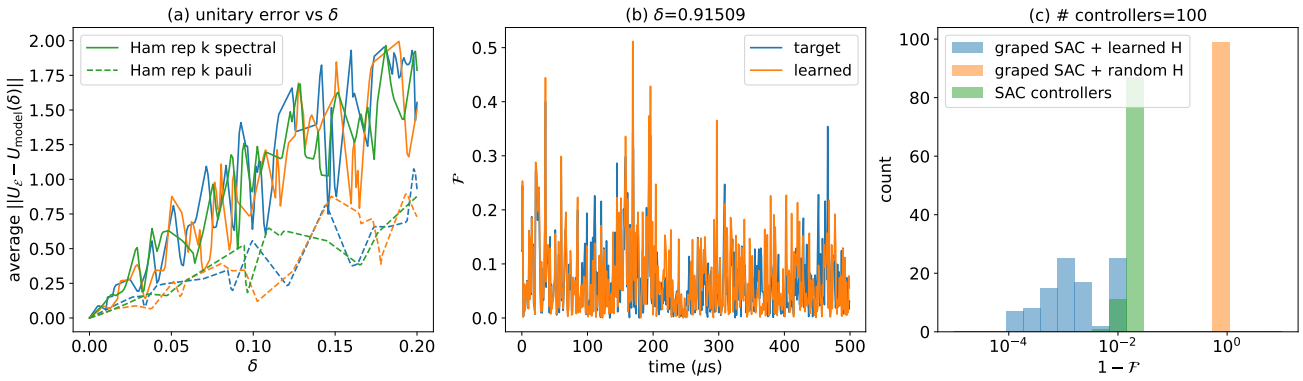


Figure 4. (a) An illustration of the non-linear relationship between the unitary model prediction error $\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|$ and Hamiltonian spectral norm (solid) error or mean squared Pauli basis difference (dashed) error as δ for the two-qubit transmon control problem. For the same 1000 random control pulses, we evaluate the average unitary prediction error of $\mathbf{M}_{\boldsymbol{\zeta}}$ with increasing δ for three different uniform randomly sampled two-qubit Hamiltonians $H_0(\boldsymbol{\zeta})$ to illustrate the variation in response to the unitary error. (b) Local and global unitary trajectories: $\mathcal{F}$ as a function of a random control pulse with either the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ or the true system Hamiltonian H_0. The learned $H_0(\boldsymbol{\zeta})$ trajectories do not coincide with the global trajectory with $\delta = 0.91509$, with the majority contribution coming from a global phase factor such that $\mathrm{Tr}[H - H_0(\boldsymbol{\zeta})] \approx 0.9$. Both trajectories start off extremely close and start diverging as time increases due to accumulation of small errors in the predicted dynamics. (c) The learned $H_0(\boldsymbol{\zeta})$ can be leveraged using GRAPE to further optimize the fidelities of LH-MBSAC's controllers. We plot a histogram of 100 LH-MBSAC controller infidelities $1 - \mathcal{F}$ before and after applying GRAPE on these controllers using the learned Hamiltonian and a random Hamiltonian. The LH-MBSAC fidelities are significantly improved after applying GRAPE. The appropriate baseline or benchmark representing our ignorance of H_0 is a random $H_0(\boldsymbol{\zeta})$ (with uniform random Pauli parameters) which, when plugged into GRAPE, yields extremely low fidelities near 0 towards the extreme right-hand side of the plot.

Despite this discrepancy between the true and learned system Hamiltonians, we find mostly good local agreement between the two random trajectories they induce thanks to the supervised training phase of the model. We show in Fig. 4(b) the local and global trajectories corresponding to $H_0(\boldsymbol{\zeta})$ and H_0 for the two-qubit transmon which shows that the two unitary trajectories w.r.t. the CNOT fidelity are not always coinciding. More specifically, we can see a high overlap in the fidelities induced by random pulses for times between 0 µs to around 100 µs. Moreover, the small differences in the generator only start manifesting as the time scales get longer and this can be explained by accruing of small errors in predicted dynamics. This confirms that the unitary model prediction error grows as a function of time. This makes intuitive sense since predictions far into the future, compared to their time-wise preceding counterparts, must necessarily have more built-up error. Furthermore, this learned 'local' $H_0(\boldsymbol{\zeta})$ and the controllers found by LH-MBSAC can be used in conjunction with the model-based GRAPE control algorithm [7, 9] to optimize the SAC controller fidelities much more quickly than via just RL alone using accelerated second-order gradient descent. The LH-MBSAC controllers act as seeds, so GRAPE does not move too far away in pulse parameter space compared to where it started. Although not done here, this can also be imposed as an explicit constraint. Note that the question of exactly when to switch over to GRAPE beyond heuristics remains unanswered.

The fidelities after applying GRAPE are evaluated w.r.t. the true system Hamiltonian H_0. Usually LH-MBSAC controllers have moderately high fidelities around $\mathcal{F} > 0.98$ which are improved to $\mathcal{F} > 0.999$. In Fig. 4(c), we show the RL controllers being optimized further using the learned $H_0(\boldsymbol{\zeta})$ with GRAPE. Experiments in this section for the two-qubit NV center system yield similar results and can be found in App. E.

D. Open System Control with Single Shot Measurements

Due to the interpretable nature of our ODE model's ansatz in Eq. (23), it is pertinent to ask if two competing but linear terms in the model $\mathbf{M}_{\boldsymbol{\zeta}}$ can be learned simultaneously. In this section, we find that for our model learning setting, the answer to this question is no. However, this is not general to all problem settings and could potentially be pursued in future work.

In the previous sections, we only learn one term represented by $H_0(\boldsymbol{\zeta})$. Utilizing the open system formulation of the control problem in Sec. II B, we consider Lindblad dissipation along with shot noise for the two-qubit transmon control problem in Eq. (14). Specifically, we consider the decoherence operator $\mathfrak{L}_{\text{diss}}^{(l)} = \sqrt{\frac{2}{R_l^*}} b_l b_l^\dagger$, acting on the lth qubit, and the decay operator $\mathfrak{L}_{\text{decay}}^{(l)} = \sqrt{\frac{2}{R_l}} b_l$ for $l = 1, 2$. R_l^* and R_l are the decoherence and decay rates. Both operators are time-independent

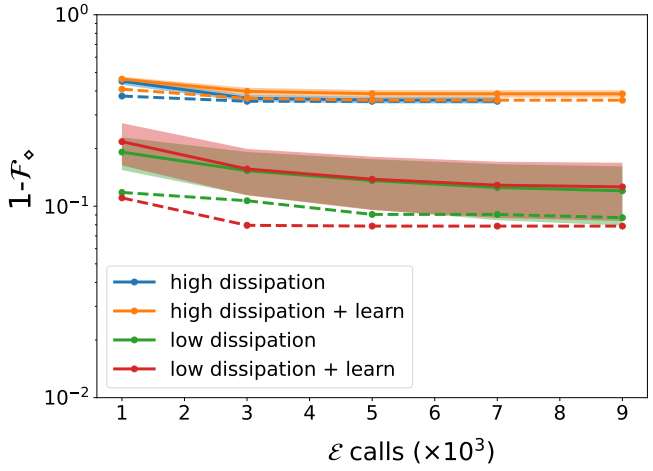


Figure 5. Diamond norm fidelity $\mathcal{F}_\diamond$ for the two-qubit transmon control problem in low and high Lindblad dissipation regimes for LH-MBSAC. The results are averaged over two seeds with the mean $\mathcal{F}_\diamond$ over 100 controllers shown in solid and the maximum $\mathcal{F}_\diamond$ in dashed lines. Shading denotes two standard deviations from the mean. Here, the 'learn' label signifies that dissipation operators are being learned in addition to the system Hamiltonian.

Alternatively, we can also represent these operators using the adjoint representation but we note that in the context of this learning problem that representation will not make much difference as our algorithm is able to effectively learn the Hamiltonian up to addition of a scalar matrix. However, practically speaking, one can obtain the energy differences of the Hamiltonian via spectroscopy [64] which can then be encoded in the eigenvalues of the adjoint representation. It is also possible to learn these eigenvalues using measurements of canonical (Gibbs) states [61].

We perform experiments for high and low dissipation corresponding to the gate times $R_l^{*^{\text{hi}}} = R_l^{\text{hi}} = 4\ \mu$s, and $R_l^{*^{\text{lo}}} = R_l^{\text{lo}} = 20\,\mu$s. Comprising both of these time-independent operators, the Lindblad term $\mathbf{L}_1$ is learned concomitantly with the system Hamiltonian. The results are shown in Fig. 5 where the "learn" label signifies that $\mathbf{L}_1$ is being learned in addition to the system Hamiltonian $H_0(\boldsymbol{\zeta})$.

We use the diamond norm fidelity [65] $\mathcal{F}_\diamond$,

$$\mathcal{F}_\diamond(\mathbf{\Phi}(\mathbf{u}(t),t),\mathbf{\Phi}_{\text{target}}) = 1 - \|\mathbf{\Phi}(\mathbf{u}(t),t) - \mathbf{\Phi}_{\text{target}}\|_\diamond, \tag{30}$$

instead of the generalised state fidelity since the latter lacks the sensitivity to detect the low dissipation regime (see App. G). We find that attempting to learn $\mathbf{L}_1$ while learning $H_0(\boldsymbol{\zeta})$ confers little to no advantage in both the high and low dissipation regimes for this control task. Further investigation shows that the estimate of the system Hamiltonian $H_0(\boldsymbol{\zeta})$ compensates for the observed discrepancy in observed dynamics due to dissipation as much as it is unitarily possible. Moreover, the learning processes for $\mathbf{L}_1$ and $H_0(\boldsymbol{\zeta})$ become entangled/mixed so learning multiple independent terms in $\mathbf{M}_{\boldsymbol{\zeta}}$ may not be suitable for LH-MBSAC.

E. Limitations and Silver Linings

There are two major limitations of LH-MBSAC. The first is that only the system or time-independent part of the Hamiltonian can be learned with the algorithm, while the more difficult problem of learning the time-dependent part of the Hamiltonian [63] is left as future work.

Moreover, we found that LH-MBSAC was not able to tackle a three-qubit transmon control problem to obtain a Toffoli gate on an extension of the transmon system. The limitation applied mostly to the RL agent; a viable Hamiltonian is learned that can be leveraged with GRAPE as before. Specific computational details are discussed in App. F. Essentially, our findings indicate this is an optimization landscape problem and an issue specific to the meta RL strategy of finding optimal pulses instead of a hyperparameter problem. There are two major reasons behind this assessment. Firstly, the values and the gradients for policy and value functions saturate with large training times, i.e., both are stuck in suboptimal extrema, which ultimately culminate with a prematurely optimized reward function. Secondly, since the model Hamiltonian is known beforehand (or also learned), GRAPE equipped with this Hamiltonian and initialized with the highest fidelity LH-MBSAC controllers also gets stuck.

However, the LH-MBSAC strategy is not limited to SAC and can augment different RL algorithms for which the three-qubit problem may be tractable. Also, since this is likely an optimization landscape issue, a reformulation of the RL control problem could also alleviate this issue by reducing the probability of SAC getting stuck by increasing the range of fidelities the RL agent sees as 'proximally optimal'. At present, the agent's goal is to maximize all fidelities it observes, with most of the observations being premature, i.e., before the final gate time. This is highlighted in Fig. 6 which shows the infidelity $1-\mathcal{F}$ as a function of time for 100 pulses found by LH-MBSAC and GRAPE for the two-qubit transmon control problem. Compared to GRAPE, LH-MBSAC pulses are much more consistent and periodic in terms of the intermediate fidelity values. This highlights that the RL approach is biased towards optimizing intermediate fidelities along with the final target fidelity (since the objective function in Eq. (19) is the regularized expected cumulative fidelity). This is quite different from the approach taken by the gradient-based GRAPE algorithm. Despite being interesting from a controller robustness point of view [12], this bias can prevent solutions that do not admit high intermediate fidelities from being found as RL can get stuck in a loop mining medium-level fidelity values. Stepping away from this particular sequential decision-making MDP formulation might be one solution to consider in future work.

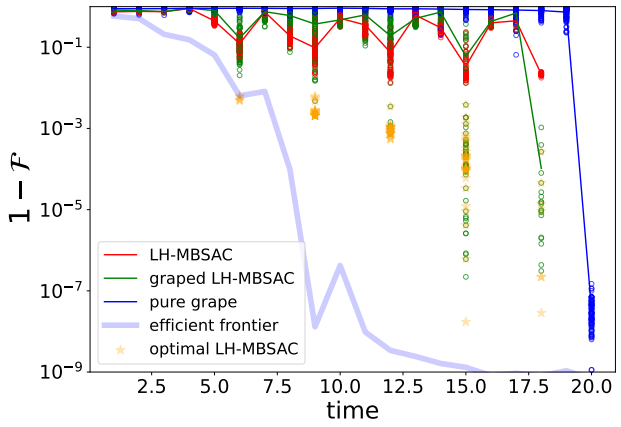


Figure 6. The infidelities over time for 100 different control pulses found by LH-MBSAC and by GRAPE using the learned system Hamiltonian $H_0(\boldsymbol{\zeta})$ for the two-qubit transmon control problem with final time $T \leqslant 20$ μs. RL pulses are further optimized using GRAPE. GRAPE is also used to obtain pulses without the RL controls as initial values for a fixed final gate time $T = 20$ μs. Short optimal controls found by RL are identified by truncating RL pulse parameters at times $t \geqslant \{6, 9\}$ μs whose final infidelities are shown as stars with $t = 6$ μs being Pareto optimal w.r.t. the efficient frontier (the surface indicating the best fidelity for that time).

There are silver linings for the aforementioned MDP formulation. RL pulses are fidelity-wise better, on average, across the duration of the pulse. Leveraging the learned system Hamiltonian, we can further improve the performance of the RL pulses by using GRAPE with the RL pulse parameters as initialization. As seen in Fig. 6, these pulses are still better than the ones found by GRAPE using the learned system Hamiltonian but with completely random pulse initializations, i.e., without LH-MBSAC controllers as seeds.

Furthermore, this RL bias towards valuing intermediate fidelities allows us to identify optimal pulses that can be executed in short times, which is a difficult problem for GRAPE even if the final gate time is explicitly added to the control objective [9].

Truncating the control sequence for pulses at time t if the infidelity is below 5×10^{-2}, we again leverage GRAPE to maximize the final fidelities at these shorter times. These are shown as stars in Fig. 6 with the fidelities at $t = 6$ μs being approximately Pareto optimal, i.e., the best fidelity for that time. The Pareto optimal efficient frontier is constructed by sampling 100 GRAPE pulses with random intializations at different final gate times.

V. CONCLUSION

We have presented a learnable Hamiltonian soft actor-critic (LH-MBSAC) algorithm for time-dependent noisy quantum gate control. LH-MBSAC augments model-free soft-actor critic by allowing the reinforcement learning (RL) policy to query a learnable model of the environment or the controllable system. It thereby reduces the total number of queries (sample complexity) required to solve the RL task. The model is a differentiable ODE with a partially characterized Hamiltonian, where only the parametrized time-independent system Hamiltonian is required to be learned. This is a good inductive bias for the quantum control task as ODE trajectories do not intersect, and the Schrödinger ordinary differential equation (ODE) preserves unitary evolution, thereby sensibly constraining the space of models to be learned. Using exploration data acquired from the policy during the RL loop, we train the model by reducing a model prediction error over the data. We show that LH-MBSAC is able to reduce the sample complexity for gate control of one- and two-qubit nitrogen-vacancy (NV) centers and transmon systems in unitary and single-shot measurement settings.

Moreover, we highlight that despite the generally nonlinear relationship between the error in the learned Hamiltonian and the model prediction error, LH-MBSAC's performance is robust to this variation. Furthermore, even if the learned Hamiltonian that minimizes the model prediction error is not the same as the true system Hamiltonian, the learned Hamiltonian which is locally consistent in terms of its dynamical predictions can be leveraged using gradient-based methods that require full knowledge of the controllable system, like GRAPE, to further optimize the controllers found by LH-MBSAC. Applying LH-MBSAC in high and low Lindblad dissipation regimes with shot noise, we found that its performance in both was not improved if the Lindblad dissipation terms are also learned in addition to the system Hamiltonian as it is likely that the latter part compensates for the extra dissipation effects.

Despite LH-MBSAC's limitations requiring it to know the time-dependent Hamiltonian and system scalability beyond two qubits (four with single shot measurements due to ancilla assisted process tomography (AAPT)), the algorithm can be used to augment many existing model-free RL approaches for quantum control. This should afford more sample-efficient RL-based optimization of quantum dynamics for near-term noisy quantum processors on a variety of architectures as shown in the paper. Specific tasks can include noisy small circuit optimization, state preparation [14, 15] or gate optimization using a partially known model of the underlying dynamics [13]. Since having an accurate model can be extremely useful for validation of quantum operations and model bias can be crippling, model-based RL methods like LH-MBSAC can improve the model specifically tailored for some downstream task, e.g., quality assessment of topological codes [66] or fine-tuning current implementations of a two-qubit cross resonance gate on some novel architecture [24] using a pre-existing but partially correct model. Here, the goal for the RL agent would be to help learn effective and potentially scalable models of the target system whilst optimizing the target functional. Another interesting goal in this direction could just be in-

corporating the number of measurements or queries of the system in the RL objective so that the learning is sample-efficient. Another avenue of future work is to combine LH-MBSAC with a more feasible measurement protocol than AAPT. AAPT is not a hard requirement for our approach and was used here for its theoretically simple estimation of a quantum process. Two angles of attack are either sparsity assumptions on the dynamics generator [67] and the generated evolution [58] or a partially observed Markov Decision Process formulation of the control problem [6, 68].

Moreover, despite the scalability problems due to the potentially hindering nature of the RL strategy towards maximizing intermediate fidelities, it can be useful in particular to identify short time optimal pulses. Learning the time-dependent part of the Hamiltonian is harder and might require a stronger learning protocol, e.g., using the zero-order hold method with the learning protocol presented in this paper, Bayesian Hamiltonian Learning [63] or more informative learning process or Hamiltonian learning methods [67, 69] which would be exciting to pursue in the future.

The study of the abilities and limitations of our Hamiltonian learning protocol using ZOH will be left to future work. Our code is available at [70].

[1] C. P. Koch, U. Boscain, T. Calarco, G. Dirr, S. Filipp, S. J. Glaser, R. Kosloff, S. Montangero, T. Schulte-Herbrüggen, D. Sugny, and F. K. Wilhelm, Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in europe, EPJ Quantum Technol. **9**, 19 (2022).

[2] 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).

[3] D. Gottesman, Quantum fault tolerance in small experiments (2016), arXiv:1610.03507.

[4] R. Harper and S. T. Flammia, Fault-tolerant logical gates in the IBM quantum experience, Phys. Rev. Lett. **122**, 080504 (2019).

[5] J. M. Chow, J. M. Gambetta, E. Magesan, D. W. Abraham, A. W. Cross, B. R. Johnson, N. A. Masluk, C. A. Ryan, J. A. Smolin, S. J. Srinivasan, *et al.*, Implementing a strand of a scalable fault-tolerant quantum computing fabric, Nature Communications **5**, 1 (2014).

[6] I. Khalid, C. Weidner, E. A. Jonckheere, S. G. Schirmer, and F. C. Langbein, Reinforcement learning vs. gradient-based optimisation for robust energy landscape control of spin-1/2 quantum networks, in *60th IEEE Conference on Decision and Control (CDC)* (IEEE, 2021) pp. 4133–4139.

[7] 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, Journal of Magnetic Resonance **172**, 296 (2005).

[8] D. M. Reich, M. Ndong, and C. P. Koch, Monotonically convergent optimization in quantum control using krotov's method, The Journal of Chemical Physics **136**, 104103 (2012).

[9] S. Machnes, U. Sander, S. J. Glaser, P. de Fouquières, A. Gruslys, S. Schirmer, and T. Schulte-Herbrüggen, Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework, Phys. Rev. A **84**, 022305 (2011).

[10] N. Wittler, F. Roy, K. Pack, M. Werninghaus, A. S. Roy, D. J. Egger, S. Filipp, F. K. Wilhelm, and S. Machnes, Integrated tool set for control, calibration, and characterization of quantum devices applied to superconducting qubits, Phys. Rev. Applied **15**, 034080 (2021).

[11] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, Universal quantum control through deep reinforcement learning, npj Quantum Information **5**, 1 (2019).

[12] I. Khalid, C. A. Weidner, E. A. Jonckheere, S. G. Shermer, and F. C. Langbein, Statistically characterizing robustness and fidelity of quantum controls and quantum control algorithms, Phys. Rev. A **107**, 032606 (2023).

[13] M. Dalgaard, F. Motzoi, J. J. Sørensen, and J. Sherson, Global optimization of quantum dynamics with alphazero deep exploration, npj Quantum Information **6**, 1 (2020).

[14] V. V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. H. Devoret, Model-free quantum control with reinforcement learning, Phys. Rev. X **12**, 011059 (2022).

[15] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, Phys. Rev. X **8**, 031086 (2018).

[16] X.-d. Yang, C. Arenz, I. Pelczer, Q.-M. Chen, R.-B. Wu, X. Peng, and H. Rabitz, Assessing three closed-loop learning algorithms by searching for high-quality quantum control pulses, Phys. Rev. A **102**, 062605 (2020).

[17] F. Schäfer, M. Kloc, C. Bruder, and N. Lörch, A differentiable programming method for quantum control, Machine Learning: Science and Technology **1**, 035009 (2020).

[18] F. Schäfer, P. Sekatski, M. Koppenhöfer, C. Bruder, and M. Kloc, Control of stochastic quantum dynamics by differentiable programming, Machine Learning: Science and Technology **2**, 035004 (2021).

[19] M. H. Goerz, S. C. Carrasco, and V. S. Malinovsky, Quantum optimal control via semi-automatic differentiation, Quantum **6**, 871 (2022).

[20] 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).

[21] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction* (MIT press, 2018).

[22] 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).

[23] V. Sivak, A. Eickbusch, B. Royer, S. Singh, I. Tsioutsios,

S. Ganjam, A. Miano, B. Brock, A. Ding, L. Frunzio, *et al.*, Real-time quantum error correction beyond break-even, Nature **616**, 50 (2023).

[24] L. Ding, M. Hays, Y. Sung, B. Kannan, J. An, A. D. Paolo, A. H. Karamlou, T. M. Hazard, K. Azar, D. K. Kim, B. M. Niedzielski, A. Melville, M. E. Schwartz, J. L. Yoder, T. P. Orlando, S. Gustavsson, J. A. Grover, K. Serniak, and W. D. Oliver, High-fidelity, frequency-flexible two-qubit fluxonium gates with a transmon coupler (2023), arXiv:2304.06087.

[25] R. S. Sutton, Dyna, an integrated architecture for learning, planning, and reacting, ACM SIGART Bulletin **2**, 160 (1991).

[26] K. Chua, R. Calandra, R. McAllister, and S. Levine, Deep reinforcement learning in a handful of trials using probabilistic dynamics models, in *Advances in Neural Information Processing Systems*, Vol. 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018).

[27] M. Janner, J. Fu, M. Zhang, and S. Levine, When to trust your model: Model-based policy optimization, in *Advances in Neural Information Processing Systems*, Vol. 32 (Curran Associates, Inc., 2019).

[28] H. P. Van Hasselt, M. Hessel, and J. Aslanides, When to use parametric models in reinforcement learning?, in *Advances in Neural Information Processing Systems*, Vol. 32 (Curran Associates, Inc., 2019).

[29] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, Neural ordinary differential equations, in *Advances in Neural Information Processing Systems*, Vol. 31, edited by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Curran Associates, Inc., 2018).

[30] E. A. Coddington and N. Levinson, *Theory of ordinary differential equations* (Tata McGraw-Hill Education, 1955).

[31] E. Dupont, A. Doucet, and Y. W. Teh, Augmented neural ODEs, in *Advances in Neural Information Processing Systems*, Vol. 32, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019).

[32] H. Yan, J. Du, V. Y. F. Tan, and J. Feng, On robustness of neural ordinary differential equations (2022), arXiv:1910.05513.

[33] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges (2021), arXiv:2104.13478.

[34] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *International Conference on Machine Learning* (PMLR, 2018) pp. 1861–1870.

[35] S. S. Hegde, J. Zhang, and D. Suter, Efficient quantum gates for individual nuclear spin qubits by indirect control, Phys. Rev. Lett. **124**, 220501 (2020).

[36] E. Magesan and J. M. Gambetta, Effective Hamiltonian models of the cross-resonance gate, Phys. Rev. A **101**, 052308 (2020).

[37] M. Clouâtré, M. J. Khojasteh, and M. Z. Win, Model-predictive quantum control via Hamiltonian learning, Trans. Quantum Eng. **3**, 1 (2022).

[38] A. Dutkiewicz, T. E. O'Brien, and T. Schuster, The advantage of quantum control in many-body Hamiltonian learning (2023), arXiv:2304.07172.

[39] H.-P. Breuer, F. Petruccione, *et al.*, *The theory of open quantum systems* (Oxford University Press on Demand, 2002).

[40] F. F. Floether, P. De Fouquieres, and S. G. Schirmer, Robust quantum gates for open systems via optimal control: Markovian versus non-Markovian dynamics, New Journal of Physics **14**, 073023 (2012).

[41] C. J. Wood, J. D. Biamonte, and D. G. Cory, Tensor networks and graphical calculus for open quantum systems (2015), arXiv:1111.6950.

[42] A. Lichnerowicz, *Elements of tensor calculus* (Courier Dover Publications, 2016).

[43] M.-D. Choi, Completely positive linear maps on complex matrices, Linear Algebra and its Applications **10**, 285 (1975).

[44] A. Jamiolkowski, Linear transformations which preserve trace and positive semidefiniteness of operators, Reports on Mathematical Physics **3**, 275 (1972).

[45] J. B. Altepeter, D. Branning, E. Jeffrey, T. C. Wei, P. G. Kwiat, R. T. Thew, J. L. O'Brien, M. A. Nielsen, and A. G. White, Ancilla-assisted quantum process tomography, Phys. Rev. Lett. **90**, 193601 (2003).

[46] R. A. Bertlmann and P. Krammer, Bloch vectors for qudits, Journal of Physics A: Mathematical and Theoretical **41**, 235303 (2008).

[47] F. Sauvage and F. Mintert, Optimal quantum control with poor statistics, PRX Quantum **1**, 020322 (2020).

[48] S. T. Flammia and Y.-K. Liu, Direct fidelity estimation from few pauli measurements, Phys. Rev. Lett. **106**, 230501 (2011).

[49] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey, *et al.*, Maximum entropy inverse reinforcement learning, in *Aaai*, Vol. 8 (Chicago, IL, USA, 2008) pp. 1433–1438.

[50] L. Younes, *Shapes and diffeomorphisms*, Applied Mathematical Sciences, Vol. 171 (Springer, 2010).

[51] R. Howard, The Gronwall inequality, lecture notes (1998).

[52] F. Frank, T. Unden, J. Zoller, R. S. Said, T. Calarco, S. Montangero, B. Naydenov, and F. Jelezko, Autonomous calibration of single spin qubit operations, npj Quantum Information **3**, 1 (2017).

[53] A. Cross, The IBM Q experience and QISKit open-source quantum computing software, in *APS March meeting abstracts*, Vol. 2018 (2018) pp. L58–003.

[54] T. D. Kühner, S. R. White, and H. Monien, One-dimensional bose-hubbard model with nearest-neighbor interaction, Phys. Rev. B **61**, 12474 (2000).

[55] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, Pytorch: An imperative style, high-performance deep learning library, in *Advances in Neural Information Processing Systems*, Vol. 32 (2019).

[56] 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).

[57] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning* (MIT press, 2018).

[58] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics **16**, 1050 (2020).

[59] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information* (Cambridge University Press, 2010).

[60] D. Burgarth, P. Facchi, G. Gramegna, and K. Yuasa, One bound to rule them all: from adiabatic to zeno, Quantum **6**, 737 (2022).

[61] A. Anshu, S. Arunachalam, T. Kuwahara, and M. Soleimanifar, Sample-efficient learning of interacting quantum systems, Nature Physics **17**, 931 (2021).

[62] J. Haah, R. Kothari, and E. Tang, Optimal learning of quantum Hamiltonians from high-temperature Gibbs states, in *IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (2022) pp. 135–146.

[63] T. J. Evans, R. Harper, and S. T. Flammia, Scalable Bayesian Hamiltonian learning (2019), arXiv:1912.07636.

[64] A. Izmalkov, S. H. W. van der Ploeg, S. N. Shevchenko, M. Grajcar, E. Il'ichev, U. Hübner, A. N. Omelyanchouk, and H.-G. Meyer, Consistency of ground state and spectroscopic measurements on flux qubits, Phys. Rev. Lett. **101**, 017003 (2008).

[65] G. Benenti and G. Strini, Computing the distance between quantum channels: usefulness of the fano representation, Journal of Physics B: Atomic, Molecular and Optical Physics **43**, 215508 (2010).

[66] A. Valenti, E. van Nieuwenburg, S. Huber, and E. Greplova, Hamiltonian learning for quantum error correction, Phys. Rev. Res. **1**, 033092 (2019).

[67] H.-Y. Huang, Y. Tong, D. Fang, and Y. Su, Learning many-body Hamiltonians with Heisenberg-limited scaling, Phys. Rev. Lett. **130**, 200403 (2023).

[68] M. Hausknecht and P. Stone, Deep recurrent Q-learning for partially observable MDPs, in *AAAI Fall Symposium Series* (2017) arXiv:1507.06527.

[69] H.-Y. Huang, S. Chen, and J. Preskill, Learning to predict arbitrary quantum processes (2023), arXiv:2210.14894.

[70] https://github.com/erg0dic/transmon_public (2023).

[71] E. Süli and D. F. Mayers, *An introduction to numerical analysis* (Cambridge University Press, 2003).

[72] 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, Nature Methods **17**, 261 (2020).

[73] S. Sheldon, E. Magesan, J. M. Chow, and J. M. Gambetta, Procedure for systematically tuning up cross-talk in the cross-resonance gate, Phys. Rev. A **93**, 060302 (2016).

[74] A. Uhlmann, Fidelity and concurrence of conjugated states, Phys. Rev. A **62**, 032307 (2000).

[75] J. Watrous, Semidefinite programs for completely bounded norms (2009), arXiv:0901.4709.

Appendices

Here we present additional details and proofs for the results in the main text.

Appendix A: Mapping Complex Linear ODEs to Coupled Real ODEs and Step-size Effects

The quantum control problem in Eqs. (4) and Eq. (14) involve ODEs (Eqs. (2), (9)) in the complex domain with a complex vector field map $f_\theta : \mathbb{R} \times \mathbb{C}^d \to \mathbb{C}^d$ (where θ denotes some learnable parameters that can be optimized). For the unitary control problem we have a linear map $f_\theta(U(\mathbf{u}(t),t),t) = H_\theta(\mathbf{u}(t),t)U(\mathbf{u}(t),t)$ where H_θ is a Hermitian Hamiltonian that generates the ODE path of the propagator $U(t)$. We make use of the following isomorphism to map the complex ODE to two coupled real ODEs in $\mathbb{R}^{2d}$ by separating the propagator into its real and imaginary parts $U = U_{\text{real}} + iU_{\text{imag}}$ and mapping the Hamiltonian isomorphically $H(\mathbf{u}(t),t) \xrightarrow{\sim} \mathbb{1} \otimes H_{\text{real}}(\mathbf{u}(t),t) - i\sigma_y \otimes H_{\text{imag}}(\mathbf{u}(t),t)$, to get the following [56] coupled real ODE system,

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} U_{\text{real}}(\mathbf{u}(t),t) \\ U_{\text{imag}}(\mathbf{u}(t),t) \end{pmatrix} = \begin{pmatrix} H_{\text{imag}}(\mathbf{u}(t),t) & H_{\text{real}}(\mathbf{u}(t),t) \\ -H_{\text{real}}(\mathbf{u}(t),t) & H_{\text{imag}}(\mathbf{u}(t),t) \end{pmatrix} \begin{pmatrix} U_{\text{real}}(\mathbf{u}(t),t) \\ U_{\text{imag}}(\mathbf{u}(t),t) \end{pmatrix}. \tag{A1}$$

The mapping is analogous for the superoperator ODE in Eq. (9). Likewise, various other metrics, e.g., fidelity $\mathcal{F}$, were analogously transformed. We made use of the real nature of the Pauli vector decomposition of H to keep track of both the time-independent learnable Hamiltonian and the time-dependent control Hamiltonian representations.

We use Heun's method [71] to implement a custom differentiable numerical ODE solver in `pytorch` [55], a popular automatic differentiation code library. The solver is able to evolve multiple ODEs under multiple generators in parallel using generalized matrix/tensor operations (ideally on a GPU to maximally leverage computational efficiency). The solver can be accessed in the `LearnableHamiltonian` module in our code [70]. To determine the optimal tradeoff between accuracy of dynamical simulation, computed gradients and the size of the computation graph that is held in memory for automatic differentiation, we conduct experiments by simulating the dynamics of random n-qubit Hamiltonians from $n = 1$ to $n = 4$ at different precision or tolerance or step size of the ODE solver (see Fig. 7).

Computational speed of the solver naturally trades off with the accuracy in the simulation and the computed gradients. We find that a step size of 10^{-2} is sufficiently accurate for forward dynamical simulation (no gradients are computed in this step) and a step size of 5×10^{-4} is required for the backward step when the gradients need to be computed to train the ODE model. The errors in the dynamical predictions (averaged over many thousands of data points) in both steps are reasonably small and monitored. The ODE solvers in `scipy` [72] and the matrix exponential method for solving linear ODEs [9] both have similar errors than our method for the step size 5×10^{-4} (likely the Bayes' optimal error for our numerical simulation).

The ability to be fast, but produce slightly less accurate predictions improved the wall time of our algorithm. Specifically, a significantly large number of trajectories can be quickly sampled in the forward step to augment the RL policy's training data while the much slower backward step can be limited to a smaller number of trajectories that need to be predicted and are divided over multiple batches.

Appendix B: Bounds on the Model Prediction Error

Consider a unitary RL control problem with the MDP in Eq. (17), where the environment's Hamiltonian and propagator at some timestep t_l are given by $H_\mathcal{E}(t_l, u_l) = H_0 + H_c(u_l, t_l)$ and $U_\mathcal{E}(\mathbf{u}_k)$. Now consider the model $\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s}_{k+1} \,|\, \mathbf{a}_k, \mathbf{s}_k)$ that predicts a single step of unitary dynamics $\mathbf{s}_k \xrightarrow{H_{\boldsymbol{\zeta}}} \mathbf{s}_{k+1}$ under its parametrized generator $H_{\boldsymbol{\zeta}} = H_0^{(L)}(\boldsymbol{\zeta}) + H_c(u_l, t_l)$ following our assumptions in Sec. III. Now we bound the error in the single step predicted propagator $U_{\boldsymbol{\zeta}}$ using the integration-by-parts lemma from Ref. [60]. We consider a continuous version of the propagators and the generators since the result is only used qualitatively.

Proposition 2. *(Bound on the model predictions) The following bound between the unitary model's predicted state $U_{\boldsymbol{\zeta}}(\mathbf{u}_{:k})$ and the environment's unitary state $U_\mathcal{E}(\mathbf{u}_k)$ holds,*

$$\left\|U_\mathcal{E} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|_{\infty,t} \leqslant t^2 \left\|H_0^{(L)}(\boldsymbol{\zeta}) - H_0\right\| \cdot \left(\frac{1}{t} + \frac{2}{t}\|H_c\|_{1,t} + \|H_{\boldsymbol{\zeta}}\| + \|H_\mathcal{E}\|\right). \tag{B1}$$

Proof: The generator difference $H_{\boldsymbol{\zeta}} - H_\mathcal{E} = H_0^{(L)}(\boldsymbol{\zeta}) - H_0$ is time-independent. So the integral action difference term becomes

$$\left\|\int_0^t ds\ H_0^{(L)}(\boldsymbol{\zeta}) - H_0\right\|_{\infty,t} = t\left\|H_0^{(L)}(\boldsymbol{\zeta}) - H_0\right\|_{\infty,t} = t\|H_0^{(L)}(\boldsymbol{\zeta}) - H_0\|, \tag{B2}$$

where in the last line, we drop the supremum over time due to time independence. Now we can rewrite

$$\|H_\mathcal{E}(\mathbf{u}(t),t)\|_{1,t} = t\|H_0 + H_c(\mathbf{u}(t),t)\|_{1,t} \leqslant t\left(\|H_0\| + \|H_c(\mathbf{u}(t),t)\|_{1,t}\right) \tag{B3}$$

using the triangle inequality. Combining both facts yields the inequality. □

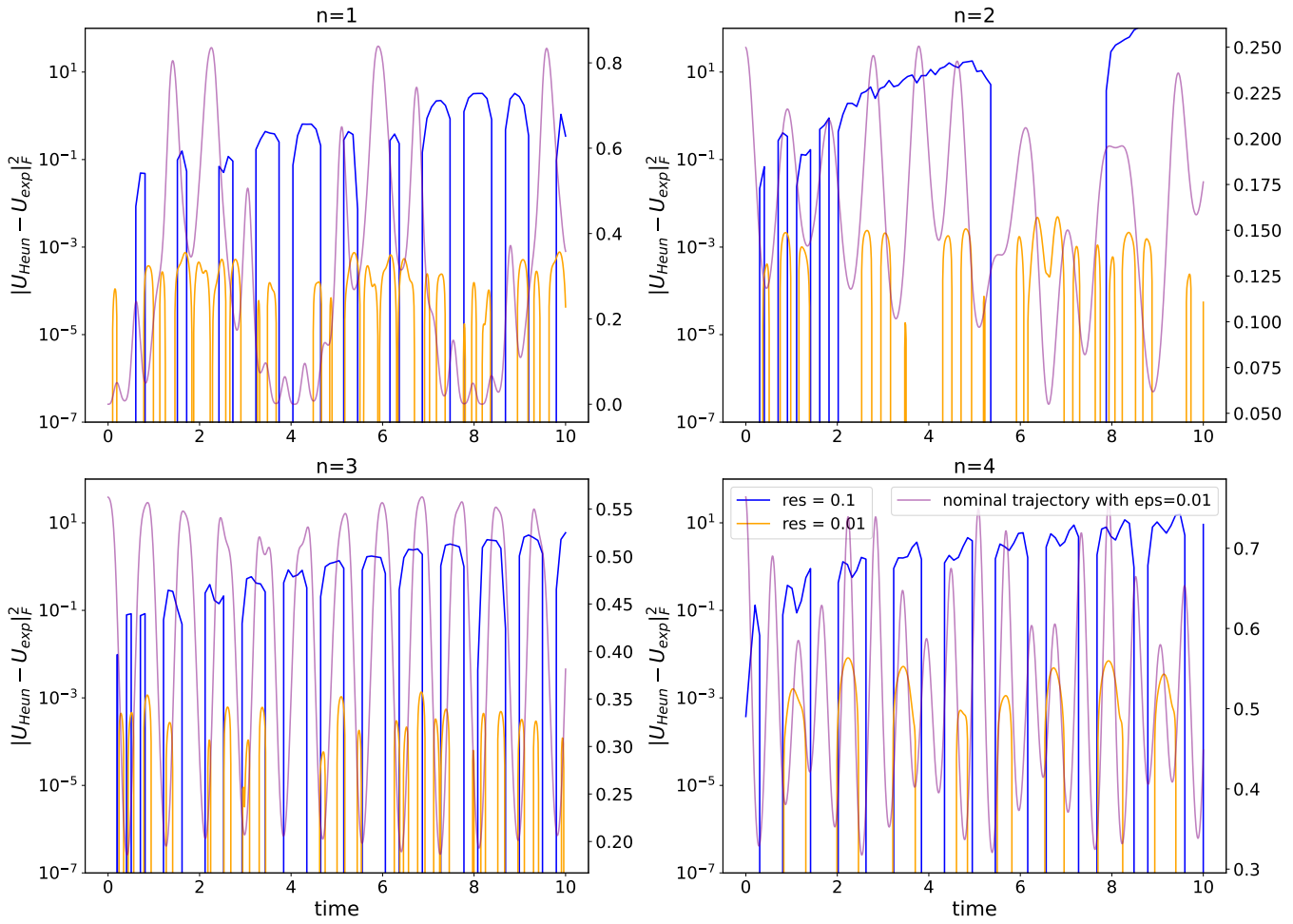


Figure 7. Frobenius norm of the prediction error of the Heun ODE solver [71] compared to the matrix exponential method. The number of qubits n are shown on top of each subfigure. The random time-dependent sinusoidal Hamiltonians are as follows: for $n = 1$, $H = -2.32\sigma_z \cos 2.19t - 0.011 \sin 3.62t + 1.79\sigma_x \cos 4.89t + 3.04\sigma_y \cos 2.69t$; for $n = 2$, $H = 1.01\sigma_z\mathbb{1} \cos 1.44t + 4.51\mathbb{1}\mathbb{1} \sin 4.55t - 2.7\sigma_y\sigma_z \sin 1.07t + 0.48\sigma_x\sigma_z \cos 2.26t$; for $n = 3$, $H = -1.28\mathbb{1}\sigma_x\mathbb{1} \cos 2.62t - 0.23\sigma_y\sigma_z\sigma_y \sin 3.75t - 1.34\mathbb{1}\sigma_y\sigma_x \sin 3.35t + 3.38\sigma_x\sigma_x\sigma_z \cos 2.34t$; for $n = 4$, $H = -0.41\mathbb{1}\sigma_z\sigma_z\sigma_x \sin 2.86t + 2.19\sigma_y\mathbb{1}\sigma_x\sigma_z \sin 1.38t - 0.87\sigma_y\sigma_x\sigma_x\sigma_z \sin 2.26t + 4.06\sigma_x\sigma_x\sigma_z\mathbb{1} \sin 1.76t$ where the shorthand used is $\mathbb{1}\sigma_x\mathbb{1} \equiv \mathbb{1} \otimes \sigma_x \otimes \mathbb{1}$. e.g. Trace fidelities w.r.t. the generalized CNOT (NOT or X-gate for $n = 1$, CNOT for $n = 2$, CCNOT for $n = 3$ and so on) are shown in the twin axis on the right. It can be seen that the step size of 10^{-1} leads to quick accumulation of error seen in the sharp peaks but a step size of 10^{-2} is more stable with more than $O(10^3)$ times less prediction error.

The inequality in Eq. (B1) can be analogously extended to the open system setting w.r.t. the Choi matrix $\mathbf{\Phi}$. Here, we focus on the unitary case for simplicity since the arguments are similar.

There are two observations worth mentioning about inequality Eq. (B1): (a) when all other variables are fixed, the error in the model's unitary predictions w.r.t. to the environment's ground truth grows as a function of time; (b) the model prediction error is a lower bound of the error in the model parameters $H_0(\boldsymbol{\zeta})^{(L)}$ w.r.t. the ground truth parameters H_0. The prediction error $L_{\text{model}}(\mathcal{D}_{\text{val}})$ can be estimated using a validation dataset $\mathcal{D}_{\text{val}}$ and relates this observed validation loss to the Hamiltonian difference. Importantly, the inequality implies that the closeness in the propagator does not always translate to closeness in the Hamiltonian. Therefore, a model Hamiltonian can be locally a good fit for propagator predictions while still having a large Hamiltonian error $\left\| H_0^{(L)}(\boldsymbol{\zeta}) - H_0 \right\|$. So arbitrary closeness in terms of the Hamiltonian error need not be necessary for good unitary predictions. But conversely, if we can be certain that the model Hamiltonian is close to the system Hamiltonian, then the unitaries must be close. This motivates that a good guess (in the form of partial knowledge about the system) of the true Hamiltonian is useful in bounding the prediction errors.

We exploit this fact to learn the local Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ that approximates the dynamics of H_0 w.r.t. $U_{\mathcal{E}}$. Qualitatively, we observe that Hamiltonian error, prop-

agator validation and training error are both improved during training (i.e., the propagator loss on the validation set is predictive of Hamiltonian error). This can be seen in Fig. 8 for the noisy shot setting. But we also note in this example that the learned Hamiltonian $H_0^{(L)}(\boldsymbol{\zeta})$ is local, as seen from the Hamiltonian error plateauing at a non-zero value.

Appendix C: Monotonic Improvement for Model Returns

We show that it is possible to improve the environment's reward under an incorrect model ansatz in $\mathbf{M}_{\boldsymbol{\zeta}}$. For that we need the following result from [27],

Theorem 1. *(Monotonic improvement for model-based returns [27]) Given k-branch rollout returns $\eta_{branch}(\pi)$ for a policy π under the model, the true returns $\eta(\pi)$ are lower bounded*

$$\eta(\pi) \geqslant \eta_{branch}(\pi) - 2r_{max}\left(\frac{\gamma^{k+1}\epsilon_\pi}{(1-\gamma)^2} + \frac{\gamma^k + 2}{1-\gamma}\epsilon_\pi + \frac{k}{1-\gamma}\left(\epsilon_{model}\right)\right) \quad \text{(C1)}$$

where the returns η are defined as

$$\eta(\pi) := \mathbb{E}_\pi\left[\sum_{t=0}^{\infty}\gamma^t\, \mathrm{r}_t(\mathbf{s}_t, \mathbf{a}_t)\right] = \mathbb{E}_{\mathrm{r}_t\sim\mathcal{E}(\mathbf{s}_{t-1},\mathbf{a}_t^\pi)}\left[\sum_{t=0}^{\infty}\gamma^t\, \mathrm{r}_t(\mathbf{s}_t, \mathbf{a}_t)\right]. \quad \text{(C2)}$$

r_{max} is the maximum reward for an MDP transition; the policy error ϵ_π is the upper bound,

$$\epsilon_\pi \geqslant D_{TV}(\pi_D(\mathbf{s},\mathbf{a})\|\pi(\mathbf{s},\mathbf{a})) \quad \text{(C3)}$$

where D_{TV} is the total variation distance and π_D is the data generating policy (i.e., the policy that generated the MDP data by interacting with the environment $\mathcal{E}$). The model error ϵ_{model} is the upper bound

$$\epsilon_{model} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[D_{TV}\left(P_{\mathcal{E}}(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\|P_M(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\right)\right]\right), \quad \text{(C4)}$$

where $P_M(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})$ is the MDP transition probability distribution under the model M that estimates the environment $\mathcal{E}$ and likewise for $P_{\mathcal{E}}$. γ is the discount factor and k is the branch rollout length.

Proof: See proof of Theorem 4.3 in [27]. □

Informally, the theorem states that as long as the returns under the model η_{branch} are improved by at least the tolerance term $2r_{\max}(\cdots)$, then the returns under the environment η are guaranteed to improve. This also assumes that the policy π generating the model returns is reasonably close to the policy that interacts with the environment to generate the MDP data that we use to compute the statistics including the returns. This policy error ϵ_π can be monitored online and controlled while running the algorithm by curtailing its training once it exceeds some tolerance threshold. Moreover, Ref. [27] shows that as long as the dataset size is large enough, the model error ϵ_m can be decoupled from the policy error ϵ_π. The optimal branch rollout length k^* is given by the minimizer of the tolerance. In practice, there are other considerations (e.g., the interplay between various hyperparameters) that need to be accounted for to determine k^*, so it is usually tuned numerically.

Using Thm. 1 for the ODE model, we can indirectly connect the Hamiltonian error using the validation loss $L_{\text{model}}(\mathcal{D}_{\text{val}})$ with ϵ_{model}. If the Hamiltonian error is small, then ϵ_{model} is small and the returns from the model and the environment are similar for any interacting policy π_θ. However, the returns need not be exactly the same and just need to be better than the tolerance provided by the term $-2r_{\max}(\cdots)$ in Eq. (C1) which is a function of ϵ_{model}. The tolerance is smaller for a more accurate model and so less of an improvement of the model returns η_{branch} is necessary. The following lemma makes this idea concrete by applying Thm. 1 to our RL control problem setup.

Lemma 1. *(Model error upper bound for the ODE model) If the model error ϵ_{model} upper bounds the risk,*

$$\epsilon_{model} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))\right]\right) \quad \text{(C5)}$$

then it also upper bounds the unitary prediction error

$$\epsilon_{model} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\left\|U_{\mathcal{E}(\mathbf{s},\mathbf{a})} - U_{\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a})}\right\|_{\infty,t}\right]\right) \quad \text{(C6)}$$

and the total variation distance between the model and environment probabilistic distributions,

$$\epsilon_{model} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[D_{TV}\left(P_{\mathcal{E}}(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\|P_{\mathbf{M}_{\boldsymbol{\zeta}}}(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\right)\right]\right). \quad \text{(C7)}$$

Proof: Since the model $\mathbf{M}_{\boldsymbol{\zeta}}$ and the environment are both deterministic by assumption, we need to modify the lower bound on the model error ϵ_{model} in Thm. 1. We can replace the total variation distance between the two supposed distributions $P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}$ by an indicator variable $\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))$ if $\mathbf{s}'_{\mathbf{M}_{\boldsymbol{\zeta}}} \neq \mathbf{s}'_{\mathcal{E}}$, which is 1 if the transitioned states do not match and 0 if they do. We can upper bound the total variation distance like this since $D_{\text{TV}}(P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}) = \sup_A |P_{\mathcal{E}}(A) - P_{\mathbf{M}_{\boldsymbol{\zeta}}}(A)| \leqslant 1$ in case the probabilities do not match and $D_{\text{TV}}(P_{\mathcal{E}}, P_{\mathbf{M}_{\boldsymbol{\zeta}}}) = 0$ when they match perfectly. Hence, there exists some ϵ_{model} such that

$$\epsilon_{\text{model}} \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))\right]\right) \geqslant \max_t\left(\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[D_{\text{TV}}\left(P_{\mathcal{E}}(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\|P_M(\mathbf{s}'\,|\,\mathbf{s},\mathbf{a})\right)\right]\right).$$

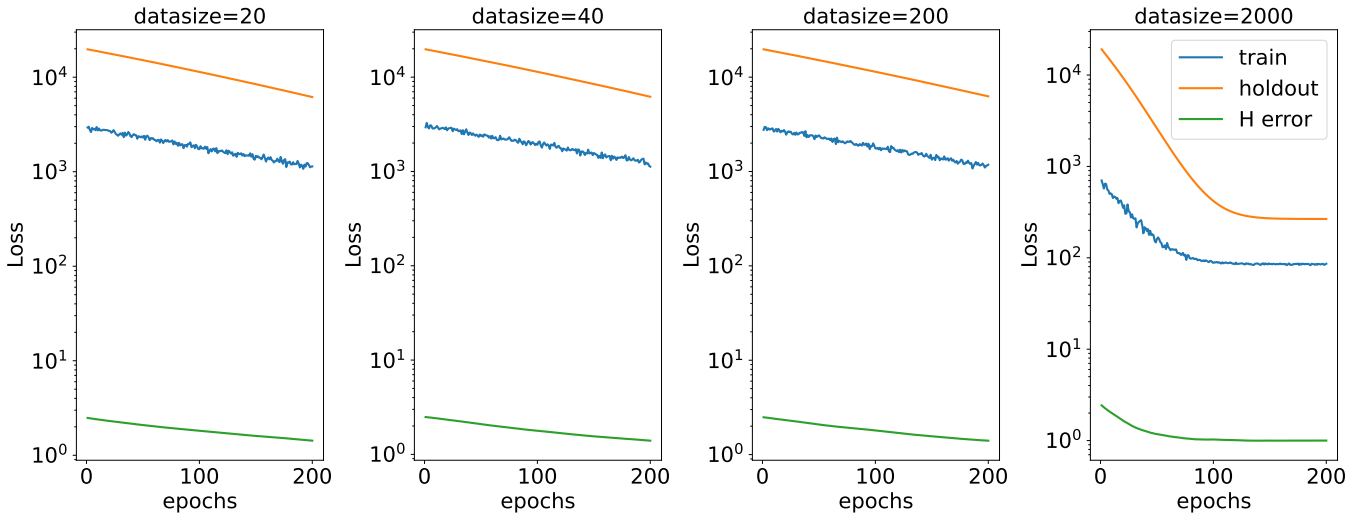


Figure 8. The Hamiltonian error, unitary training $L_{\text{model}}(\mathcal{D}_{\text{train}})$ and validation (holdout) loss $L_{\text{model}}(\mathcal{D}_{\text{val}})$ as functions of training epochs for the two-qubit transmon unitary control problem with noisy measurements and $M = 10^5$. Data size denotes the number of single-step unitary transitions. The validation set is fixed to 5000 transitions under random policy actions $\mathbf{a}_k$. All three error measures improve as a function of training. Adding more training data appears to provide diminishing returns in predicting the local unitary dynamics.

The risk $\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))\right]$ is essentially the fraction of unitaries that the model predicts incorrectly and is related to the unitary error in Prop. 2 by the fact that

$$\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|_{\infty,t} \leqslant \mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a})), \tag{C8}$$

provided that $\left\|U_{\mathcal{E}} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|_{\infty,t}$ is normalised to be in $[0,1]$. So we have

$$\mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\left\|U_{\mathcal{E}(\mathbf{s},\mathbf{a})} - U_{\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a})}\right\|_{\infty,t}\right] \leqslant \mathbb{E}_{\mathbf{s}\sim\pi_D^{(t)}}\left[\mathbb{I}(\mathbf{M}_{\boldsymbol{\zeta}}(\mathbf{s},\mathbf{a}) \neq \mathcal{E}(\mathbf{s},\mathbf{a}))\right]. \tag{C9}$$

So ϵ_{model} upper bounds the expected unitary error if and only if ϵ_{model} upper bounds the expected risk in the unitary prediction error. □

Appendix D: How Much Data is Needed for Model Training?

A hallmark for a good ansatz for the model $\mathbf{M}_{\boldsymbol{\zeta}}$ estimating the dynamics of the controllable system would be less demand of supervised learning MDP data needed for low prediction error.

We consider the Hamiltonian error, unitary train and holdout error. Hamiltonian error δ is the spectral norm error between the learned and true system Hamiltonian. The others are mean squared errors. Cross-validation is used to estimate the model's generalization ability on a holdout dataset of unseen random unitary data, also sampled from the MDP transitions and collected by the policy π during training.

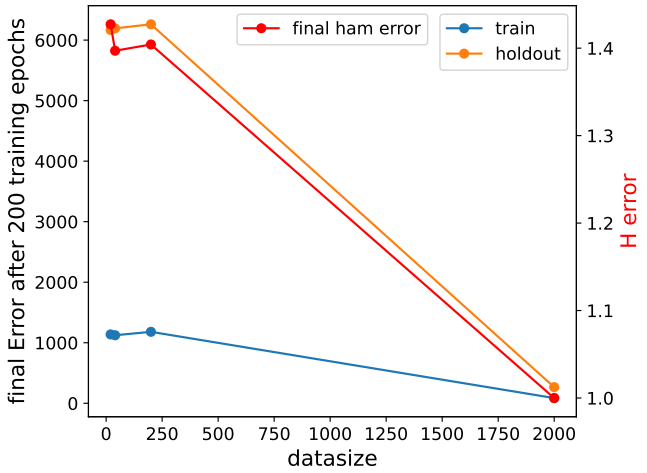


Figure 9. Effect of training data size on model generalization metrics: Hamiltonian error, unitary training $L_{\text{model}}(\mathcal{D}_{\text{train}})$ and validation (holdout) loss $L_{\text{model}}(\mathcal{D}_{\text{val}})$ for noisy single shot measurement-based unitary control of the transmon.

As seen from Fig. 8, for the two-qubit transmon control problem, for very small dataset sizes comprising $20-200$ unitary transitions, the single step unitary prediction error is large compared to training with about $2,000$ unitaries or about 100 full length pulses with 20 timesteps, though the decrease in error is diminishing with dataset size. All errors are in agreement across the datasets over 200 training epochs. This is further corroborated by Fig. 9 where the final errors after 200 epochs are plotted. There is a reduction in the final errors for the 2000 dataset size, but the improvement is diminishing in mag-

nitude and plateaus at this loss for larger dataset sizes. This is still much less than what was required to train a neural network model for $\mathbf{M}_{\boldsymbol{\zeta}}$ during the initial stages of our research where the training dataset size needed to be of the order of 10^6. Moreover, these experiments provide us with an idea of what dataset size to use to train the model $\mathbf{M}_{\boldsymbol{\zeta}}$ by setting the number of initial exploration MDP transitions to add to the policy's buffer for the transmon control problem. We also adopted multiple training phases to continuously train $\mathbf{M}_{\boldsymbol{\zeta}}$ using fresh batches of training data collected by the policy.

Appendix E: Leveraging the Learned Hamiltonian for the Two-qubit NV Center

Similar to the results found in Sec. IV C, here we report the structural differences between the learned and target Hamiltonians for the two-qubit NV center.

The matrix difference between the true H_0 and learned Hamiltonian $H_0(\boldsymbol{\zeta})$ is,

$$H - H_0(\boldsymbol{\zeta}) = \begin{bmatrix} 0.0116 & 0.0013i & -0.0001 - 0.0002i & -0.0007 \\ -0.0013i & -0.0111 & -0.0001 + 0.0002i & 0.0003 + 0.0003i \\ -0.0001 + 0.0002i & 0.0001 + 0.0002i & -0.0108 & -0.0005 - 0.0002i \\ -0.0007 & 0.0003 - 0.0003i & -0.0005 + 0.0002i & -0.013 \end{bmatrix}$$

Moreover, the non-linear relationship between the model prediction errors and the spectral norm error δ or the mean squared Pauli expectation value error is confirmed as before in Fig. 10(a). Local and global trajectory differences under a random control pulse and the results of using GRAPE on RL controllers are shown in Fig. 10(b) and (c) respectively. The learned Hamiltonian is able to improve the controller fidelities to greater than 0.999.

Appendix F: Three-qubit transmon Control Problem

In this section we discuss the issue of scalability of LH-MBSAC's performance related to the three-qubit transmon control problem in Sec. IV E in detail.

Working with two level systems, we extend the two-qubit transmon Hamiltonian to its three-qubit version $H^{(3)}_{\text{tra}}$. The system part generalizes trivially. For the control part $H^{(3)}_{\text{tra}_c}$, we generalize the cross resonance interaction presented in Ref. [73] to construct the following time-dependent part of the three-qubit transmon Hamiltonian,

$$\frac{H^{(3}_{\text{tra}_c}(t)}{\hbar} = \sum_{l=1}^{3} \Big(a_l(t)(Z_l X_{l+1} + X_{l+1} + Y_{l+1} + Z_l) + b_l(t)(X_l Z_{l+1} + X_l + Y_l + Z_{l+1}) \Big) \quad \text{(F1)}$$

where $a_l(t), b_l(t)$ are the real drive amplitudes and X_l, Y_l, Z_l are the corresponding Pauli operators on the lth qubit.

To start, we mention our hyperparameter strategy. Only an initial hyperparameter search is performed for the two-qubit transmon control problem, and we were successfully able to transfer the same hyperparameters to all problems in the paper that were studied including the ones presented in Fig. 2.

It is a desirable property for the stabiltiy of RL algorithms to be robust to hyperparameter changes for different target problems, which we found to be the case. The search was only conducted for the model-free SAC since LH-MBSAC is just a model-based augmentation of the underlying SAC algorithm so there is no strong reason for the hyperparameters to fail to transfer.

However, for the three-qubit transmon control problem, we encountered issues and had to repeat the search. This was extensive, and what we focused on are: more initial exploration data, using bigger layer sizes for the policy and value function neural networks, changing the learning and update rates for the policy and value functions, amongst other things. An extremely thorough search is difficult since the problem is more computationally challenging, and it is hard to determine when to terminate the training during a trial run that necessarily needs to be premature during the hyperparameter search. Please see the accompanying code for the list of hyperparameters we searched over using Bayesian optimization in `tune_hypers.py` along with some results in the `hyper_tests` folder.

Furthermore, we make observations that make this issue seem less like a hyperparameter issue and more like an optimization landscape problem:

1. The values and the gradients for policy and value functions that saturate are both stuck in suboptimal extrema and ultimately we get stuck at a prematurely optimized reward function. This is illustrated in Fig. 11. Essentially, SAC gets stuck in a loop mining medium level fidelities and its policy outputs saturate on the extremes of the control amplitudes. It is already detailed in Sec. IV E that RL pulses are biased towards maintaining high intermediate fidelities due to the nature of the MDP used in the paper. Fig. 6 example pulses found by

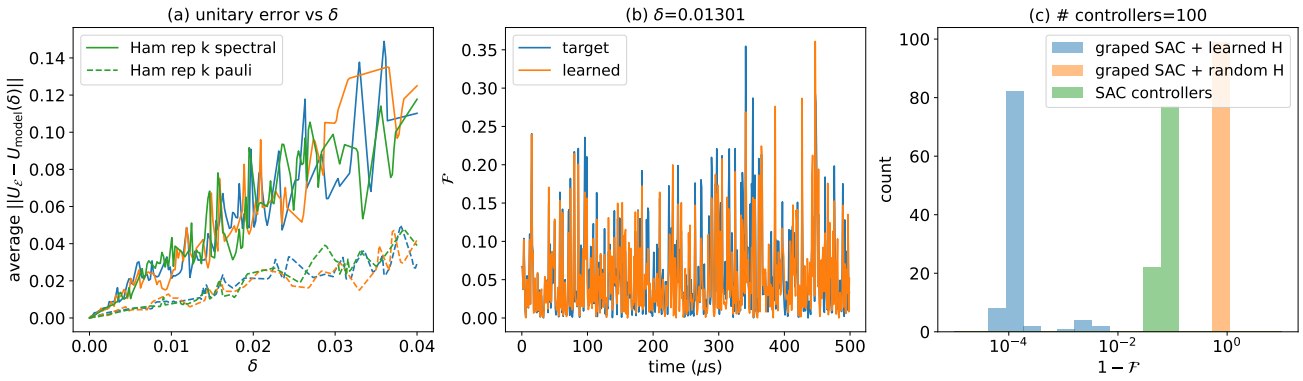


Figure 10. (a) The non-linear relationship between the prediction error $\left\|U_\mathcal{E} - U_{\mathbf{M}_{\boldsymbol{\zeta}}}\right\|$ and Hamiltonian spectral norm error or mean squared Pauli expectation value error δ for the two-qubit NV center Hamiltonian. For the same 1000 random control pulses, we evaluate the average unitary prediction error of $\mathbf{M}_{\boldsymbol{\zeta}}$ with increasing δ for three different uniform randomly sampled two-qubit Hamiltonians $H_0(\boldsymbol{\zeta})$. (b) Local and global unitary trajectories: $\mathcal{F}$ as a function of a random control pulse with either the learned $H_0(\boldsymbol{\zeta})$ or true H_0. The learned trajectories and global trajectory overlap less with increasing time with the spectral norm error of $\delta = 0.01301$ and a global phase factor $\text{Tr}[H - H_0(\boldsymbol{\zeta})]$ of ~ 0.01. (c) The learned $H_0(\boldsymbol{\zeta})$ can be leveraged using GRAPE to further optimize the fidelities of LH-MBSAC's controllers. Repeating the procedure in Sec. IV C, yields fidelities of greater than 0.999.

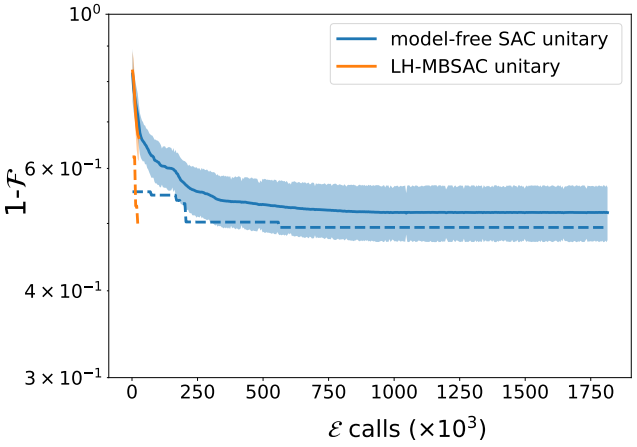


Figure 11. Noiseless unitary sample complexity for the three-qubit transmon where the target gate is the Toffoli gate. Since LH-MBSAC is based on SAC, the latter's training curves are obtained first to see if it viably solves the problem, and it was trained for much longer i.e. in the order of millions of samples as seen in Fig. 11. Mean (solid) and maximum fidelities (dashed) saturate as the policy and value functions gradients and outputs saturate due to the agent getting stuck in a sub-optimal extremum of the optimization landscape.

RL vs. GRAPE for the two-qubit transmon, confirming this.

2. Since we have the model Hamiltonian, we insert it into GRAPE initialized with the highest fidelity SAC controller values, and it also gets stuck (at slightly better fidelities).

Despite these issues, the system Hamiltonian is still learned. It can be inserted into GRAPE with uniform random initialization of control pulse parameters to achieve fidelities of over 0.999.

Appendix G: Comparison of Fidelities for Lindbladian Dynamics

We study the agreement between three different fidelity measures of realized noisy gates on open systems with Lindblad decay and decoherence for the two-qubit transmon gate control problem. The fidelity measures are the diamond norm fidelity [65], the generalized state fidelity [48], and the average gate fidelity [74]. The diamond norm fidelity, derived from the diamond norm or the completely bounded trace norm, is the most expensive to compute as it involves solving a convex optimization problem:

$$\begin{aligned}\mathcal{F}_\diamond(\mathbf{\Phi}(\mathbf{u}(t),t),\mathbf{\Phi}_{\text{target}}) &= 1 - \|\mathbf{\Phi}(\mathbf{u}(t),t) - \mathbf{\Phi}_{\text{target}}\|_\diamond \\ &= 1 - \max_\rho \|\mathbf{\Phi}(\mathbf{u}(t),t)\circ\rho - \mathbf{\Phi}_{\text{target}}\circ\rho\|_1\,, \quad \text{(G1)}\end{aligned}$$

where the maximization is over the space of all density matrices ρ. This can be done by solving an equivalent semi-definite program [75]. $0.5 \leqslant \mathcal{F}_\diamond(\mathbf{\Phi}(\mathbf{u}(t),t) \leqslant 1$.

To study the sensitivities of the measures to dissipation and their agreement w.r.t. each other, we consider low, medium and high dissipation regimes. We evaluate 100 of our controllers found for the noisy single shot measurements setting of the two-qubit transmon in these regimes. The results are plotted in Fig. 12. Here, `deca` and `deco` refer to inverse decoherence and decay rates $2/T_l^*, 2/T_l$ respectively, for the lth qubit, measured in MHz. We re-normalize the trace of the realized operator $\mathbf{\Phi}(\mathbf{u}(t),t)$ during our experiments, as is standard practice. Due to the exhaustive nature of its computation,

$\mathcal{F}_\diamond$ is the most sensitive to noise and loss of coherence out of all the measures. The generalized state fidelity is the least sensitive and the average gate fidelity falls in the middle. For very low to medium dissipation levels, e.g., $(0.05, 0.05)$, $(0.05, 0.1)$, or $(0.05, 0.2)$ for the pair $(\texttt{deca}, \texttt{deco})$, the generalized state fidelity is near perfect while the gate and diamond norm fidelities are more sensitive and closer to 0.9. For this reason, in Sec. IV D, we chose to use the diamond norm fidelity to more accurately gauge controller performance—this was especially true for the low dissipation regime results.

As a side note, some controllers shown in Fig. 12 are more robust to dissipation than others as revealed by the noisy variation across the controller index vs. fidelity plot. The controllers are not ordered, so the fidelity in the zero dissipation regime has some noise/variation as seen for $\texttt{deca}, \texttt{deco} = (0.05, 0.05)$. Across all the subfigures, the robustness is captured by all the fidelity measures where the variation magnitudes and positions are more or less aligned.

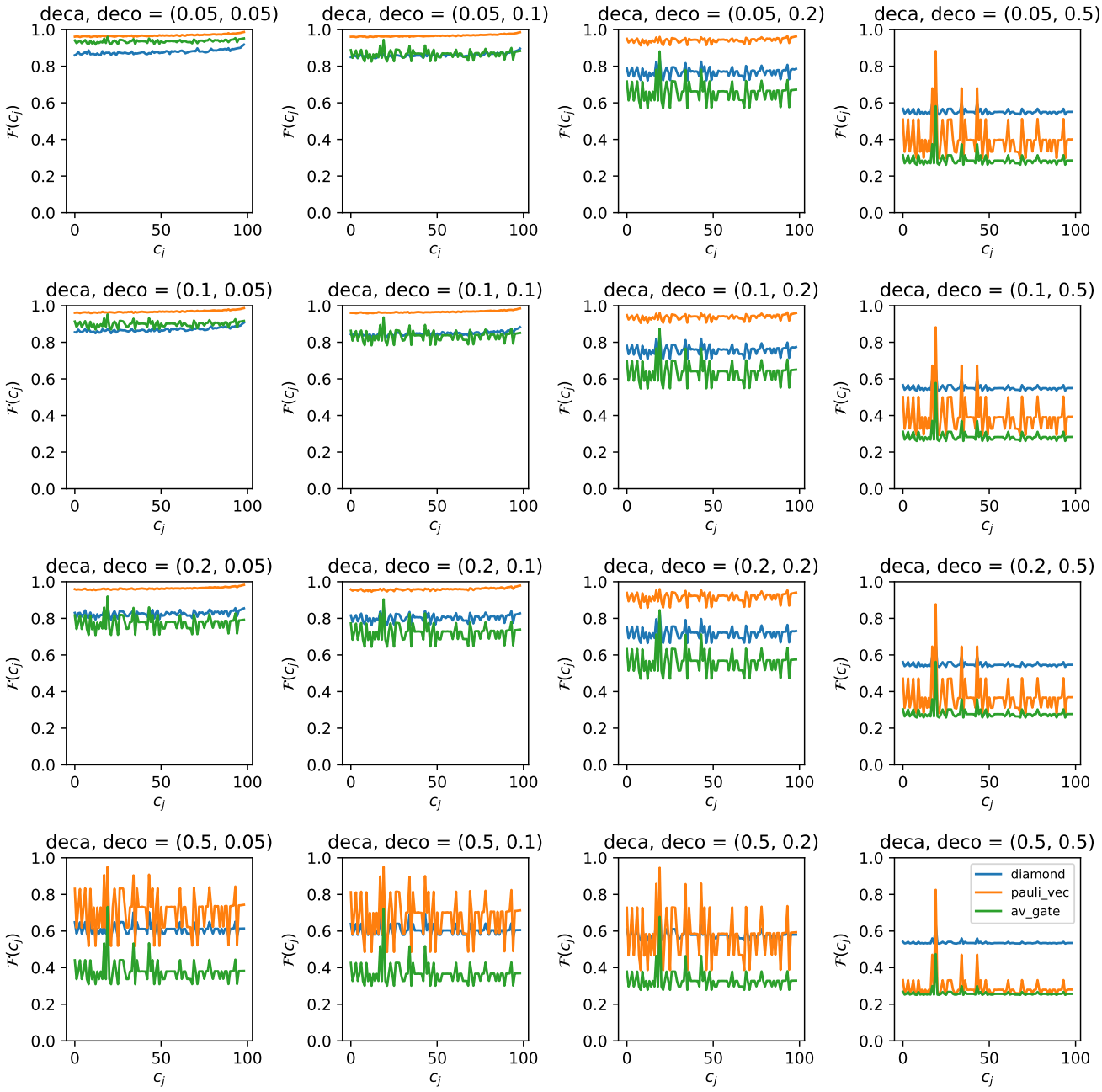


Figure 12. How much the fidelity measures relate to one another as the dissipation strength varies in terms of the decoherence and the decay coefficients in Eq. (6) for the Lindbladian l_d operators. Here, `deca`, `deco` refer to inverse decay and decoherence rates $2/T_l^*, 2/T_l$ respectively, for the lth qubit measured in MHz. The x-axis refers to a controller c_j obtained for the two-qubit transmon gate control problem with single shot measurement noise where the target is the CNOT gate. The controllers are in random order w.r.t. the fidelity, but the ordering is preserved across each subfigure. The number of single shot measurements is 10^6 and `diamond`, `pauli_vec`, `av_gate` refer to the diamond norm fidelity [65], the generalised state fidelity [48] and the average gate fidelity [74].