Reinforcement Learning vs. Gradient-Based Optimisation for Robust Energy Landscape Control of Spin-1/2 Quantum Networks

I. Khalid C. A. Weidner E. A. Jonckheere S. G. Schirmer F. C. Langbein

Abstract— **We explore the use of policy gradient methods in reinforcement learning for quantum control via energy landscape shaping of XX-Heisenberg spin chains in a completely model agnostic fashion. Their performance is compared to finding controllers using gradient-based L-BFGS optimisation with restarts, with full access to an analytical model, target functional and its gradient. Hamiltonian noise and coarse-graining of fidelity measurements are considered. Reinforcement learning is able to tackle challenging, noisy quantum control problems where L-BFGS optimization algorithms struggle to perform well. Robustness analysis of the controllers found under different levels of Hamiltonian noise indicates that controllers found by reinforcement learning appear to be less affected by noise than those found with L-BFGS.**

I. INTRODUCTION

Finding robust solutions to Hamiltonian control of quantum devices from superconducting qubits to spintronic circuits to microwave QED to trapped ions [1], [2], [3], [4] is crucial to achieve high-fidelity operations in quantum systems that form the building blocks of Noisy Intermediate Scale era Quantum (NISQ) devices [5]. Although early-stage devices are expected to be error-prone and in limited in size, they could pave the way to revolutionize computation and simulation at a fundamental level, and have already proven to be effective tools in physically simulating molecular networks [6], [7], [8]. Currently, there are two challenges for NISQ devices: scalability with system size and robustness to known and unknown uncertainties. For the former, most of the problem lies in exploration of an exponentially growing parameter space in the size of the system, which has been addressed using variational approaches [9], [10] amongst many others. In this paper we focus on the latter challenge: optimal control with partial observability in the absence of an accurate physical model, a regime that is particularly challenging for the dominant, model-based, open-loop control approaches.

Two frameworks developed for such control — dual control theory initiated by Feldbaum in the 1960s [11], [12] and reinforcement learning (RL) for optimal control [13] — both coalesce the control problem to approximate dynamic programming solved using Bellman's principle of optimality [14]. The philosophy is that of initial exploration and learning of the unknown system model by probing it for data, and, later, exploiting that information to control it. Initially the control actions taken by the controlling agent are sub-optimal as it works with a highly uncertain model although they can still be seen as optimal in the sense of solving the Bellman equation step-wise based on the acquired information. Iterated composition of the solutions achieves near optimal solutions, eventually.

Our motivation for turning to RL is to look at adaptive model-agnostic ways of performing general optimization tasks; specifically, the quantum control problem of Eq. (1). These methods, in principle at least, promise to have less overhead compared with functional variation or Pontyragin-variation-based methods for optimal control which use an analytical model, and have been the focus of over half a century of fruitful contribution to quantum control, including algorithms such as GRAPE [15] and Krotov [16] that utilise gradient-based optimisation of a model-based target functional. Limited knowledge about the system and control Hamiltonians as well as interactions with the environment, however, has a strong effect on the performance of such control schemes.

RL methods are either model-based or model-free, but all methods can in principle be fully model-agnostic. Model-based methods involve creation of a model from scratch, whereas model-free methods skip this step entirely. Prior work demonstrated the usefulness of deep RL for quantum optimal control in its application to synthesis of transmon gates [17], coherent transport by adiabatic passage through semi-conductor quantum dots [18] and robust two-qubit gmon gate synthesis [19]. RL is an interesting paradigm to follow as it aims to tackle and optimize the trade-off between exploitation and exploration that is the hallmark of dual control.

In this paper we employ RL to find robust quantum controls with a fully model-agnostic approach using single shot measurements, which can be collected experimentally. Instead of passing unitary operators or density matrices as the observable information to the RL agent, as considered in previous work, we only give it access to experimentally observed data and the control parameters that it can change [10]. This is in line with real world scenarios where an RL agent might be needed to be deployed in an experimental setting with high levels of uncertainty, commonly seen in current setups. We further provide a computational resource comparison between some classes of policy-gradient-

I. Khalid and F. C. Langbein are with the School of Computer Science and Informatics, Cardiff University, UK KhalidMI@cardiff.ac.uk, frank@langbein.org.

C. A. Weidner is with the Institute for Physics and Astronomy at Aarhus University, Denmark cweidner@phys.au.dk

E. A. Jonckheere is with the Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089, jonckhee@usc.edu.

S. G. Schirmer is with the College of Science, Swansea University, Swansea, Wales, UK, s.schirmer@swansea.ac.uk.

based RL algorithms to motivate our choice of PPO (Proximal Policy Optimisation) as the best performing algorithm. We also demonstrate the resilience of RL methods in finding optimal controllers to (1) measurement and (2) Hamiltonian noise, where analytical methods break down or consume too many resources. Of course analytical model optimization has an advantage over RL when the model describes the physical system well, as no exploration is required. Increasing uncertainties in the model, however, make an RL or exploratory approach necessary. Moreover, although L-BFGS is more likely to find high-fidelity controllers, preliminary robustness analysis for the controllers found by RL suggests that they may be more robust to noise than those found by L-BFGS.

This paper is organized as follows. Section II presents the control problem formulated in an RL paradigm and its noise model augmentations. Main results about algorithmic performances of RL and quasi-Newton methods and their robustness analysis are presented in Section III. Section IV presents a discussion of this work and potential future directions.

II. Preliminaries

A. The Control Problem

We consider a simple spin-1/2 system, the XX-Heisenberg spin chain model, to introduce our formulation of the quantum control problem for RL. Its Hamiltonian H_{spin} is

$$\begin{aligned} H_{\text{spin}} = H_C + H_0 &= \sum_{n=1}^{N} \Delta_n H_n + H_0 \\ &:= \sum_{n=1}^{N} \Delta_n Z_n + \sum_{m\neq n}^{n} J_{mn}\left(X_n X_m + Y_n Y_m\right) \end{aligned} \tag{1}$$

where H_C is the control Hamiltonian and H_0 describes the natural spin dynamics. $X_n = F_n(\sigma_x) := \mathbb{1}^{(1)}\otimes\cdots\otimes\mathbb{1}^{(n-1)}\otimes \sigma_x^{(n)}\otimes\mathbb{1}^{(n+1)}\cdots\otimes\mathbb{1}^{(N)}$ is the expanded Pauli X operator σ_x applied on the n-th spin in the system; $\Delta = \{\Delta_n\}$ are external control parameters; J_{mn} are the interaction couplings between spin n and m. Y_N, Z_n are defined equivalently to X_n for the Pauli operators σ_y and σ_z. We only consider a spin chain with uniform nearest-neighbour couplings such that $J_{n,n\pm1} = 1$ (and all other entries are 0), which can be thought of as a type of quantum wire. The spin network Hamiltonian H_{spin} commutes with diagonal operators and therefore the dynamics can be decomposed into excitation subspaces [20]. Here, we are only concerned with single excitations, i.e., only one bit of information can propagate through the network at a given time, for simplicity. Therefore we have the single excitation subspace Hamiltonian

$$H_{ss} := \sum_{n} \Delta_n |n\rangle\langle n| + \sum_{m\neq n} J_{mn}|m\rangle\langle n|. \tag{2}$$

The unitary time evolution of the single bit propagating through the network is given by the Schrödinger Eq., $i\hbar\frac{d}{dt}|\psi(t)\rangle = H_{\text{ss}}(t)|\psi(t)\rangle$, where $|\psi(t)\rangle$ is the N dimensional spin-state vector. This is solved by

$$|\psi(t)\rangle = \mathcal{T}\exp\left[-i\int_{t_0}^{T} H_{\text{ss}}(t)\,dt\right]|\psi(t_0)\rangle = U_\Delta(T)|\psi(t_0)\rangle \tag{3}$$

where time t is measured in units of $1/\hbar$ and U_Δ is the unitary exponential of the generating Hamiltonian from time t_0 to T and $\mathcal{T}$ is the time ordering operator. The suffix Δ makes the dependence of the unitary on the control parameters explicit. Consider some target state $|\psi^*\rangle$ and a state propagated by the unitary from an initial state $|\psi(t_0)\rangle$. The state propagation performance is given by the fidelity

$$\mathcal{F}_\Delta := |\langle\psi^*|U_\Delta(T)|\psi(t_0)\rangle|^2, \tag{4}$$

measuring how close the propagated and target states are. The resulting optimal control problem is the determination of the control parameters Δ that, e.g., represent the action of applied external magnetic fields, s.t.

$$\Delta^*, T^* = \arg\max_{\Delta,T} \mathcal{F}_\Delta. \tag{5}$$

We specifically consider transitions between one-hot encoding state vectors (canonical Euclidean basis vectors), consistent with a single bit propagating through the network.

The most common paradigm for quantum control is dynamic [21], [22], i.e., assuming time-dependent controls, $\Delta_n(t)$, the implementation of which typically requires the ability to rapidly modulate or switch controllers implemented by physical fields (e.g. lasers or magnetic fields). An alternative to dynamic control is time-invariant control, i.e., time-independent control parameters Δ_n [23]. This is analogous to shaping the potential landscape to facilitate the flow of information from an initial state to the target state. For example, information encoded in electron or nuclear spins in quantum dots whose potential can be controlled by varying voltages applied to surface control electrodes, creating a potential landscape. The static control problem has fewer parameters, and so, in some sense, is simpler and smaller. Moreover, previous work found evidence concerning good robustness properties of the static controls [24]. They may also be simpler to implement experimentally as we do not need to modulate control fields, or could be part of a more complex dynamic control scheme. However, the optimisation landscape is challenging [23], and there is no guarantee that the controllers Δ_n found are robust with respect to uncertainties in the system and interactions with the environment.

B. Reinforcement Learning Control Paradigm

RL is formulated in the context of a finite Markov decision process (MDP): given an initial state $\mathcal{S}$, a next state $\mathcal{S}'$ can be achieved that carries with it some reward $\mathcal{R}$ by performing some action $\mathcal{A}$. State transitions are assumed to be Markovian and probabilistic and captured by the dynamics model $P(\mathcal{S}', \mathcal{R}|\mathcal{S}, \mathcal{A})$, indicating the probability of going from $\mathcal{S}$ to $\mathcal{S}'$ with the action $\mathcal{A}$, gaining $\mathcal{R}$. A trainable policy function $\pi(\mathcal{A}|\mathcal{S})$ is a non-parametric probability distribution of executing action $\mathcal{A}$ given state $\mathcal{S}$. An RL agent following π and interacts with an environment $\mathcal{E}$ associates a state transition $Y : \mathcal{S} \xrightarrow{\mathcal{A}} \mathcal{S}'$ with a reward function $\mathcal{R}(Y)$. A state-action value function $Q(\mathcal{S}, \mathcal{A})$ or the value function $V(\mathcal{S}) = \max_a Q(\mathcal{S}, a)$ is learnt via the feedback loop interaction of π with $\mathcal{E}$. The environment can be noisy and highly stochastic and yet through the high learning potential of differentiable neural nets as function approximators, a near-optimal π or Q can be learnt for general control tasks and more [25]. Learning Q, for example, involves approximately solving the Bellman optimality equation iteratively, as an update rule, at every timestep k,

$$\begin{aligned} Q_k(s,a) &:= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k \mathcal{R}_{\tau+k+1}|S_\tau = s, A_\tau = a\right] \\ &\equiv \sum_{s',r} P\left(s',r|s,a\right)\left[r+\gamma\max_{a'}Q_{k-1}(s',a')\right] \end{aligned} \tag{6}$$

where γ is some future discounting factor and s', a', r are the summed over next state, next action and reward. Note that Q_k is also the expectation over different policy functions π of the total discounted rewards obtained from the current timestep onwards.

General theorems for policy and Q (or value) functions guarantee iterated policy improvement. This involves computing a new policy, e.g., $\pi'(s) = \arg\max_{a'} Q(s, a')$ for actions a' drawn from some old policy π. A model is thus not needed for approximately solving the Bellman equation as we can directly optimize over the policy by successively computing better policies (e.g. greedily) to yield an optimal Q function $Q^*(s, a)$,

$$Q^*(s,a) = \max_\pi Q(s, \pi(s)). \tag{7}$$

For continuous state and action spaces, this approach does not work well. For such high dimensional spaces, we optimize over the policy by making use of the gradient of some expected cumulative performance distribution in terms of the gradient of a differentiable policy π_θ. Here, π_θ is represented by a linear two-layer neural network with θ nonparametrically denoting its trainable weights and biases. We assume a similar nonparametric neural network form for Q and/or the value function. Many policy gradient algorithms are based on this idea [25]. Using backpropagation [26] to update θ in the direction of the policy gradient, we improve the policy π_θ. By approximately solving the Bellman Eq. (6) iteratively for finite steps, we evaluate how well it does. Improvement and evaluation are repeated until a convergence criterion is met that we state below.

For our control problem, we define the model agnostic MDP: for learning timestep τ, let $\mathcal{S}_\tau := \{\Delta_{\tau-1}, t_{\tau-1}\}$ and $\mathcal{A}_\tau := \{\delta\Delta_{\tau-1}, \delta t_{\tau-1}\}$ be an action changing $\mathcal{S}_\tau$ by the given values. The reward is $\mathcal{R}_\tau := \mathcal{F}(|\psi_{\tau-1}\rangle, |\psi^*\rangle)$ where $t_{\tau-1} = T$ is time for which the Hamiltonian is evolved. The readout time $t_{\tau-1}$ with the $\Delta_{\tau-1}$ are the control parameters for π to change such that the reward is improved. Note that this means π is a control landscape exploration strategy with the aim to find control parameters that achieve the state transition. So the goal, rather than the path to get there, is important, even if of course a shorter path makes finding the goal more efficient. We construct an environment $\mathcal{E}$ that a differentiable policy π_θ can interact with to obtain $(\mathcal{S}_\tau, \mathcal{A}_\tau, \mathcal{R}_\tau)$. The state vector satisfies $\mathcal{S}_\tau = \mathcal{S}_\tau \mod \mathcal{S}_{\text{limit}}$ and we set the the limit $\mathcal{S}_{\text{limit}}$ to be ± 10 for $\Delta_{\tau-1}$ and 30 for $t_{\tau-1}$ to ensure that the control parameters are physical and realisable in experiments. A reward threshold, e.g. 0.99, is set as a convergence criterion yielding a single solution vector $\mathcal{S}^*_\tau$ effectively reducing the problem to optimal time-independent Hamiltonian searching. The RL optimization procedure is run for some number of epochs until the reward threshold is achieved. Each epoch consists of a fixed number of timesteps of exploring the landscape from an initial random position. The policy parameters θ and the Q function are updated via backpropagation every epoch.

The utility of the fact that RL assumes nothing about the analytical form of the model is expected to be useful if the environment $\mathcal{E}$ is stochastic. To test this hypothesis, we consider two noise models: (1) directly augmenting H_{ss} with a structured perturbation $P \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$ where P is a matrix of the same form as H_{ss}, i.e. tridiagonal, with normally distributed random values with variance σ^2_{noise} and mean 0. This simulates noisy or tunably inaccurate physics, e.g. due to leakage of spin couplings. (2) coarse-graining the fidelity $\mathcal{R}_\tau$ to simulate single-shot or inaccurate measurements by replacing it with $\tilde{\mathcal{R}}_\tau \sim \text{Bin}(M, \mathcal{R}_\tau)$, drawn from a binomial distribution where M is the number of measurements made and $\mathcal{R}_\tau$, the true fidelity, is the binomial probability and $\tilde{\mathcal{R}}_\tau$ represents the average single shot measurements to estimate the fidelity probabilities.

We only consider leakage within the nearest neighbour spins. Another possible source of noise could be leakage to the next nearest neighbours due to cross-couplings between spins in transmon systems or finite laser beam sizes in cold atom or ion systems. For the purposes of this work, however, we neglect next-nearest neighbor coupling as it is negligible or can typically be mitigated in practical systems. Note that we have also made the actions $\mathcal{A}_\tau$ noisy by perturbing the diagonal of H_{ss} but we could have also coarse-grained the actions to account for the finite resolution of the magnetic or laser field that actually implements the controls in a real experiment.

C. Policy Gradient Reinforcement Learning Algorithms

Within the policy gradient subset of RL, we try a number of algorithms to empirically evaluate which one is most suitable for our static control problem. We consider trust region policy optimization (TRPO) [27], proximal policy optimization (PPO) [28], deep deterministic policy gradient optimization (DDPG) [29], twin policy delayed DDPG (TD3) [30] and REINFORCE [25].

REINFORCE is a pure policy-based algorithm that applies a stochastic gradient ascent update to the policy parameters $\theta \leftarrow \theta + \nabla V_{\pi_\theta}(\mathcal{S}_0)$ for some initial state $\mathcal{S}_0$. The value function gradient is computed using the policy gradient theorem as $\mathbb{E}_\pi\left[\sum_{k=0}^{\infty}\gamma^k \mathcal{R}_{\tau+k+1}\nabla\pi_\theta/\pi_\theta\right]$ via Monte Carlo sampling over trajectories following π.

The others are actor-critic algorithms with an acting policy critiqued by Q_{π_θ} or V_{π_θ}. The actor-critic methods make use of a replay buffer to store MDP transitions of the form $(\mathcal{S}_\tau, \mathcal{S}_{\tau+1}, \mathcal{A}_\tau, \mathcal{R}_\tau)$ and update Q_{π_θ} or V_{π_θ} following the Bellman update (Eq. (6)) by random sampling batches of $\{\mathcal{S}', \mathcal{S}, \mathcal{R}\}$. TD3 and DDPG make use of the deep deterministic policy gradient for θ updates [31] and TRPO and PPO use a variant of the natural policy gradient [32]. TD3 uses two Q functions and backpropagated updates are in the direction of least change while DDPG employs a vanilla combination of Q and a deterministic policy function jittered with correlated exploration noise. Note that there is no objective constraint on the policy that makes sure it does not vary wildly during parameter updates for different episodes. PPO and TRPO improve upon this by using a KL-divergence constraint between the new and old policy to make sure its variation is constrained during each update. TRPO uses a trust region method [33] to compute the Hessian of the KL-divergence with a backtracking line search [34] to update the parameters of the policy. PPO is simpler and uses clipped variation bounds on the KL-divergence that is used directly in the parameter updates of the policy.

III. RESULTS

A. Cost of Reinforcement Learning Algorithms

We first analyse the cost of the policy gradient algorithms from Section II-C. The costs are expressed as the number of environment $\mathcal{E}$ (or target functional) calls, corresponding to estimating the fidelity via single-shot measurements, for an algorithmic run that successfully terminates at a fidelity threshold. This closely links the performance to experimental costs and makes different algorithms comparable without resorting to timing or iteration counts.

We choose to study a noisy transition $|0\rangle \rightarrow |2\rangle$ for chains of length $N = 3, \ldots, 7$. We use 100 single-shot fidelity measurements to estimate the fidelity of a controller and a Hamiltonian perturbation noise of $\sigma_{\text{noise}} = 0.05$. The *"perceived" fidelity* is the stochastic fidelity produced by the noisy environment, as observed from noisy measurements. We compare it to the *"true" fidelity* of the controller under ideal conditions without noise. A perceived fidelity threshold of 0.99 is set as termination criterion. Fig. 1 shows the median performance of DDPG, PPO and TD3 over 50 runs. In terms of environment calls, DDPG performs significantly worse compared to PPO and TD3, but it is more difficult to decide between the other two.

TRPO and REINFORCE were excluded from the study as sufficient statistics could not be obtained. Their behaviour was highly variable and inconsistent due to a lack of successful termination which prevented further analysis. For REINFORCE, we suspect that this was because of the absence of a replay buffer to sample a sufficient variation of transitions and a value/Q function that maps actions to expected rewards to ground policy parameter updates. Similarly, TRPO, although successful in achieving fidelities larger than 0.99 on complicated transitions such as $|0\rangle \rightarrow |3\rangle$ for $N = 7$, was too complicated algorithmically (e.g. the Hessian computation for the KL constraint) and took much longer than the rest.

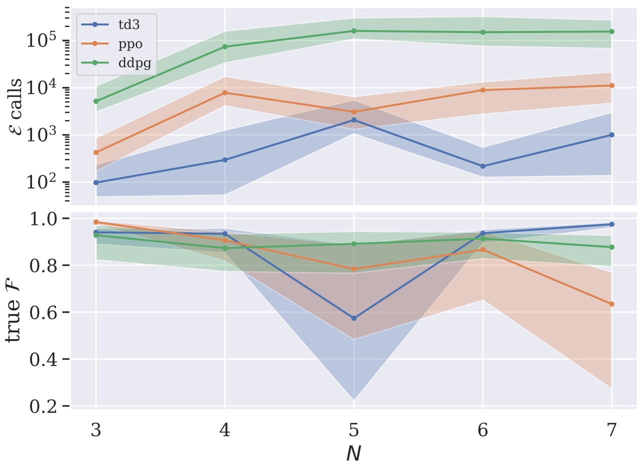

Fig. 1. Top: Cost comparison between PPO, TD3 and DDPG for a transition from $|0\rangle$ to $|2\rangle$ for chains of length $N = 3, \ldots, 7$ with 100 single shot measurements and $\sigma_{\text{noise}} = 0.05$. The algorithms were run 50 times and the median $\mathcal{E}$ calls are plotted with the interquartile range shown to highlight variation. DDPG is worse compared to the others. A perceived fidelity threshold of 0.99 was set as the termination criterion. Bottom: True fidelities for each case. The true fidelities of the controllers generally deteriorate with increasing chain length.

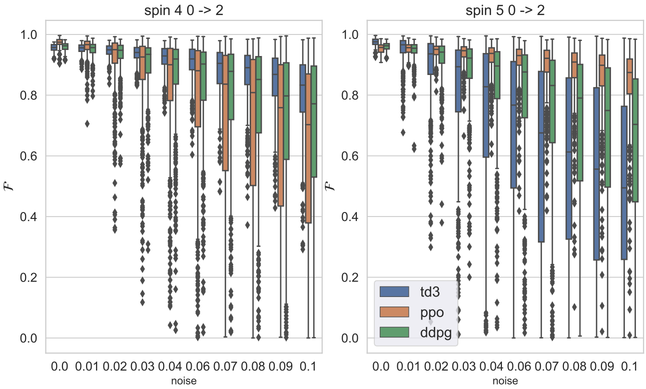

Fig. 2. Robustness analysis for PPO, TD3 and DDPG for a transition from $|0\rangle$ to $|2\rangle$ for the 50 controllers found in Section III-A for chains of length $N = 4, 5$. Ten levels of perturbation noise $\sigma_{\text{noise}} = 0, \ldots, 0.1$ are considered for each controller which is evaluated ten times to yield 500 points per box-plotted fidelity distribution. Left: The increase in interquartile width of the box plots is slowest for TD3 followed by DDPG and then PPO which performs most robustly under this measure. Right: The increase in interquartile width of the box plots is slowest for PPO followed by DDPG and then TD3. Here the length N is chosen to highlight the worst case MCRA for TD3 and for PPO.

B. Robustness of Reinforcement Learning Controllers

The robustness of the controllers found by RL in Section III-A remains unclear and serves as a further criterion to choose a suitable RL algorithm. We conduct a Monte Carlo robustness analysis (MCRA) using variable Hamiltonian perturbation noise σ_{noise} of the 50 controllers computed for each chain length for all three algorithms. For each controller

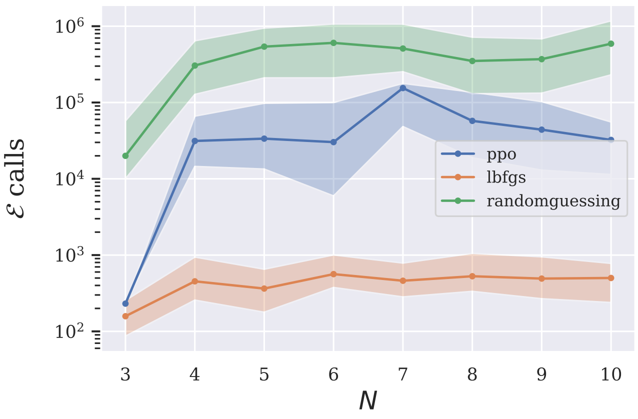

Fig. 3. Comparision between L-BFGS, PPO and randomly guessing controllers for a transition from $|0\rangle$ to $|2\rangle$ for chains of length $N = 3$ to $N = 10$ without noise. The algorithms were run 50 times and the median $\mathcal{E}$ calls are plotted with the interquartile range shown to highlight variation. A threshold of $\mathcal{F} = 0.99$ is set for termination of a run. PPO performance is upper bounded by random guessing and lower bounded by L-BFGS with access to a perfect model.

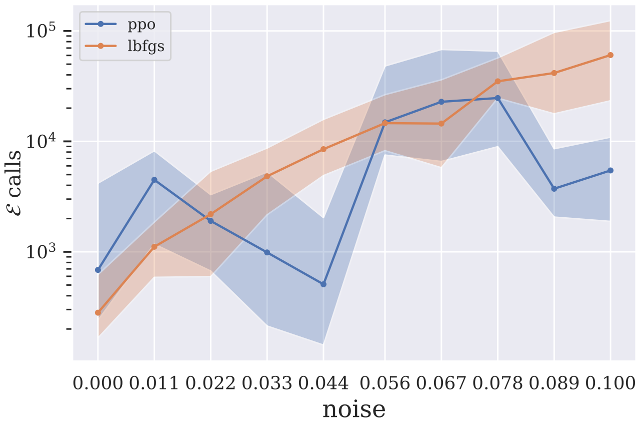

Fig. 4. Number of $\mathcal{E}$ calls comparison between L-BFGS and PPO for a transition from $|0\rangle$ to $|2\rangle$ for a chain of length $N = 4$ as a function of Hamiltonian perturbation noise σ_{noise} with a termination fidelity threshold of 0.98. The algorithms were run 50 times and the median $\mathcal{E}$ calls are plotted with the interquartile range shown to highlight variation. PPO $\mathcal{E}$ calls remain around 10^4. An approximately exponential rise in $\mathcal{E}$ calls for L-BFGS is observed.

$\mathcal{S}_\tau$ found, we perturb the Hamiltonian $\mathcal{H}_s s$ using noise of the same triagonal form with mean 0 and the variance $\sigma^{(i)}_{\text{noise}} = 0.1k/9$, $k = 0, \ldots, 9$. We then evaluate the true fidelities $\mathcal{F}$ of the controller $\mathcal{S}_\tau$ for each level of perturbation without any additional noise. We repeat this ten times for all 50 controllers and combine the results into a single fidelity distribution. This allows us to judge the expected fidelity of the controllers found by the algorithm.

The distributions are represented non-parametrically as 1D box-plots as shown in Fig. 2 (the other cases are similar, but are omitted due to space limitations). This figure highlights that some fidelity distributions are heavy tailed with many outliers, meaning there is significant variation of fidelity between some controllers under perturbation. DDPG controllers, despite making more function calls, were the least robust when it came to preserving the interquartile width of the performance distribution. For PPO vs. TD3, there are cases where TD3 is better than PPO's and vice versa. However, PPO's performance was more consistent compared with TD3's. TD3, similar to REINFORCE and TRPO, showed a high variation in successful termination, getting stuck indefinitely at local minima for some problems, and there were gaps in the collected statistics due to timeouts. So we were only able to collect statistics for some N for some of the cases in Section III-A without rerunning multiple times. On balance, we find that PPO performs most consistently compared to the other RL algorithms for multiple repetitions for different spin transitions. Therefore, we decided to focus on PPO for the comparison with gradient-based optimisation.

C. Cost of PPO vs. L-BFGS

A first step to compare our chosen RL algorithm, PPO, with gradient-based optimisation is to analyse the costs in terms of number of $\mathcal{E}$ calls (see Section III-A) under the noiseless dynamics of the ideal model. For gradient-based optimisation, we use L-BFGS with restarts, which performed well on the studied control problem in earlier work [23].

Fig. 3 shows how function calls scale with the length of the spin chain, $N = 3, \ldots, 10$, for a transition $|0\rangle$ to $|2\rangle$ for PPO, L-BFGS and randomly guessing controllers. The randomly guessed controllers are used to benchmark potential deviations in the computational difficulty of the problem. We stop once a fidelity threshold of 0.99 is crossed. The spin chain transition is computationally similar for all N as it depends largely on the relative distance between the spins, the control and time constraints, which are kept constant for all the problems we study. There is an initial jump from $N = 3$ after which all algorithms manifest a relatively flat increase in the number of function calls as the length of the chain increases. This is likely because transitions in the short 3-chain are easier to achieve as simple Rabi oscillations which are generally trap free, and due to the existence of analytical solutions for this case which are absent for longer chains.

It is not surprising to observe that for an accurate model L-BFGS is mostly two orders of magnitude better than PPO. PPO has to consume most of the calls to build up an internal representation of the model before it can start optimizing.

Adding small stochastic noise to the Hamiltonian should degrade the performance of L-BFGS considerably in terms of the number of function calls. To analyze this, we relax the termination constraint on fidelity to 0.98 and consider only perturbations to H_{ss} without single shot measurement noise. Note that single-shot measurement or perturbation noise renders L-BFGS incapable of estimating fidelities over 0.99 without making many millions of function calls (hence the reduction to 0.98 here). Fig. 4 demonstrates an approximately exponential rise in $\mathcal{E}$ calls for L-BFGS as the strength of the perturbation is increased from $\sigma_{\text{noise}} = 0$ to $\sigma_{\text{noise}} = 0.1$. Clearly Hamiltonian perturbations deteriorate

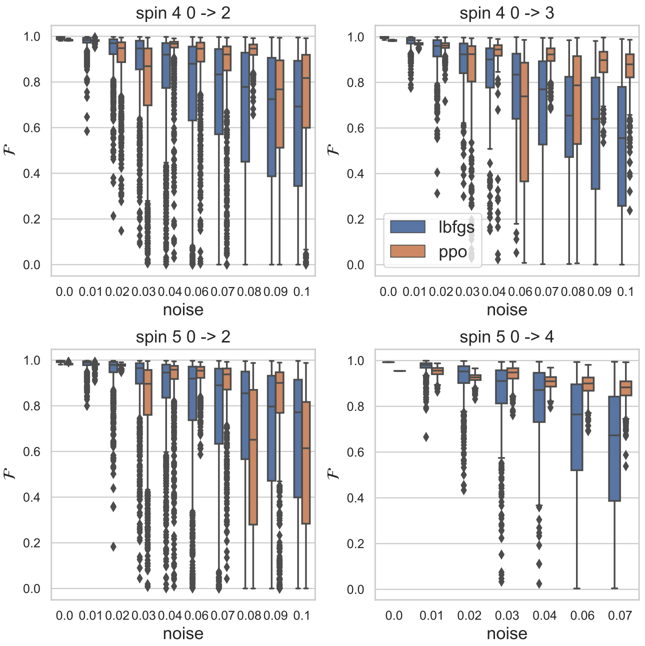

Fig. 5. Comparison of 100 model-based L-BFGS controllers computed without noise and 100 PPO controllers trained only under Hamiltonian perturbation noise σ_{noise} corresponding to the x axis value for transitions to the middle and end of chains of length $N = 4, 5$.

the performance of L-BFGS, while PPO keeps performing on a similar level than without noise. Single shot measurement noise that was considered in Section III-A has not been employed here, as this would have made the task L-BFGS even harder and it has not been designed for noisy optimisation tasks. Overall these results are likely due to high sensitivity of the optimization descent step of L-BFGS to small perturbations in the low rank Hessian components. This causes the number of steps it has to take in the control landscape to steeply increase. It is interesting to note that $\mathcal{E}$ calls go down for PPO from around 10^5 to around 10^4 in Fig. 3, and we observe a similar effect in Fig. 1.

D. Robustness of PPO and L-BFGS Controllers

We conduct an MCRA (see Section III-B) to compare robustness of 100 controllers found by model-based L-BFGS under ideal conditions and model-free PPO under low Hamiltonian perturbation noise. There are two cases worth considering: (1) Robustness of PPO controllers found at different levels of Hamiltonian perturbation; (2) The robustness of PPO controllers w.r.t. Hamiltonian perturbation found at a particular noise level. Both cases are compared to 100 L-BFGS controllers for each transition using the ideal model without noise. The termination condition, in all cases, is $\mathcal{F} \geq 0.99$.

For (1), we consider transitions to the middle and end for $N = 4, 5$, as shown in Fig. 5. We use PPO controllers trained with Hamiltonian perturbation noise σ_{noise} that corresponds to the noise level on the x axis from 0.01 to 0.1. We find, as expected, that the width of the fidelity distribution for L-BFGS controllers slowly increases as σ_{noise} is increased from 0 to 0.1. The expected fidelity is further dropping from being concentrated around $\mathcal{F} = 0.99$ to a very flat width and increasingly heavier tail, down to $\mathcal{F} = 0$. For PPO controllers, however, we observe that at certain noise levels, e.g., $\sigma_{\text{noise}} = 0.01, 0.04, 0.07$, the controllers found for all problems have narrow distributions compared with L-BFGS. At other noise levels, e.g., $\sigma_{\text{noise}} = 0.08, 0.1$ for $N = 5, |0\rangle$ to $|2\rangle$, they have wider distributions for some problem, but also narrow distributions for others, e.g., $\sigma_{\text{noise}} = 0.08, 0.1$ for $N = 4, |0\rangle$ to $|2\rangle$. We conjecture that added structured perturbations may have a smoothing effect on the optimization landscape which would result in either filtration or creation of "barriers" near optima in some cases.

For (2), we consider in addition to the cases of (1), also transitions to the middle for $N = 6, 7$. Here the PPO controllers have been computed for low Hamiltonian perturbation noise $\sigma_{\text{noise}} = 0.01$. Both the L-BFGS controllers and the PPO controllers become worse with increasing noise levels. However, the PPO controllers drop off slower, except in the case of $N = 6$, $|0\rangle$ to $|3\rangle$. This suggests that overall PPO is more likely to find robust controllers.

To investigate this further, the performance of a well-performing PPO and L-BFGS controller for the $N = 5$, $|0\rangle$ to $|4\rangle$ transition is compared. For each algorithm, out of the 100 controllers found within the set of controllers for (2), we select the one with the highest median fidelity across the ten noise levels to account for the heavy tail nature of the performance distribution. Then, for each controller, the Hamiltonian is perturbed as $H_{ss} + \delta P$ where P is the perturbation direction and δ its strength. P is sampled uniformly on a $9D$ Euclidean sphere, created by the five perturbation for Δ_n and a further four for the coupling strengths. The fidelity was computed along these directions for a perturbation strength from -0.1 to 0.1. The density of the curves is estimated at specific perturbation strengths and plotted as a violin plot. Results are shown in Fig. 7. The PPO controller is clearly not at a maximum of the fidelity, so some perturbations have a better chance to improve the fidelity. The L-BFGS controller is at a fidelity maximum, which means that most perturbation directions, including those on the couplings which are not control parameters, reduce the fidelity. Similar behaviour has been observed for other controllers.

IV. DISCUSSION AND CONCLUSION

Our main finding is that policy gradient RL methods allow nonparametric constructions of optimization models even under highly noisy conditions as seen in Section III-A where pure model-based methods perform poorly as seen in Section III-C. We have quantified costs in terms of the number of function or environment calls. In the absence of noise, RL performance is lower bounded by model-based optimisation and upper bounded by pure random guessing. This implies that a nonparametric model is being constructed. The cost of model construction is relatively bounded by random guessing for RL under noisy conditions. However, the number of calls

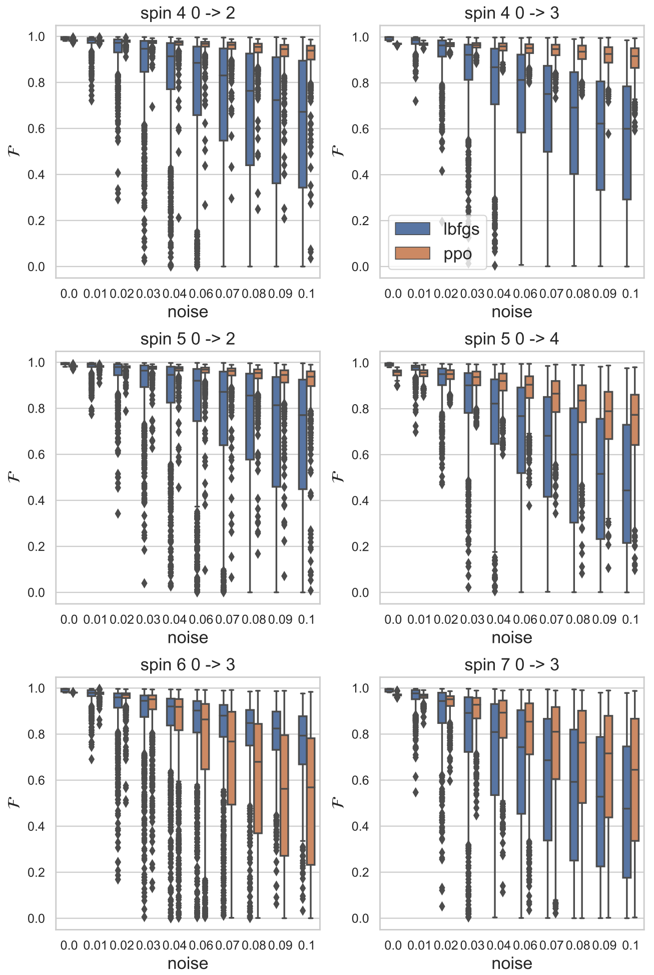

Fig. 6. Comparison of controllers found by model-based L-BFGS without noise and PPO trained under low Hamiltonian perturbation noise $\sigma_{\text{noise}} = 0.01$ and perfect measurements. We consider transitions to the middle and end of chains of length $N = 4, 5$ and to the middle of $N = 6, 7$.

is still high. Model-based RL or Bayesian methods could be explored to reduce the reliance on information acquisition.

In Section III-B, a Monte Carlo robustness analysis and consistency of PPO for variations of the energy landscape control problem is used to motivate our choice of PPO for comparison with L-BFGS with restarts to understand robustness of controllers found by RL. We demonstrate that RL controllers found under low Hamiltonian perturbation noise levels are typically more robust compared with those found by L-BFGS but there is variation within the quality of their robustness that needs to be explored more as a function of their clustering and correlation of locations in the optimization landscape. It appears that in some cases RL finds controllers that may not be optimal for the ideal model, but perform robustly at high fidelity under noisy conditions. This suggests that Hamiltonian noise in particular can improve robustness of some controllers. RL is a promising avenue for feedback adaptive control with less overhead compared with variational methods and is arguably comparatively better with uncertainties. However, a careful construction of the control problem in an RL paradigm is needed before its application.

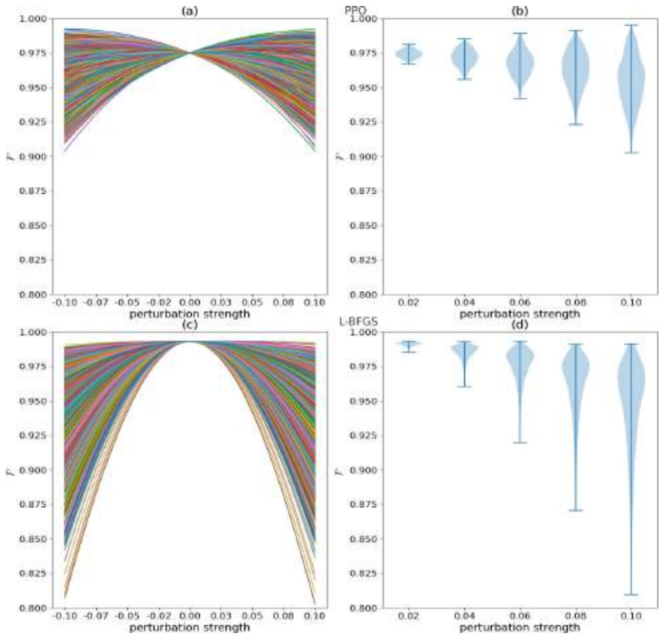

Fig. 7. Robustness comparison of a well performing PPO (top) and L-BFGS (bottom) controller for $N = 5$, $|0\rangle$ to $|4\rangle$. The left plots (a) and (c) show $1,000$ overlayed fidelity curves, sampled along different Hamiltonian perturbation directions. The right plots (b) and (d) show the density distribution of these curves at specific perturbation strengths, sampling the fidelity at a given strength uniformly over the perturbation directions.

V. Acknowledgments

I. Khalid is supported by a Cardiff University PhD scholarship.

References

[1] J Majer, JM Chow, JM Gambetta, Jens Koch, BR Johnson, JA Schreier, L Frunzio, DI Schuster, Andrew Addison Houck, Andreas Wallraff, et al. Coupling superconducting qubits via a cavity bus. *Nature*, 449(7161):443–447, 2007.

[2] J. Q. You and Franco Nori. Quantum information processing with superconducting qubits in a microwave field. *Phys. Rev. B*, 68:064509, Aug 2003.

[3] S. A. Wolf, D. D. Awschalom, R. A. Buhrman, J. M. Daughton, S. von Molnár, M. L. Roukes, A. Y. Chtchelkanova, and D. M. Treger. Spintronics: A spin-based electronics vision for the future. *Science*, 294(5546):1488–1495, 2001.

[4] J. I. Cirac and P. Zoller. Quantum computations with cold trapped ions. *Phys. Rev. Lett.*, 74:4091–4094, May 1995.

[5] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, Aug 2018.

[6] Immanuel Bloch, Jean Dalibard, and Sylvain Nascimbene. Quantum simulations with ultracold quantum gases. *Nature Physics*, 8(4):267–276, 2012.

[7] Christian Gross and Immanuel Bloch. Quantum simulations with ultracold atoms in optical lattices. *Science*, 357(6355):995–1001, 2017.

[8] Immanuel Bloch. Quantum simulations come of age. *Nature Physics*, 14(12):1159–1161, 2018.

[9] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[10] Marin Bukov, Alexandre G. R. Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, and Pankaj Mehta. Reinforcement learning in different phases of quantum control. *Phys. Rev. X*, 8:031086, Sep 2018.

[11] AA Feldbaum. Dual control theory. i. *Avtomatika i Telemekhanika*, 21(9):1240–1249, 1960.

[12] AA Feldbaum. Dual control theory. ii. *Avtomatika i Telemekhanika*, 21(11):1453–1464, 1960.

[13] Dimitri P Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific Belmont, MA, 2019.

[14] Richard Bellman. The theory of dynamic programming. Technical report, Rand corp santa monica ca, 1954.

[15] Navin Khaneja, Timo Reiss, Cindie Kehlet, Thomas Schulte-Herbrüggen, and Steffen J Glaser. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms. *Journal of magnetic resonance*, 172(2):296–305, 2005.

[16] Vadim Krotov. *Global methods in optimal control theory*, volume 195. CRC Press, 1995.

[17] Mogens Dalgaard, Felix Motzoi, Jens Jakob Sørensen, and Jacob Sherson. Global optimization of quantum dynamics with alphazero deep exploration. *npj Quantum Information*, 6(1):1–9, 2020.

[18] Riccardo Porotti, Dario Tamascelli, Marcello Restelli, and Enrico Prati. Coherent transport of quantum states by deep reinforcement learning. *Communications Physics*, 2(1):1–9, 2019.

[19] Murphy Yuezhen Niu, Sergio Boixo, Vadim N Smelyanskiy, and Hartmut Neven. Universal quantum control through deep reinforcement learning. *npj Quantum Information*, 5(1):1–8, 2019.

[20] X. Wang, P. Pemberton-Ross, and S. G. Schirmer. Symmetry and subspace controllability for spin networks with a single-node control. *IEEE Transactions on Automatic Control*, 57(8):1945–1956, 2012.

[21] Steffen J Glaser, Ugo Boscain, Tommaso Calarco, Christiane P Koch, Walter Köckenberger, Ronnie Kosloff, Ilya Kuprov, Burkhard Luy, Sophie Schirmer, Thomas Schulte-Herbrüggen, et al. Training schrödinger's cat: quantum optimal control. *The European Physical Journal D*, 69(12):1–24, 2015.

[22] Daoyi Dong and Ian R Petersen. Quantum control theory and applications: a survey. *IET Control Theory & Applications*, 4(12):2651–2671, 2010.

[23] F. C. Langbein, S. Schirmer, and E. Jonckheere. Time optimal information transfer in spintronics networks. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 6454–6459, 2015.

[24] S.G. Schirmer, E. Jonckheere, and F.C. Langbein. Design of feedback control laws for spintronics networks. *IEEE Transactions on Automatic Control*, 63(8):2523–2536, 2018.

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

[26] Raul Rojas. The backpropagation algorithm. In *Neural networks*, pages 149–182. Springer, 1996.

[27] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[29] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[30] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pages 1587–1596. PMLR, 2018.

[31] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pages 387–395. PMLR, 2014.

[32] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[33] Richard H Byrd, Jean Charles Gilbert, and Jorge Nocedal. A trust region method based on interior point techniques for nonlinear programming. *Mathematical programming*, 89(1):149–185, 2000.

[34] Jorge Nocedal and Ya-xiang Yuan. Combining trust region and line search techniques. In *Advances in nonlinear programming*, pages 153–175. Springer, 1998.