IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 January 2025, accepted 28 February 2025, date of publication 14 March 2025, date of current version 25 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3551232

RESEARCH ARTICLE

Advanced Quantum Control With Ensemble Reinforcement Learning: A Case Study on the XY Spin Chain

FARSHAD RAHIMI GHASHGHAEI[1], NEBRASE ELMRABIT[2], AYYAZ-UL-HAQ QURESHI[3], ADNAN AKHUNZADA[4], (Senior Member, IEEE), AND MEHDI YOUSEFI[5]

[1]School of Computing and Digital Technology, Birmingham City University, B4 7XG Birmingham, U.K.
[2]College of Computing and Information Technology, Ministry of Technical and Vocational Education, Zawia, Libya
[3]Department of Cyber Security and Networks, Glasgow Caledonian University, G4 0BA Glasgow, U.K.
[4]College of Computing and Information Technology, University of Doha for Science and Technology, Doha, Qatar
[5]Independent Researcher, G5 8EH Glasgow, U.K.

Corresponding authors: Adnan Akhunzada (Adnan.adnan@udst.edu.qa) and Nebrase Elmrabit (nelmrabit@gmail.com)

This work was supported by Qatar National Library (Open Access Funding).

ABSTRACT This research presents an ensemble Reinforcement Learning (RL) approach that combines Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) algorithms to tackle quantum control problems. This research aims to use the complementary strengths of DQN and PPO algorithms to develop robust and adaptive control policies for noisy and uncertain quantum systems. We comprehensively analyse the proposed ensemble learning, including algorithmic details, implementation specifics, and experimental results. Through extensive experimentation and evaluation, we demonstrate the effectiveness of the ensemble approach in learning control strategies for manipulating quantum systems towards a random target state. The results highlight the potential of ensemble RL techniques in addressing the challenges of quantum control tasks, such as system noise and dynamics. By integrating multiple RL agents within an ensemble framework, We aim to advance current developments in quantum control and create a new path for the development of adaptive control systems for quantum systems. The performance of the ensemble model is assessed against Gradient Ascent Pulse Engineering (GRAPE) and robust Model Predictive Control (MPC) to demonstrate its efficiency in highly challenging and noisy environments.

INDEX TERMS Adaptive control, deep Q-network (DQN), ensemble learning, proximal policy optimization (PPO), quantum control, reinforcement learning (RL).

I. INTRODUCTION

Quantum systems have the potential to change fields such as materials science, drug discovery, and cryptography [1], [2] [3], [4]. Quantum systems demonstrate complex behaviours according to quantum mechanics laws that open up many opportunities for applications such as quantum computing, quantum communication, and quantum sensing. However, Harnessing the power of quantum systems requires careful control, which can be difficult due to noise, uncertainty, and insufficient understanding of complicated quantum dynamics [5].

In recent years, machine learning techniques, particularly Reinforcement Learning (RL), have appeared promising approaches for tackling quantum control problems [6], [7] [8]. RL provides a framework for learning control policies through interaction with the quantum system environment, making it suitable for dynamic and uncertain quantum systems [9]. Using RL algorithms, researchers aim to develop adaptive control strategies capable of navigating the complex landscape of quantum dynamics. These strategies can potentially improve quantum technology by providing efficient and robust manipulation of quantum systems for various applications.

This paper evaluates the synergy between ensemble RL and quantum control, aiming to develop robust and

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Xu.

adaptive control strategies for quantum systems. Ensemble learning, which combines multiple learning algorithms to improve performance and robustness, provides a compelling framework for addressing the challenges of quantum control. By integrating Deep Q-Network (DQN) and Proximal Policy Optimization (PPO) into an ensemble framework, we use their complementary strengths to enhance the adaptability of control policies. DQN is a value-based algorithm that effectively estimates action values in discrete action spaces, while PPO is a policy-based approach that optimises policies directly, making it suitable for both discrete and continuous actions. DQN can struggle with instability and exploration efficiency, whereas PPO uses a clipped objective for stable updates and better exploration strategies. By integrating DQN's value estimation with PPO's policy optimisation, this hybrid approach can enhance learning efficiency and performance, allowing for more robust solutions in complex reinforcement learning environments.

The effectiveness of the DQN in discrete action spaces and the strength of the PPO in continuous action spaces combine to create a more resilient and flexible control strategy. This approach mitigates individual algorithms' weaknesses, enhancing the control system's overall robustness. Furthermore, the ensemble method addresses critical challenges in quantum control, such as handling high-dimensional state spaces and dealing with uncertainties and noise. Our research demonstrates the practical benefits of ensemble RL through case studies and experimental results, showcasing its potential to achieve high-fidelity control and reduce error rates. This work aims to advance quantum technologies by providing a more reliable control methodology.

This study's primary aim is to investigate the feasibility and effectiveness of ensemble RL agents in quantum control tasks. The first contribution of this research involves the development of an ensemble RL approach that combines DQN and PPO agents to tackle quantum control problems. Secondly, it comprehensively analyses the proposed ensemble learning, including algorithmic details, implementation specifics, and experimental results. Thirdly, through extensive experimentation and evaluation, the study demonstrates the effectiveness of the ensemble approach as a learning control for the XY spin chain quantum system to maximize the fidelity between a random evolved quantum state and a random target state. Additionally, the research assesses the performance of the ensemble model against GRAPE and robust MPC to show its efficiency in highly challenging environments. Lastly, the study offers new paths for using several potential techniques to improve quantum control strategies.

The main contributions of this paper are outlined as follows:

- Introducing an innovative control method that integrates DQN and PPO agents to address the complexities of quantum control and enhance the precision and robustness of control strategy.
- Providing a thorough examination of the ensemble learning method, including detailed algorithmic explanations, implementation specifics, and a broad set of experimental results among the ensemble agent and isolated agents.
- Demonstrating the ability of ensemble learning method to maximize fidelity between evolved and target quantum states and outperform other methods such as GRAPE and robust MPC in challenging environments.

The rest of this paper is structured as follows. Section II will provide a comprehensive literature review. Section III will detail the methodology employed in this paper. Section IV will present and analyse the results obtained from the experiments with a discussion about the findings and the limitations of this research. Finally, Section V will conclude the paper, summarising key insights and outlining new paths for future research.

II. BACKGROUND AND RELATED WORK

This section comprehensively overviews foundational concepts and related works on quantum control and RL. It covers the main strategies of quantum control, including a description and limitations of each technique used to manage quantum systems. Additionally, the section explores RL and its applications in control, presents the theoretical justification for the ensemble learning method, and details the specific RL algorithms employed in this research.

A. QUANTUM CONTROL AND CORE STRATEGIES

Quantum control refers to the manipulation of quantum systems to achieve specific objectives, such as preparing quantum states or executing quantum operations. In a quantum computer, a quantum bit (qubit) can exist in a superposition of states, representing both 0 and 1 simultaneously. To direct a qubit to a specific state, such as 1, one can apply precise adjustments to its environment, including variations in magnetic fields or other interactions. This controlled manipulation is essential for ensuring that the qubit accurately reflects the desired state, facilitating reliable quantum computations.

As illustrated in Figure 1, three main strategies exist for quantum control: optimal control, robust control, and learning control. Optimal Control, such as the GRAPE, designs control pulses based on a perfect system model, aiming for the best possible performance under ideal conditions [10]. This approach is based on optimal control theory principles and uses mathematical optimisation techniques to identify the control pulses that minimise a particular cost function [11]. In the context of quantum control, the cost function often represents the infidelity of the desired state transfer or gate operation. The goal is to adjust control parameters in such a way that this infidelity is minimized. While optimal control methods may achieve outstanding results in principle, their efficacy depends on an ideal quantum system model.

Real-world quantum computers are typically noisy and complex due to model errors, Hamiltonian uncertainty, and

environmental impacts [12]. These issues can potentially cause poor performance by introducing inconsistencies between the ideal model used for control design and the actual behaviour of the quantum system. Moreover, these systems often drift over time, challenging control methods. This refers to the fact that the properties of quantum systems can change over time due to various factors, such as environmental influences or hardware degradation. This drift can further complicate the task of quantum control, as it means that the optimal control pulses may need to be continuously updated to account for these changes.

On the other hand, robust control, such as robust MPC, focuses on stability and good performance under uncertainty, reducing the optimality for reliability. This strategy is well-suited for environments where the system's behaviour is uncertain or accurate models are unavailable [13]. However, robust control systems have their own challenges from the drift that happens in quantum devices over time. To avoid this drift, controls must be updated, and new effective controls have to be identified. The updates require significant computational resources and may introduce complexity into the control system [14]. Moreover, the effectiveness of these updates may diminish over time as quantum devices evolve and drift occurs. Exploring new techniques and methodologies requires more experimentation and validation, which can consume much time and resources.

Another approach is Learning Control, which optimises control even with incomplete system models [15]. This approach uses machine learning techniques to continuously improve the control strategy based on feedback from the system. There are two main approaches within learning control: stochastic optimization and RL. Stochastic optimization finds robust control pulses amidst noise but may have limited exploration capabilities [16]. This approach uses random search methods to explore the control space and find control pulses that are robust to noise. RL, on the other hand, employs an agent to explore control options and learn through trial and error [5]. The agent interacts with the quantum system and receives rewards based on the system's response, which allows it to learn an optimal control strategy over time.

B. REINFORCEMENT LEARNING

As mentioned earlier, reinforcement learning is a type of machine learning in which an agent learns through trial and error in a given environment in order maximize rewards. For example, training a robot (RL agent) to navigate a maze (environment). Each time the robot makes a move (action), it receives feedback which would be a reward for correct moves. Over time, it learns the best actions to reach the end of the maze with the highest reward possible. Instead of depending on a pre-existing dataset, reinforcement learning creates data through agent-environment interactions. The effectiveness is demonstrated by the evaluation, which is based on training performance, rewards trends, and policy updates. The agent's goal is to improve its decisions based on past experiences to perform better. RL has shown promise in tackling complex challenges, particularly in quantum control. However, it does come with its own set of practical limitations. RL often requires substantial data collected through repeated interactions with the environment [17]. This data-intensive nature of RL can pose challenges in scenarios where data is rare or expensive to generate. Moreover, the performance of RL in achieving desired outcomes may not always be optimal. Traditional RL algorithms often require vast data and computational power to achieve optimal performance. This can make RL computationally expensive and time-consuming, particularly for complex tasks with high-dimensional state and action spaces [18].

To address these challenges, researchers have explored the use of ensemble methods in RL. Ensemble methods combine multiple RL agents with diverse exploration-exploitation strategies [19], [20]. This approach offers improved generalization and stability in learning control policies. Recent studies have showcased the effectiveness of ensemble RL approaches in quantum control scenarios, demonstrating their ability to learn adaptive and robust control strategies for complex quantum systems.

C. MARKOV DECISION PROCESS

A Markov Decision Process (MDP) is a mathematical foundation to determine the optimal policy in the framework of uncertainty through probabilistic state transitions [21], [22]. This provides a theoretical framework for using RL agents to interact with quantum systems and learn from them. Formally defined MDPs include a set of states S, a set of available actions A, a model describing transition probabilities P, and rewards R. Each state represents a possible configuration of the quantum system; each action represents a decision or control input that can change the state.

Transition probabilities describe the stochastic behaviour of a system in that they define the probability of going from one state to another after applying a certain action [23]. The reward is numerical notations given at every transition, representing the immediate benefit or cost associated with those transitions. The goal of an MDP is to find a policy that is a strategy to choose current actions based on the state in order to maximize an expected sum of rewards over time [24].

Therefore, MDPs offer a way of modelling through the learning of the agent's interaction with the environment how to decide, the observation of the consequences of such interaction, and improvement of the policy guided by the rewards achieved. Figure 2 illustrates how an RL algorithm can learn by interacting with the environment through the MDP framework.

D. THEORETICAL JUSTIFICATION FOR USING ENSEMBLE RL

The studies by Khalid et al. [6] and Giannelli et al. [25] provide valuable insights into applying RL techniques in

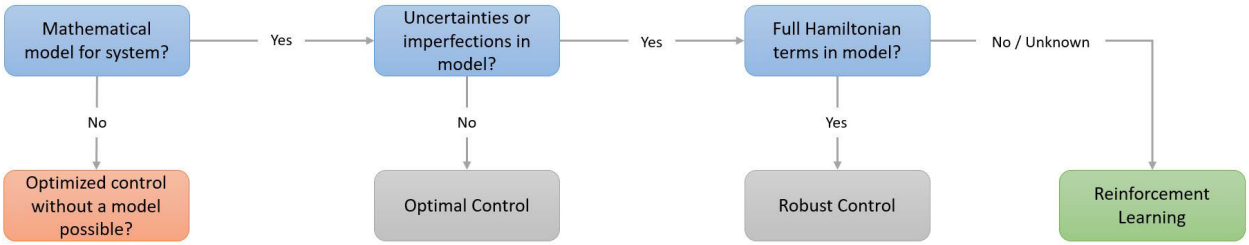

FIGURE 1. Decision for choosing reinforcement learning.

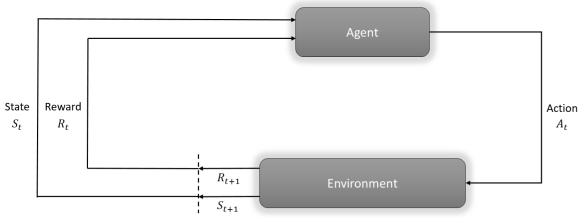

FIGURE 2. Reinforcement learning process cycle.

solving complex quantum control problems. The first study compares RL methods with gradient-based optimisation methods for robust energy landscape control of spin-1/2 quantum networks. It demonstrates that RL controllers, particularly those found under low Hamiltonian perturbation noise levels, tend to be more robust than L-BFGS controllers. This robustness is crucial, especially in quantum systems susceptible to noise, and RL methods like PPO show resilience in finding optimal controllers even in the presence of noise. Similarly, the second study applies RL to achieve efficient population transfer in a three-level quantum system, emphasising achieving results similar to the STIRAP process. The RL approach results in developing a model that efficiently achieves the desired population transfer, even in the presence of noise. However, both studies identify some limitations and areas for optimisation within the RL approach, including computational time, the high number of free parameters, and the lack of hyperparameter optimisation.

Our ensemble learning approach combines the strengths of DQN and PPO to produce a more adaptable and adjustable control framework. This method improves the control system's ability to manage high-dimensional state spaces and various noise levels, resulting in greater dependability and effective quantum control. Furthermore, shifting computational load across multiple algorithms may considerably decrease computational effort and improve the efficiency of learning optimal solutions. This integrated strategy overcomes the highlighted limitations and provides a more robust and scalable solution.

E. REINFORCEMENT LEARNING ALGORITHMS: DQN AND PPO

DQN is an advancement of Q-Learning, a RL algorithm that doesn't rely on a model. The term "Q" in Q-Learning signifies "quality", which is a metric of an action's effectiveness in maximizing future rewards. Traditional Q-Learning employs a table to store and update these Q-values. However, this approach becomes unfeasible when dealing with a large number of states and actions [26]. This is where DQN comes into play. DQN employs a deep neural network to approximate the Q-function. It also enables the algorithm to learn from high-dimensional inputs such as images or sensor data [27]. The process commences with the initialization of the Q-network. This network takes the state of the environment as input and outputs the Q-value for each possible action. The agent then interacts with the environment. It selects an action based on the current policy, which is usually epsilon-greedy. After performing the action, the agent observes the next state and the reward from the environment.

The Q-network is then updated based on the observed reward and the maximum predicted Q-value for the next state [28]. This update is done using a loss function, typically the mean squared error between the predicted Q-value and the observed reward plus the discounted maximum predicted Q-value for the next state [29].

A key innovation in DQN is the use of a technique called Experience Replay. This involves storing the agent experiences and then randomly sampling from this memory to update the network. This helps to break the correlation between consecutive experiences and stabilize the learning process. Deep reinforcement learning (DRL) has proven to be a powerful tool for crafting optimal strategies across diverse complex systems, particularly when there is no prior insight into the control landscape. In [30], the authors presented a DRL-based technique that allows for immediate and precise control of quantum systems, leading to significant improvements in control accuracy and high fidelity. This suggests that DQN, as a DRL algorithm, could be a powerful tool for quantum control tasks, capable of handling high-dimensional state and action spaces.

Unlike DQN, which focuses on value estimation, Proximal Policy Optimisation (PPO) emphasizes policy optimisation. This method has gained popularity due to its stability and efficiency in training RL agents, especially in continuous action spaces [31]. PPO is a type of policy optimisation method that seeks to find the best policy by minimizing the difference between the new and old policy through a novel

objective function. The main idea of PPO is to avoid having too large a policy update [32].

To do that, PPO uses a ratio that tells us the difference between the new and old policy and clips this ratio from a specific range $[1-\epsilon, 1+\epsilon]$ where ϵ is a small positive number that controls the balance between exploration and exploitation during training [33]. Doing that will ensure that the policy update will not be too large. The PPO algorithm starts with the initialization of the policy network. This network takes the state of the environment as input and outputs the policy for each possible action [34]. The agent then interacts with the environment. It selects an action based on the current policy, which is usually epsilon-greed. After performing the action, the agent observes the next state and the reward from the environment. The policy network is then updated based on the observed reward and the maximum predicted Q-value for the next state.

Unlike DQN where a Q-value function is learned, PPO typically uses a function approximator (like a neural network) to directly learn the policy [34]. This function takes the state as input and outputs the probabilities of each action according to the policy. The Q-values are then implicitly defined by this policy. This update is done using a loss function, typically the mean squared error between the predicted Q-value and the observed reward plus the discounted maximum predicted Q-value for the next state.

In the context of quantum control, PPO has been investigated for fine-tuning control parameters for quantum gates with high precision. This suggests that PPO could be a valuable tool for quantum control tasks, striking a balance between exploration and exploitation and facilitating efficient learning in continuous action spaces.

III. METHODOLOGY

This section explains the methodology used to achieve effective quantum control of a spin chain system using our ensemble learning approach. We integrate the quantum system's mathematical framework and the DQN and PPO algorithms to maximize the fidelity between the evolved quantum state and the random target state.

A. QUANTUM SYSTEM

Our chosen spin chain system consists of a chain of N identical spins. The Hamiltonian (H^{dyn}) of the system captures its total energy and dictates how the state evolves over time. As described by Zhang et al. [35], it is formulated considering two key interactions: the coupling strength (J) between neighbouring spins and the influence of external magnetic fields (B_n) at each spin site. The mathematical form of the Hamiltonian is presented as follows:

$$H^{dyn}(t) = \frac{J}{2}\sum_{n=1}^{N-1}\left(\sigma_n^x\sigma_{n+1}^x + \sigma_n^y\sigma_{n+1}^y\right) + \sum_{n=1}^{N} B_n(t)\sigma_n^z \quad (1)$$

where J is the coupling strength, $B_n(t)$ represents the time-dependent control acting as an external magnetic field for the n^{th} spin, which takes binary values (0,B), and σ_n^x, σ_n^y, and σ_n^z are the Pauli matrices for the n^{th} spin.

In order to simulate real-world conditions, a Gaussian noise model is used on the external magnetic fields. The noise model is defined as follows:

$$B_n^{\text{noisy}}(t) = B_n(t) + \sigma_{\text{noisy}} \quad (2)$$

where σ_{noisy} represents the noise drawn from a normal distribution with a mean of 0 and a standard deviation specified by the noise level.

For the time evolution of the quantum state (ψ), we use the time-dependent Schrödinger equation:

$$i\hbar\frac{d}{dt}\psi(t) = H^{dyn}(t)\psi(t) \quad (3)$$

where $\hbar$ is the reduced Planck's constant, set to 1 for simplicity. The solution to this equation for a small time step (Δt) is given by the unitary time evolution operator (U):

$$U = e^{-iH^{dyn}\Delta t/\hbar} \quad (4)$$

We compute U using the matrix exponential function and update the state (ψ) as follows:

$$\psi(t+\Delta t) = U\psi(t) \quad (5)$$

A crucial aspect of our approach is measuring the closeness between the evolved quantum state (ψ) and the desired target state (ψ_{target}). Fidelity is a metric used for this purpose. It is defined as:

$$F(\psi, \psi_{\text{target}}) = |\langle\psi_{\text{target}}|\psi\rangle|^2 \quad (6)$$

Maximizing fidelity ensures that the evolved state closely approximates the desired target state, which is essential for various quantum control applications.

B. REINFORCEMENT LEARNING ALGORITHMS

To achieve effective control over this spin chain system, DQN and PPO were implemented as two separate agents. These algorithms are selected for their strengths in handling discrete and continuous action spaces.

1) DEEP Q-NETWORK (DQN)

The DQN algorithm represents a confluence of Q-Learning and deep neural networks to address the challenges of RL in high-dimensional spaces. At its core, DQN modifies the traditional Q-Learning update rule to integrate the representational power of deep learning. The updated rule is given by:

$$Q^{new}(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma\max_a Q(s_{t+1}, a) - Q(s_t, a_t)\right] \quad (7)$$

In this equation, s_t denotes the present state, a_t the action undertaken, r_{t+1} the consequent reward, α the learning rate, γ the discount factor, and $\max_a Q(s_{t+1}, a)$ the maximal future reward estimated across all possible actions.

One of the features of DQN is the employment of experience replay, which mitigates the issue of correlated training samples by randomly sampling from a pool of stored experiences $(s_t, a_t, r_{t+1}, s_{t+1})$. This approach diversifies the training data and allows for more efficient use of previous experiences. Another significant feature of DQN is the utilization of a separate target network to stabilize the training process. This network provides a stationary target for the $\max_a Q(s_{t+1}, a)$ term during the update of Q values, and its parameters θ^- are updated less frequently than those of the primary network θ.

The neural network, parameterized by θ, serves as a function approximator for the Q-function. It accepts the state as input and outputs the predicted Q-values for all actions. The network is trained to minimize the discrepancy between the predicted Q-values and the target Q-values, which is captured by the loss function:

$$L(\theta) = \mathbb{E}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta^-) - Q(s, a; \theta)\right)^2\right] \quad (8)$$

Here, s' and a' represent the subsequent state and action. The optimisation of the network parameters θ is achieved through gradient descent on this loss function. DQN uses several tactics to improve learning stability and efficiency:

- **State representation**: The quantum state ψ is represented by its real and imaginary components, forming the input to the neural network.
- **Action selection**: An epsilon-greedy strategy is used, where the agent selects actions based on Q-values with a probability of $1 - \epsilon$ and explores random actions with a probability of ϵ.
- **Experience replay**: Transitions are stored in a replay buffer, and mini-batches are sampled for training. This stabilizes learning by breaking the correlation between consecutive experiences.
- **Target network**: A separate target network provides stable target Q-values, updated periodically to match the main network.

2) PROXIMAL POLICY OPTIMISATION (PPO)

The PPO algorithm is designed to optimise policies while ensuring stable and efficient learning. The main components of PPO in our implementation are as follows.

The PPO agent employs a neural network to approximate both the policy (action probabilities) and the value function (state values). The network takes the state as input and outputs the action probabilities and the state value. The agent selects actions based on the probabilities output by the policy network. The action a_t at state s_t is sampled from the policy $\pi_\theta(a|s)$. Advantages are computed using Generalized Advantage Estimation (GAE). The advantage A_t is calculated as:

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + (\gamma\lambda)^2\delta_{t+2} + \cdots \quad (9)$$

where

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \quad (10)$$

$V(s_t)$ is the value function estimate at time t, γ is the discount factor, and λ is a parameter controlling the bias-variance tradeoff.

The PPO algorithm optimises a clipped surrogate objective function to ensure the new policy does not deviate significantly from the old one. The objective function is:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t\left[\min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t, \right.\right. \\ \left.\left. \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right] \quad (11)$$

Here, $\pi_\theta(a_t|s_t)$ is the new policy probability, $\pi_{\theta_{old}}(a_t|s_t)$ is the old policy probability, $\hat{A}_t$ is the advantage estimate, and ϵ is the clipping parameter.

The value function loss is computed as the mean squared error between the estimated value and the target value:

$$L^{VF}(\theta) = \hat{\mathbb{E}}_t\left[(r_t + \gamma V(s_{t+1}) - V(s_t))^2\right] \quad (12)$$

An entropy term is added to the loss to encourage exploration by ensuring the policy does not become too deterministic:

$$L^S(\theta) = -\hat{\mathbb{E}}_t\left[\pi_\theta(a_t|s_t)\log\pi_\theta(a_t|s_t)\right] \quad (13)$$

The total loss function combines the clipped surrogate objective, value function loss, and entropy bonus:

$$L(\theta) = L^{CLIP}(\theta) + c_1 L^{VF}(\theta) - c_2 L^S(\theta) \quad (14)$$

where c_1 and c_2 are coefficients that balance the contributions of the value loss and the entropy bonus.

The parameters θ are updated using gradient descent on the total loss:

$$\theta \leftarrow \theta - \alpha\nabla_\theta L(\theta) \quad (15)$$

where α is the learning rate.

This comprehensive approach ensures that the policy updates are both efficient and stable, leading to improved performance in complex environments.

PPO incorporates several key techniques:

- **State representation**: The state is represented by the real and imaginary components of ψ, similar to DQN.
- **Policy and value networks**: PPO employs separate neural networks to approximate the policy (action probabilities) and the value function.
- **Advantage estimation**: Advantages are computed using GAE, reducing variance and improving learning stability.
- **Clipped objective**: PPO uses a clipped surrogate objective to prevent large updates, ensuring stable and efficient learning.

Both DQN and PPO employ Temporal Difference (TD) learning as a foundation for value updates. In DQN, the TD error drives the Q-value updates through bootstrapping,

allowing the algorithm to learn from experience efficiently. PPO uses TD learning for estimating the value function, contributing to the computation of advantages through GAE, which smooths out learning by TD steps.

C. ENSEMBLE LEARNING

To enhance the robustness and performance of our quantum control system, an ensemble agent is employed. This agent combines predictions from both agents trained with different algorithms and parameters. The ensemble agent merges the suggestions made by isolated DQN and PPO agents and uses their strategies to make more mindful decisions. The proposed algorithm incorporates elements of both on-policy and off-policy methods. DQN operates off-policy, utilizing a replay buffer to learn from past experiences, independent of the current policy. In contrast, PPO operates on-policy, as it updates its policy based on actions taken from the most recent states encountered. This hybrid approach that combines DQN and PPO, serves as a robust learning method for complex environments. By leveraging both on-policy and off-policy strategies, this ensemble method adapts to a wide range of scenarios, potentially enhancing stability, flexibility, and overall performance.

1) ACTION SELECTION MECHANISM

The ensemble agent selects actions by combining the decisions of the DQN and PPO agents based on a weighted average strategy. When the ensemble agent needs to choose an action in a given state, it first requests actions from both the DQN and PPO agents. The ensemble agent calculates weights for each agent based on their past performance. Figure 3 shows the process of how the ensemble framework performs and chooses one of the algorithms for action.

These weights are determined using a formula that takes into account the total rewards and total episodes experienced by each agent. The addition of 1 in the formulas serves to prevent division by zero and smooth the weights, enhancing the stability of the learning process. Since the total episode count begins at zero during the first episode, adding 1 to the denominator ensures valid calculations and avoids undefined behavior. Similarly, incorporating 1 into the numerator acts as a small regularization term, providing stability, particularly in the early episodes.

The specific formulas used for weight calculation are:

$$\text{weight}_{\text{dqn}} = \frac{1 + \text{total\_rewards}_{\text{dqn}}}{1 + \text{total\_episodes}_{\text{dqn}}} \tag{16}$$

$$\text{weight}_{\text{ppo}} = \frac{1 + \text{total\_rewards}_{\text{ppo}}}{1 + \text{total\_episodes}_{\text{ppo}}} \tag{17}$$

After calculating the weights, the ensemble agent normalizes them. This normalization step ensures that the weights represent probabilities and can be used to select actions. The normalized weights are computed as follows:

$$\text{norm\_weight}_{\text{dqn}} = \frac{\text{weight}_{\text{dqn}}}{\text{weight}_{\text{dqn}} + \text{weight}_{\text{ppo}}} \tag{18}$$

$$\text{norm\_weight}_{\text{ppo}} = \frac{\text{weight}_{\text{ppo}}}{\text{weight}_{\text{dqn}} + \text{weight}_{\text{ppo}}} \tag{19}$$

Finally, the ensemble agent selects an action by sampling from a categorical distribution defined by the normalized weights. The probability of selecting the action suggested by the DQN agent is given by $\text{norm\_weight}_{\text{dqn}}$, and the probability of selecting the action suggested by the PPO agent is given by $\text{norm\_weight}_{\text{ppo}}$. If the $\text{norm\_weight}_{\text{ppo}}$ is less than $\text{norm\_weight}_{\text{dqn}}$, the ensemble agent selects the action suggested by the DQN agent. Otherwise, it selects the action suggested by the PPO agent.

Updates and reward calculations are carried out at the end of each step, rather than at the end of the episode. The variables $\text{total\_episodes}_{\text{dqn}}$ and $\text{total\_episodes}_{\text{ppo}}$ are for managing the weighting strategy in the ensemble framework, ensuring consistent decision-making across the full episode count for both DQN and PPO agents.

By combining the decisions of both agents based on their respective weights, the ensemble agent uses each agent's strengths and adapts its behaviour dynamically based on their performance. This allows the ensemble agent to achieve better performance and robustness than a single agent alone.

D. TRAINING PROCEDURE

The training procedure involves simulating the spin chain dynamics and training the RL agents over multiple episodes to achieve high fidelity between the evolved and target states.

- **Initialization**: Each episode begins with a randomly initialized quantum state ψ and a target state ψ_{target}.
- **Agent actions**: At each time step, both DQN and PPO agents select actions to adjust the magnetic field strengths B_n.
- **State evolution**: The Hamiltonian is updated based on the chosen actions, and the quantum state ψ evolves according to the Schrödinger equation.
- **Reward calculation**: Fidelity between the evolved and target states is computed, serving as the reward signal.
- **Experience storage and replay**: The agent's store experiences, and the ensemble agent aggregates their actions. The ensemble agent's action is executed, and the experiences are used for updating the individual agents' networks.
- **Episode termination**: An episode concludes after a set number of time steps, with the total reward recorded for performance evaluation.

It is important to mention that in this ensemble and agent selection technique, exploration occurs only within the selected agent, rather than across the ensemble during the agent selection phase. This approach emphasizes targeted exploration within each agent, allowing the ensemble to leverage the unique exploration capabilities of individual agents. Once an agent is chosen, it performs exploration according to its own parameters and mechanisms.

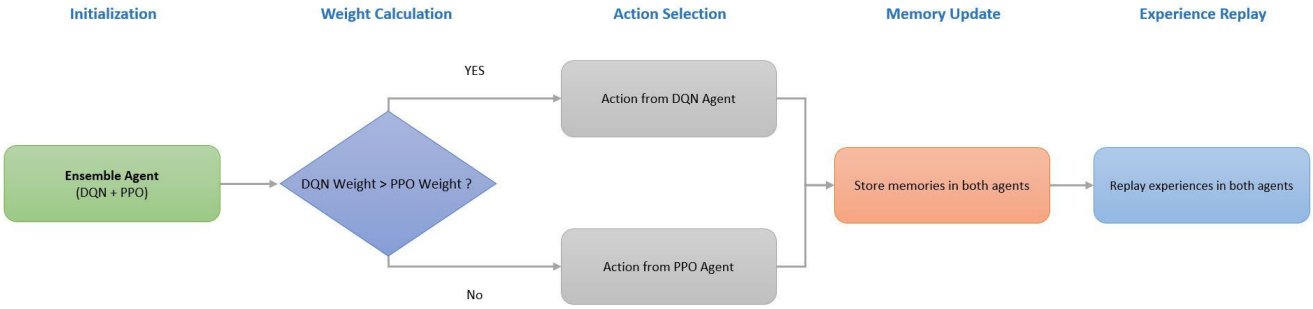

FIGURE 3. Ensemble training process.

TABLE 1. Parameters used in the project.

Category	Parameter	Description	Value
Quantum Control	N	Number of spins in the chain	3 – 7
	J	Coupling strength	1
	B	Magnetic field strength	100
	dt	Time step	0.15
	$\hbar$	Reduced Planck's constant	1
DQN Agent	state size	The number of complex amplitudes representing the quantum state (doubled for real and imaginary parts)	2×2^N
	action size	Two actions representing whether to apply a magnetic field or not (control on/off)	2
	memory	Replay memory size	40000
	γ	Discount factor for future rewards	0.9
	ϵ	Initial exploration rate	0.3
	ϵ_{min}	Minimum exploration rate	0.01
	ϵ_{decay}	Decay rate for exploration	0.2
	α	Learning rate for the optimiser	0.01
	batch size	Size of the minibatch	50
	train start	Minimum size of memory before training starts	900
PPO Agent	state size	The number of complex amplitudes representing the quantum state (doubled for real and imaginary parts)	2×2^N
	action size	Two actions representing whether to apply a magnetic field or not (control on/off)	2
	γ	Discount factor for future rewards	0.9
	α	Learning rate for the optimiser	0.01
	ϵ	Clipping parameter	0.2
	value coefficient	Coefficient for value loss	0.5
	entropy coefficient	Coefficient for entropy loss	0.01
Training	C	Update period for the target network	200
	M	Total number of training episodes	1000
Target State	ψ_{target}	Randomly generated target state	-

E. PSEUDOCODE FOR ENSEMBLE-BASED QUANTUM CONTROL

The given pseudocode in the algorithm 1 describes the ensemble agent approach to quantum control that combines DQN and PPO algorithms. This approach initializes the ensemble agent and specifies important functions for calculating the Hamiltonian, transforming complex states to real, and evaluating fidelity.

F. IMPLEMENTATION DETAILS

The implementation is carried out in Python [36] using TensorFlow [37] for constructing and training the neural networks, SciPy [38] for numerical computations, and Matplotlib [39] for visualizations. We used Windows 10 64-bit Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz, 12 GB RAM to test the implementation. Table 1 provides the specific parameters, such as learning rates, discount factors, and network architectures that are chosen to optimise performance for this setup.

IV. RESULTS AND DISCUSSION

This section provides and evaluates the results of agent training on the spin chain quantum control problem. We aim to provide a thorough analysis of the performance of RL

Algorithm 1 Ensemble Agent for Quantum Control

Require: $N, J, B, \Delta t, \hbar, \psi_{\text{target}}$
  Initialize ensemble agent (DQN and PPO)
  Function *hamiltonian*($J, B_{\text{fields}}, \sigma_{\text{noisy}}$):
    Initialize H
    Initialize Pauli matrices σ^x, σ^y, and σ^z
    Update H for XY terms using σ^x and σ^y
    Update H for magnetic field terms using σ^z
    Add Gaussian noise to H with standard deviation σ_{noisy}

    return H
  Function *complex_to_real*(ψ): **return** combination of real and imaginary parts
  Function *fidelity*($\psi_{\text{target}}, \psi$): **return** $|\langle\psi_{\text{target}}|\psi\rangle|^2$
  Initialize training episodes M
  for episode $= 1$ to M **do**
    Initialize ψ, B_{fields}, *total_reward*
    for $t = 0$ to $1/\Delta t$ **do**
      state $\leftarrow$ *complex_to_real*(ψ) // One of all possible states (2×2^N)
      action $\leftarrow$ ensemble agent action // Action selected using DQN/PPO; exploration occurs within the selected agent.
      Update B_{fields} based on *action* // Binary On/Off control
      $H \leftarrow$ *hamiltonian*($J, B_{\text{fields}}, \sigma_{\text{noisy}}$)
      $\psi_{\text{next}} \leftarrow$ evolve ψ with H and Δt
      Normalize ψ_{next}
      reward $\leftarrow$ *fidelity*($\psi_{\text{next}}, \psi_{\text{target}}$) + distance reward ($t$)

      total_reward $\leftarrow$ *total_reward* + *reward*
      next_state $\leftarrow$ *complex_to_real*(ψ_{next})
      Store experience and train ensemble agent
      if $\|\psi_{\text{next}} - \psi\| < 10^{-5}$ **then**
        break
      end if
      $\psi \leftarrow \psi_{\text{next}}$
    end for
    Record *total_reward*
  end for

algorithms separately and when combined into an ensemble framework.

Following this, the efficacy of GRAPE, MPC, and the ensemble agent will be examined in both noise-free and noisy settings. This comparison aims to provide comprehensive insights into their specific advantages and limits. By examining their performance under both ideal and difficult settings, the goal is to determine the best option for real-world applications. The total reward values may be negative in early steps due to the randomness. To improve accuracy and account for these negative values during early training, the reported results are averaged over five runs.

TABLE 2. Total rewards during training episodes.

Category	Episode Range	Total Reward
Average Reward for DQN	1-1000	0.6424
Average Reward for PPO	1-1000	0.6191
Average Reward for Ensemble	1-1000	0.7313

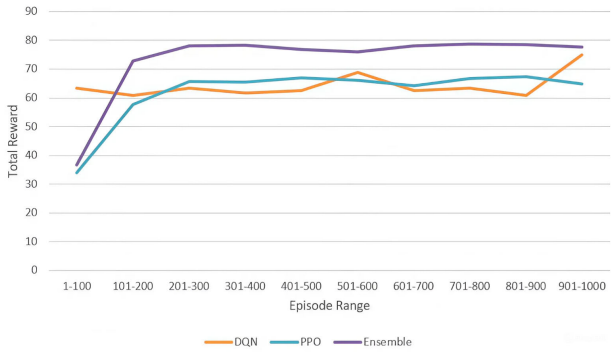

FIGURE 4. Total rewards for six spin.

A. TRAINING PERFORMANCE

The training process was conducted over 1000 episodes, with the total reward (fidelity + distance reward) accumulated by the agents recorded at intervals of 100 episodes. The rewards represent the fidelity between the evolved quantum state and the target state, with higher rewards indicating better performance in approximating the desired state. To further enhance this, we calculated a distance reward, which increases as the quantum state approaches the target. This additional reward encourages the agents to experiment with different actions, making the learning process more effective. Table 2 summarizes the total rewards achieved during the different intervals of the training process. The results in Figure 4 indicate that the Ensemble agent outperformed both the DQN and PPO agents. The performance of the Ensemble agent improved consistently over the training episodes and achieved the highest total rewards in most periods. The average reward for the Ensemble agent was 0.7313, which is higher than the average reward for both DQN (0.6424) and PPO (0.6191).

In analyzing the DQN results, it is observed that the algorithm begins with a total reward of approximately 63.47 in the first episode range (1-100), followed by fluctuations in subsequent ranges, with values dropping to around 60.87 in the 101-200 range and 61.69 in the 301-400 range. However, it is important to note that such fluctuations are not uncommon in reinforcement learning algorithms, especially during the exploration phase. Despite these fluctuations, a significant improvement in performance is seen towards the final episodes, where the total reward reaches approximately 74.99 in the 901-1000 range. This consistent increase towards the end suggests that the DQN

algorithm is making meaningful progress and indicates a trend towards convergence. While the variations in the earlier episode ranges introduce some uncertainty regarding immediate convergence, they do not preclude the possibility that the DQN algorithm is converging overall. The final episode results indicate that the algorithm started an effective path to learn, reflecting the nature of reinforcement learning where convergence can occur gradually over time.

The PPO agent, starting with a significantly lower total reward of 33.9114 in the initial episodes (1-100), showed a steady and gradual improvement over the training period. By the end of the training episodes (901-1000), the PPO agent achieved a total reward of 64.8358. This steady improvement highlights the PPO algorithm's ability to learn and adapt over time, although it did not reach the peak performance of the Ensemble agent. The Ensemble agent, combining the strengths of both DQN and PPO, showed a superior learning pattern. Starting with a total reward of 36.7073 in the initial episodes (1-100), the Ensemble agent showed a sharp increase in performance, achieving a total reward of 72.7812 in the next set of episodes (101-200).

This rapid improvement continued, with the Ensemble agent consistently achieving higher rewards in subsequent episodes, peaking at 78.7188 in episodes (701-800). The consistently high performance of the Ensemble agent suggests that the combination of DQN and PPO allows it to use the strengths of both algorithms, resulting in a more robust learning process.

Figure 5 and Table 3 show the comparison of control methods, including GRAPE, robust MPC, and the ensemble agent, and present information about how they perform in quantum systems with different spin configurations. The number of iterations has been set to 1000. Initially, GRAPE demonstrated superior fidelity in noise-free conditions but faced scalability issues as the number of spins increased, revealing limitations in handling larger quantum systems. Similarly, robust MPC maintained stable fidelity levels across various spin configurations, but its performance declined, indicating potential challenges in achieving high fidelity in more complex quantum environments.

The fidelity achieved by the ensemble agent also decreased as the number of spins increased, but it showed slightly better resilience to larger quantum systems compared to GRAPE and MPC. This suggests that the ensemble agent might offer improved scalability for controlling quantum systems with a higher number of spins.

In Figure 6 and Table 4, when all three methods were subjected to noise and uncertainty, the fidelity of the GRAPE significantly decreased across all spin configurations, which shows its sensitivity to environmental perturbations. In contrast, MPC showed resilience to noise and uncertainty, maintaining relatively stable fidelity levels, though potentially insufficient for demanding real-world applications.

On the other hand, the ensemble agent distinguished itself by sustaining superior fidelity levels even in challenging conditions, which shows its ability to adapt and remain reliable in noisy and uncertain quantum systems. This resilience shows that the ensemble method can be the best control method for practical quantum applications, where consistent and robust performance is crucial for success. Overall, the comparison underscores the capability of the ensemble agent to surpass GRAPE and MPC in handling noise and uncertainty and highlights its potential for addressing challenges in real-world quantum environments.

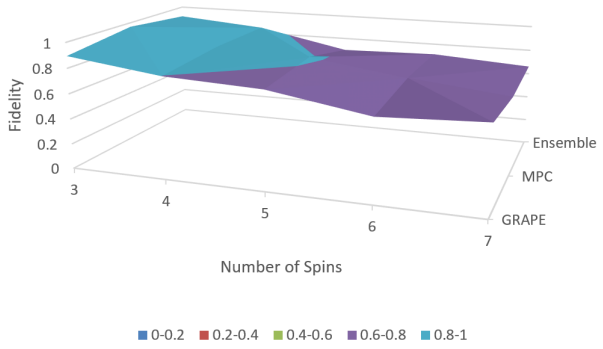

FIGURE 5. Performance comparison of different control methods without noise.

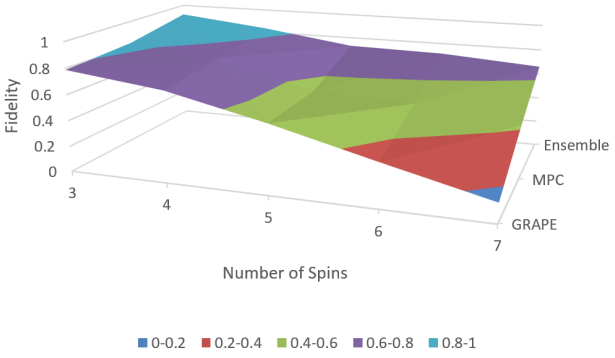

FIGURE 6. Performance comparison of different control methods with noise.

B. DISCUSSION AND LIMITATIONS

The results obtained from the training phase offer significant insights into the efficacy of the RL algorithms and the ensemble method. The initial improvement in rewards underscores the capability of the RL agents to manage and control the quantum system effectively. However, the fluctuations in performance reveal the complex interaction between the agents' learning processes and the challenges presented by the quantum states. These variations throughout the training phase highlight the agents' difficulties in adapting to the diverse quantum state configurations. The ensemble learning approach demonstrates a positive impact on overall performance. By utilizing both DQN and PPO, the ensemble agent showed superior performance in our scenario. This adaptability and reliability are valuable in practical applications involving uncertainties and fluctuations. The

TABLE 3. Values of different control methods without noise.

Spins	3	4	5	6	7
GRAPE	0.8944	0.7989	0.7649	0.6394	0.675
MPC	0.9662	0.846	0.8157	0.7054	0.6227
Ensemble	0.9126	0.8507	0.6962	0.7116	0.661

TABLE 4. Values of different control methods with noise.

Spins	3	4	5	6	7
GRAPE	0.7878	0.6984	0.5289	0.3352	0.1482
MPC	0.8218	0.749	0.542	0.5098	0.4187
Ensemble	0.9169	0.836	0.7291	0.7143	0.6604

ensemble agent performs better overall due to its ability to balance exploration and exploitation compared to the isolated DQN and PPO agents. This results in a more robust learning process that can adapt to the varying difficulties of the quantum control task. The results also suggest that while solo agents like DQN and PPO can achieve high rewards, instability and fluctuations often mark their performance. In contrast, the ensemble agent's performance is more consistent, indicating higher reliability in controlling the quantum system. This consistency is particularly valuable in practical quantum control applications. This approach showcases superior performance compared to DQN and PPO and offers distinct advantages that make it more suitable for real-world scenarios. Unlike GRAPE and robust MPC, which are specialized control methods designed for specific tasks, the ensemble method's adaptability allows it to tackle a broader range of quantum control challenges. Additionally, ensemble agents' ability to balance exploration and exploitation improves the robustness of the learning process, making it well-suited for dynamic and evolving environments encountered in real-world applications. This adaptability and reliability are invaluable in domains where precision, stability, and scalability are essential, positioning the ensemble method as a promising solution for addressing the complexities of quantum control tasks in practical settings.

Our findings provide valuable insights into the effectiveness of RL algorithms and ensemble techniques in addressing the complexity of quantum control tasks. Despite the observed fluctuations, the ensemble method emerges as a promising approach, offering both stability and adaptability. It is important to note that the low fidelity in the outcomes that have been reported is driven by the structural complexity of choosing random states for every iteration, which makes the control problem more challenging for all approaches. The reason behind this choice of scenario was to represent a more tricky environment and highlight the superior performance of the ensemble method over other methods.

Our study also acknowledges its limitations. Fidelity is used as an optimization factor because it directly measures the similarity between quantum states, which is essential for accurate state preparation. While generalization plays a crucial role in many learning tasks, we focus on state-specific optimization, as the primary objective is to maximize the fidelity of the prepared state. The present study primarily focuses on the evaluation of agent performance based on reward metrics, leaving room for further exploration in the time domain response analysis. Specifically, understanding how the quantum state evolves and compares to the target quantum state under the influence of the trained agent would provide deeper insights into the behavior of the quantum system during training. This would add a valuable dimension to comprehending the agent's effectiveness beyond mere reward outcomes. Additionally, the Eligibility Traces technique, a method known to enhance the learning process by bridging the gap between Monte Carlo and TD learning, has not been employed in this research. The inclusion of such a technique could potentially refine the training process and improve the convergence rate. However, to maintain a clear focus and avoid additional complexity in the scope, these aspects were not incorporated in this study. The limited number of training episodes, the reward structure, and the inherent randomness of the quantum system can induce fluctuations in the training outcomes. These factors contribute to variations in evaluation outcomes with and without noise, as well as making the convergence of the DQN algorithm less apparent.

V. CONCLUSION AND FURTHER WORK

This study explored the application of RL algorithms to the quantum control of a spin chain system. We implemented and trained two RL algorithms, including DQN and PPO, as well as an ensemble approach that combined these algorithms. The results indicate that ensemble learning can improve performance by using multiple agents, giving us more adaptive and effective control strategies. Furthermore, the performance of the ensemble agent was compared to GRAPE and robust MPC to validate this approach's efficiency in both noise-free and noisy environments.

The study findings showed that the ensemble agent was able to learn, improve its performance, increase the fidelity and maintain stable performance under noisy conditions. This

suggests that combining multiple agents can enhance the overall effectiveness and robustness of the control strategies.

Despite these positive results, further improvement should still be considered. One useful direction is the integration of Hierarchical Reinforcement Learning (HRL). HRL allows for decomposing complex control tasks into simpler, more manageable subtasks. By structuring the control problem hierarchically, agents can focus on solving specific components of the task, which can lead to more efficient learning and better performance. Additionally, establishing a framework for lifelong learning can further enhance the robustness and adaptability of the agents, enabling them to improve and adapt to new challenges continuously.

APPENDIX

DATA AVAILABILITY STATEMENT

The code repository for this project can be accessed by clicking https://github.com/FarshadRahimiGhashghaei/AdvancedQuantumControl (accessed on 7 August 2024). Within the Ensemble Learning.py file, the ensemble agent tries to optimise magnetic fields in the XY spin chain using Hamiltonian construction, time evolution simulation, and neural networks for policy and value updates. The GRAPE.py and MPC.py files apply GRAPE and robust MPC algorithms to optimise magnetic fields while accounting for noise.

REFERENCES

[1] B. Camino, J. Buckeridge, P. A. Warburton, V. Kendon, and S. M. Woodley, "Quantum computing and materials science: A practical guide to applying quantum annealing to the configurational analysis of materials," *J. Appl. Phys.*, vol. 133, no. 22, Jun. 2023, Art. no. 221102, doi: 10.1063/5.0151346.

[2] V. Lordi and J. M. Nichol, "Advances and opportunities in materials science for scalable quantum computing," *MRS Bull.*, vol. 46, no. 7, pp. 589–595, Jul. 2021, doi: 10.1557/s43577-021-00133-0.

[3] Y. Cao, J. Romero, and A. Aspuru-Guzik, "Potential of quantum computing for drug discovery," *IBM J. Res. Develop.*, vol. 62, no. 6, pp. 1–20, Nov. 2018, doi: 10.1147/JRD.2018.2888987.

[4] F. Ghashghaei, Y. Ahmed, N. Elmrabit, and M. Yousefi, "Enhancing the security of classical communication with post-quantum authenticated-encryption schemes for the quantum key distribution," *Computers*, vol. 13, no. 7, p. 163, Jul. 2024, doi: 10.3390/computers13070163.

[5] 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*, vol. 12, no. 1, Mar. 2022, Art. no. 011059, doi: 10.1103/physrevx.12.011059.

[6] I. Khalid, C. A. 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 *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 4133–4139, doi: 10.1109/CDC45484.2021.9683463.

[7] M. F. Lazin, C. R. Shelton, S. N. Sandhofer, and B. M. Wong, "High-dimensional multi-fidelity Bayesian optimization for quantum control," *Mach. Learning: Sci. Technol.*, vol. 4, no. 4, Oct. 2023, Art. no. 045014, doi: 10.1088/2632-2153/ad0100.

[8] V. Cimini, M. Valeri, E. Polino, S. Piacentini, F. Ceccarelli, G. Corrielli, N. Spagnolo, R. Osellame, and F. Sciarrino, "Deep reinforcement learning for quantum multiparameter estimation," *Proc. SPIE*, vol. 5, no. 1, Feb. 2023, Art. no. 016005, doi: 10.1117/1.ap.5.1.016005.

[9] 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*, vol. 8, no. 3, Sep. 2018, Art. no. 031086, doi: 10.1103/physrevx.8.031086.

[10] 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.*, vol. 9, no. 1, p. 19, Jul. 2022, doi: 10.1140/epjqt/s40507-022-00138-x.

[11] T. S. Mahesh, P. Batra, and M. H. Ram, "Quantum optimal control: Practical aspects and diverse methods," *J. Indian Inst. Sci.*, vol. 103, no. 2, pp. 591–607, Sep. 2022, doi: 10.1007/s41745-022-00311-2.

[12] B. Cheng et al., "Noisy intermediate-scale quantum computers," *Frontiers Phys.*, vol. 18, no. 2, Mar. 2023, Art. no. 21308, doi: 10.1007/s11467-022-1249-z.

[13] C. A. Weidner, E. A. Reed, J. Monroe, B. Sheller, S. O'Neil, E. Maas, E. A. Jonckheere, F. C. Langbein, and S. G. Schirmer, "Robust quantum control in closed and open systems: Theory and practice," 2023, *arXiv:2401.00294*.

[14] G. A. L. White, C. D. Hill, and L. C. L. Hollenberg, "Performance optimization for drift-robust fidelity improvement of two-qubit gates," *Phys. Rev. Appl.*, vol. 15, no. 1, Jan. 2021, Art. no. 014023, doi: 10.1103/physrevapplied.15.014023.

[15] D. Dong, C. Chen, T.-J. Tarn, A. Pechen, and H. Rabitz, "Incoherent control of quantum systems with wavefunction-controllable subspaces via quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern., B (Cybernetics)*, vol. 38, no. 4, pp. 957–962, Aug. 2008, doi: 10.1109/TSMCB.2008.926603.

[16] S. Cong, J. Zhang, S. Kuang, and S. Harraz, "Real-time optimal state estimation-based feedback control for stochastic quantum systems in the non-Markovian case," *J. Syst. Sci. Complex.*, vol. 36, no. 6, pp. 2274–2291, Dec. 2023, doi: 10.1007/s11424-023-2266-x.

[17] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, "When does reinforcement learning stand out in quantum control? A comparative study on state preparation," *npj Quantum Inf.*, vol. 5, no. 1, p. 85, Oct. 2019, doi: 10.1038/s41534-019-0201-8.

[18] T. Seyde, P. Werner, W. Schwarting, M. Wulfmeier, and D. Rus, "Growing Q-networks: Solving continuous control tasks with adaptive control resolution," 2024, *arXiv:2404.04253*.

[19] Y. Song, P. N. Suganthan, W. Pedrycz, J. Ou, Y. He, Y. Chen, and Y. Wu, "Ensemble reinforcement learning: A survey," *Appl. Soft Comput.*, vol. 149, Dec. 2023, Art. no. 110975, doi: 10.1016/j.asoc.2023.110975.

[20] L. Liu, J. Wu, X. Li, and H. Huang, "Dynamic ensemble selection with reinforcement learning," in *Advanced Intelligent Computing Technology and Applications* (Lecture Notes in Computer Science). Singapore: Springer, 2023, pp. 629–640, doi: 10.1007/978-981-99-4761-4_53.

[21] V. Goyal and J. Grand-Clément, "Robust Markov decision processes: Beyond rectangularity," *Math. Oper. Res.*, vol. 48, no. 1, pp. 203–226, Feb. 2023, doi: 10.1287/moor.2022.1259.

[22] J. Song, W. Yang, and C. Zhao, "Decision-dependent distributionally robust Markov decision process method in dynamic epidemic control," *IISE Trans.*, vol. 56, no. 4, pp. 458–470, Jun. 2023, doi: 10.1080/24725854.2023.2219281.

[23] I. Khalid, C. A. Weidner, E. A. Jonckheere, S. G. Schirmer, and F. C. Langbein, "Sample-efficient model-based reinforcement learning for quantum control," *Phys. Rev. Res.*, vol. 5, no. 4, Oct. 2023, Art. no. 043002, doi: 10.1103/physrevresearch.5.043002.

[24] O. Ben-Porat, Y. Mansour, M. Moshkovitz, and B. Taitler, "Principal-agent reward shaping in MDPs," in *Proc. AAAI Conf. Artif. Intell.*, Mar. 2024, vol. 38, no. 9, pp. 9502–9510, doi: 10.1609/aaai.v38i9.28805.

[25] L. Giannelli, S. Sgroi, J. Brown, G. S. Paraoanu, M. Paternostro, E. Paladino, and G. Falci, "A tutorial on optimal control and reinforcement learning methods for quantum technologies," *Phys. Lett. A*, vol. 434, May 2022, Art. no. 128054, doi: 10.1016/j.physleta.2022.128054.

[26] Y. Huang, "Deep Q-networks," in *Deep Reinforcement Learning: Fundamentals, Research and Applications*, H. Dong, Z. Ding, and S. Zhang, Eds., Singapore: Springer, 2020, pp. 135–160, doi: 10.1007/978-981-15-4095-0_4.

[27] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015, *arXiv:1511.05952*.

[28] Q. Fu, K. Li, J. Chen, J. Wang, Y. Lu, and Y. Wang, "Building energy consumption prediction using a deep-forest-based DQN method," *Buildings*, vol. 12, no. 2, p. 131, Jan. 2022, doi: 10.3390/buildings12020131.

[29] A. Iqbal, M.-L. Tham, and Y. C. Chang, "Double deep Q-network-based energy-efficient resource allocation in cloud radio access network," *IEEE Access*, vol. 9, pp. 20440–20449, 2021, doi: 10.1109/ACCESS.2021.3054909.

[30] H. Ma, D. Dong, S. X. Ding, and C. Chen, "Curriculum-based deep reinforcement learning for quantum control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 11, pp. 8852–8865, Mar. 2022, doi: 10.1109/TNNLS.2022.3153502.
[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[32] C. Ching-Yun Hsu, C. Mendler-Dünner, and M. Hardt, "Revisiting design choices in proximal policy optimization," 2020, *arXiv:2009.10897*.
[33] M. Sun, V. Kurin, G. Liu, S. Devlin, T. Qin, K. Hofmann, and S. Whiteson, "You may not need ratio clipping in PPO," 2022, *arXiv:2202.00079*.
[34] H. Tang, Z. Meng, J. Hao, C. Chen, D. Graves, D. Li, C. Yu, H. Mao, W. Liu, Y. Yang, W. Tao, and L. Wang, "What about inputting policy in value function: Policy representation and policy-extended value function approximator," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2022, vol. 36, no. 8, pp. 8441–8449, doi: 10.1609/aaai.v36i8.20820.
[35] X.-M. Zhang, Z.-W. Cui, X. Wang, and M.-H. Yung, "Automatic spin-chain learning to explore the quantum speed limit," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 5, May 2018, Art. no. 052333, doi: 10.1103/physreva.97.052333.
[36] Python.org. (2019). *Python*. Accessed: May 30, 2024. [Online]. Available: https://www.python.org/
[37] TensorFlow. (2019). *TensorFlow*. Accessed: May 30, 2024. [Online]. Available: https://www.tensorflow.org/
[38] Scipy.org. (2020). *SciPy.org—SciPy.org*. Accessed: May 30, 2024. [Online]. Available: https://scipy.org/
[39] Matplotlib.org. (2012). *Matplotlib: Python Plotting—Matplotlib 3.1.1 Documentation*. Accessed: May 30, 2024. [Online]. Available: https://matplotlib.org/

FARSHAD RAHIMI GHASHGHAEI received the B.Eng. degree in computer engineering from the Institute for Higher Education ACECR Khouzestan, Ahvaz, Iran, and the M.Sc. degree in cyber security from Birmingham City University. He is currently a Motivated Researcher with expertise in quantum computing. His research interests include quantum cryptograph, quantum computing, cryptography, and machine learning, driven by a passion to enhance innovation and secure communication.

NEBRASE ELMRABIT received the M.Sc. degree in computer and network security from Middlesex University, London, and the Ph.D. degree in computer science (cyber security) from Loughborough University, in 2018. He is currently an Accomplished Cybersecurity Expert with research interests that include insider threat prevention, cybersecurity architecture, privacy, and digital forensics.

AYYAZ-UL-HAQ QURESHI received the B.S. degree in electrical (computer) engineering and the M.Sc. degree in advanced computer networks and the Ph.D. degree from Glasgow Caledonian University, U.K. He is currently a Lecturer with the Department of Cyber Security and Networks (CSN). He is currently an Experienced Researcher with many years of teaching experience in prestigious institutions. His research interests include cybersecurity, computer networks, artificial intelligence, cognitive systems, the Internet of Things, and LoRaWAN technologies.

ADNAN AKHUNZADA (Senior Member, IEEE) brings 15 years of expertise in research and development (R&D) at the nexus of the ICT industry and academia. Renowned for his high-impact publications, U.S. patents, and commercial products, his patented innovations in cybersecurity and AI have secured multi-million-dollar projects with global entities, such as Vinnova and EU Horizon. In 2023, Stanford University recognized him as one of the top 2% scientists globally for his outstanding scholarly contributions. Leveraging his robust cybersecurity skills and advanced technological knowledge, he excels in solving industrial challenges and developing state-of-the-art security tools and frameworks. His expertise spans cybersecurity & AI, secure future internet, secure & dependable software defined networks, and large-scale distributed systems (including cloud, fog, edge, the IoT, IoE, the IIoT, and CPS). Additionally, his work on lightweight cryptographic communication protocols, QoS/QoE, and adversarial machine learning is shaping the future of secure and dependable systems. He is also a Professional Member of ACM.

MEHDI YOUSEFI received the B.Sc. degree in software engineering, the M.Sc. degree in network security, and the Ph.D. degree in cyber security. He specializes in cyber security, information security, network security, computer networking, and machine learning. He is currently an Experienced Academician in cyber security and networking with seven years of experience. He has more than seven years of industry experience in networking and cyber security in the financial sector.