Unveiling the Best Performers: A Comparative Analysis of RL Algorithms in QRL with TensorFlow Quantum

Arshia Sangwan

CSAI, Plaksha University, Mohali

arshia.sangwan@plaksha.edu.in

Rahath Malladi

RCPS, Plaksha University, Mohali

rahath.malladi@plaksha.edu.in

Keywords **— Quantum Reinforcement Learning (QRL), TensorFlow Quantum (TFQ), Quantum Computing, Reinforcement Learning, OpenAI Gym, Parametrized Quantum Circuits (PQCs)**

I. INTRODUCTION

Reinforcement learning (RL) has achieved significant success in various domains but faces challenges when applied to quantum systems due to their inherent complexity. Quantum reinforcement learning (QRL) seeks to address this by marrying RL techniques with the power of quantum computing. This project aims to analyse and compare the performance of different RL algorithms within the framework of QRL, utilizing the capabilities of TensorFlow Quantum (TFQ) (Broughton et al., 2021).

II. PROBLEM STATEMENT

With the growing interest in QRL, a comprehensive understanding of different RL algorithms' effectiveness in this unique setting is crucial. This study seeks to address the following:

1. How does the performance of various RL algorithms differ when applied to QRL problems?
2. Which specific algorithms are best suited for different types of QRL tasks?

III. METHODOLOGY

This project will leverage TensorFlow Quantum (TFQ), a leading library for constructing and training quantum machine learning models. We will integrate various well-established RL algorithms, including Deep Q-Learning (DQN), Soft Actor-Critic (SAC), and Proximal Policy Optimization (PPO), with Parametrized Quantum Circuits (PQCs) within the TFQ framework (Jerbi et al., 2021). Each algorithm will be evaluated on a set of standardized QRL benchmarks, potentially sourced from the OpenAI Gym library, which facilitates the active learning of RL agents (Skolik et al., 2022). This comprehensive approach allows for a rigorous comparative analysis of the algorithms' effectiveness, assessed through metrics like reward accumulation and learning efficiency.

IV. OUTCOMES

Through this study, we expect to gain valuable insights into the performance of different RL algorithms in QRL environments. The project aims to:

1. Identify the strengths and weaknesses of each algorithm in the context of QRL.
2. Provide recommendations for choosing the most suitable RL algorithm for different QRL applications.
3. Offer valuable resources for further research in the field of QRL.

V. LITERATURE REVIEW

A) Methodologies & Approaches

Quantum Reinforcement Learning (QRL) presents two principal classification methodologies: classical-quantum trade-off and Gate-Annealing Quantum approaches. Within the former, algorithms are broadly classified into classical-quantum hybrid and fully quantum categories. While hybrid algorithms blend classical RL techniques with quantum circuits, fully quantum algorithms leverage quantum computing throughout the learning process. Notable examples include Deep Q-Learning (DQN) and Quantum Policy Gradient (QPG). Evaluating performance across these methodologies reveals nuanced dependencies on factors such as task complexity and quantum system noise (Neumann et al., 2023).

B) Quantum Agents in Gym

Introducing Quantum Agents (Skolik et al., 2022) and Parametrized Quantum Policies (Jerbi et al., 2021) showcases the fusion of quantum computations with classical learning algorithms. Parametrized Quantum Circuits (PQCs) offer a versatile framework for policy representation, with algorithms like RAW-PQC and SOFTMAX-PQC demonstrating competitive performance in benchmarking tasks. These advancements underscore the theoretical learning advantages and efficient policy sampling facilitated by PQCs.

C) Challenges and Scope

QRL faces several challenges, including hardware limitations, algorithm development, and theoretical understanding. Future research directions emphasize hybrid quantum-classical algorithms, novel QRL algorithm development, and integration with Quantum Error Correction. Despite these challenges, QRL holds immense potential across diverse applications such as quantum control, game playing, and quantum optimization.

D) Applications of QRL

QRL finds application in various domains, including quantum control, game playing, quantum simulation, and quantum finance. These applications highlight the versatility and transformative potential of QRL in solving complex real-world problems.

In conclusion, QRL represents a promising frontier at the nexus of quantum computing and reinforcement learning principles. Despite challenges, ongoing research endeavours aim to unlock its full potential, paving the way for transformative advancements in decision-making processes across diverse domains.

(This is a mere gist of our entire literature review, please read further and go through our detailed literature review here - Literature Review - QRL - Arshia S & Rahath M)

VI. PROGRESS TILL DATE

We've implemented the Policy Gradient RL with PQC Policies and Deep Q-Learning with PQC Q-function approximators on the vanilla cartpole-v1 environment of OpenAI's Gym library. An iteration on applying the same algorithms on the Mountain_Car-v0 environment are also currently in progress and can be seen on the GitHub Repo for your reference. To elaborate the QRL implementation on the cartpole-v1 further in detail:

Policy Gradient RL with PQC Policies

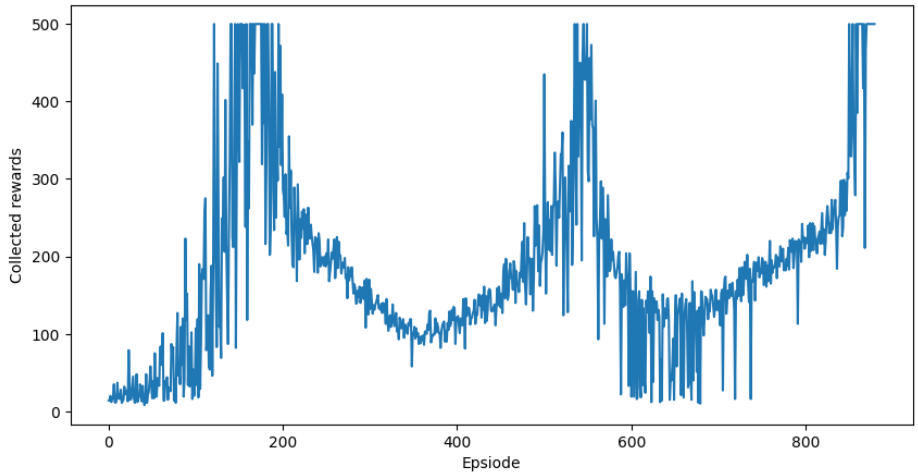

Figure 1 – Rewards vs Episodes – PGRL with PQC π

The plot above displays the rewards collected by the agent per episode throughout its interaction with the environment. Various phases of the agent's learning can be observed and understood as follows:

Initial Learning Phase: At the very beginning of the training, there are high peaks in rewards, which suggests that the agent occasionally achieved high rewards but was not consistent up-till a point.

Mid-Training Variability: As we approach the intermediate phase/middle episodes, the collected rewards significantly drop, indicating that the agent might be exploring new strategies, or the policy updates might be leading to less optimal decisions temporarily. This might also be due to the exploration-exploitation trade-off where the agent is trying out less frequently chosen actions to learn more about the environment.

Late Training Improvement: Towards the end of the graph, the rewards consistently reach higher levels, indicating that the agent is likely converging towards a better policy that yields higher rewards more consistently. Thus, the performance of the agent gets close to optimal, i.e., 500 rewards per episode (which is predetermined based on the existing literature).

Deep Q-Learning with PQC Q-Function Approximators

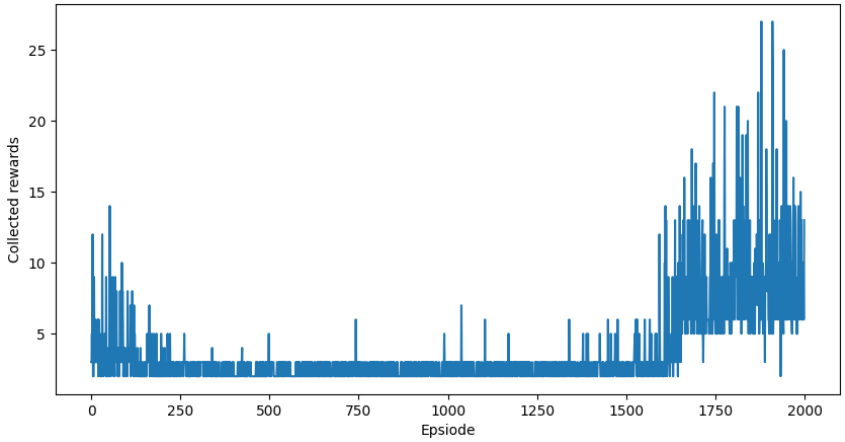

Figure 2 – Rewards vs Episodes - DQL with PQC QF^n Approximators (In Iteration)

In our trials until now, we have been trying to optimize the performance of the agent and make it converge at a favorable consistent reward. Learning takes longer for Q-learning agents since the Q-function is a "richer", more complex function to be learned than the policy. In this current graph of ours, the collected rewards are nowhere close to the convergence criteria and this is due to a few bugs in our current implementation. We're iterating on the same to make it perform better, multiple other techniques to help the agent converge to the reward criteria and a few implicitly intrinsic TensorFlow functions are currently being utilized to help the agent learn better.

VII. FURTHER PLANS

For the next one month, we aim to work on building the simulations for most of the environments in the OpenAI gym and specifically aim to investigate the performance of any two to three actor-critic agents such as Asynchronous Advantage Actor-Critic (A3C), Synchronous Advantage Actor-Critic (A2C), Deterministic Policy Gradient (DPG), Deep Deterministic Policy Gradient (DDPG), Distributed Distributional DDPG (D4PG), and Phasic Policy Gradient (PPG). This research plan is divided into specific tasks to ensure systematic implementation & evaluation, and is as follows:

Task 1: Environment Setup and Preparation

1. Iterate again, re-review and finalize the list of environments from the OpenAI Gym suitable for QRL analysis.
2. Update the necessary dependencies including TensorFlow Quantum (TFQ) and relevant RL libraries (e.g., OpenAI Baselines, Gym, etc.).
3. Ensure compatibility and integration of chosen RL algorithms with Parametrized Quantum Circuits (PQCs) within the TFQ framework.
4. Set up the experimental environment for further simulations and data collection.

Task 2: Implementation of Additional RL Algorithms

1. Begin implementation of Asynchronous Advantage Actor-Critic (A3C) algorithm with PQCs on the chosen environment.
2. Simultaneously, initiate the implementation of Synchronous Advantage Actor-Critic (A2C), Deterministic Policy Gradient (DPG), and/or Deep Deterministic Policy Gradient (DDPG) algorithms.
3. Verify the correctness of implementations through unit tests and debugging procedures.

Task 3: Fine-tuning and Optimization

1. Conduct preliminary simulations of implemented algorithms on the selected environment to identify potential performance gaps.
2. Fine-tune hyperparameters and configurations to optimize the performance of each algorithm.
3. Address any issues or discrepancies observed during the initial simulation runs.

Task 4: Evaluation and Analysis

1. Perform extensive simulations of all implemented RL algorithms on the chosen environments.
2. Collect and analyse performance metrics including reward accumulation, learning efficiency, and convergence rates.

3. Compare the performance of different RL algorithms in QRL environments, identifying strengths and weaknesses.
4. Document findings and insights obtained from the evaluation process.
5. Provide recommendations for selecting the most suitable RL algorithm for various QRL applications based on empirical evidence.

Expected Deliverables:

1. Detailed documentation of implemented RL algorithms and their integration with PQCs in TFQ.
2. Results of simulations, including performance metrics and comparative analysis.
3. Recommendations for selecting RL algorithms for QRL applications based on empirical findings.
4. Research paper section summarizing the methodology, outcomes, and future research directions.

By following this research plan of ours, we aim to advance the understanding of QRL algorithms and contribute valuable insights to the research community. Hence the proposal and the update.

REFERENCES

1. Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Halavati, R., Niu, M. Y., Zlokapa, A., Peters, E., Lockwood, O., Skolik, A., Jerbi, S., Dunjko, V., Leib, M., Streif, M., Von Dollen, D., Chen, H., … Mohseni, M. (2021). *TensorFlow Quantum: A Software Framework for Quantum Machine Learning* (arXiv:2003.02989; Version 2). arXiv. http://arxiv.org/abs/2003.02989

2. Jerbi, S., Gyurik, C., Marshall, S. C., Briegel, H. J., & Dunjko, V. (2021). *Parametrized quantum policies for reinforcement learning* (arXiv:2103.05577). arXiv. http://arxiv.org/abs/2103.05577

3. Neumann, N. M. P., de Heer, P. B. U. L., & Phillipson, F. (2023). Quantum reinforcement learning. *Quantum Information Processing*, *22*(2), 125. https://doi.org/10.1007/s11128-023-03867-9

4. Skolik, A., Jerbi, S., & Dunjko, V. (2022). Quantum agents in the Gym: A variational quantum algorithm for deep Q-learning. *Quantum*, *6*, 720. https://doi.org/10.22331/q-2022-05-24-720

Note: Driven by our unique interests in our respective majors, CSAI and RCPS, we aim to unlock the potential of QRL by analyzing RL algorithms in this domain of Quantum Computing, thus paving our path into this interesting field!

Thank You!