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. PROGRESS TILL DATE

Just to reiterate our initial update, we 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. Just to summarize the results of the initial QRL implementations on the cartpole-v1:

1. Cartpole - Policy Gradient RL with PQC Policies

The following graph (figure 1) displays the rewards collected by the agent per episode throughout its interaction with the environment, showcasing various phases of learning:

The initial high peaks indicate occasional high rewards without consistency, a mid-training drop suggesting exploration of new strategies or less optimal decisions due to the exploration-exploitation trade-off, and a late training improvement where rewards consistently reach higher levels, indicating convergence towards an optimal policy yielding close to 500 rewards per episode. The Implementation can be found on our GitHub repo and the corresponding plot is shown here for your reference. The stable yet oscillatory behaviour is obtained and supported by the forward kinematics and the dynamics of the cart pole agent itself.

The convergence space to the optimal reward is skewed owing to the false presumptions the agent must've taken during the course of its initial exploration and exploitation.

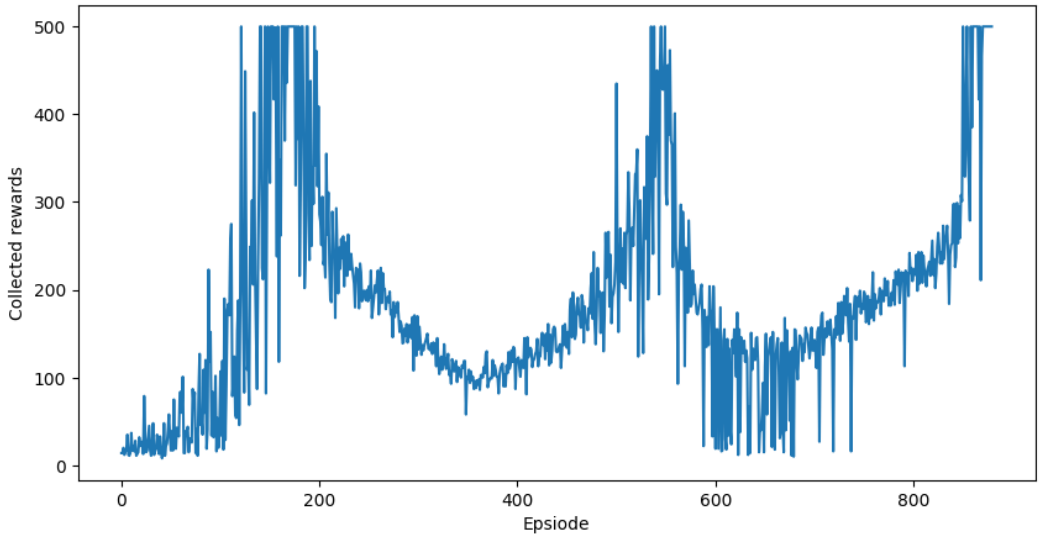

Figure 1 – Rewards vs Episodes – Cartpole - Policy Gradient-based RL with PQC π

2. Cartpole - Deep Q-Learning with PQC Q-Function Approximators

The following graph **(figure 2)** displays the mean reward per episode for training a cart pole agent using Deep Q-learning with PQC function approximators. The x-axis represents the number of episodes, and the y-axis represents the mean reward. The graph shows that the mean reward increases steadily over time, eventually reaching a value of around 160. This suggests that the agent is successfully learning to control the cart pole, but is not completely successful yet, as the convergence is still carrying on.

The rate at which the mean reward increases appears to slow down over time. This suggests that the agent is eventually converging to a near-optimal policy but is not essentially there yet. The graph does not show any significant dips or fluctuations in the mean reward. This suggests that the training process is relatively stable.

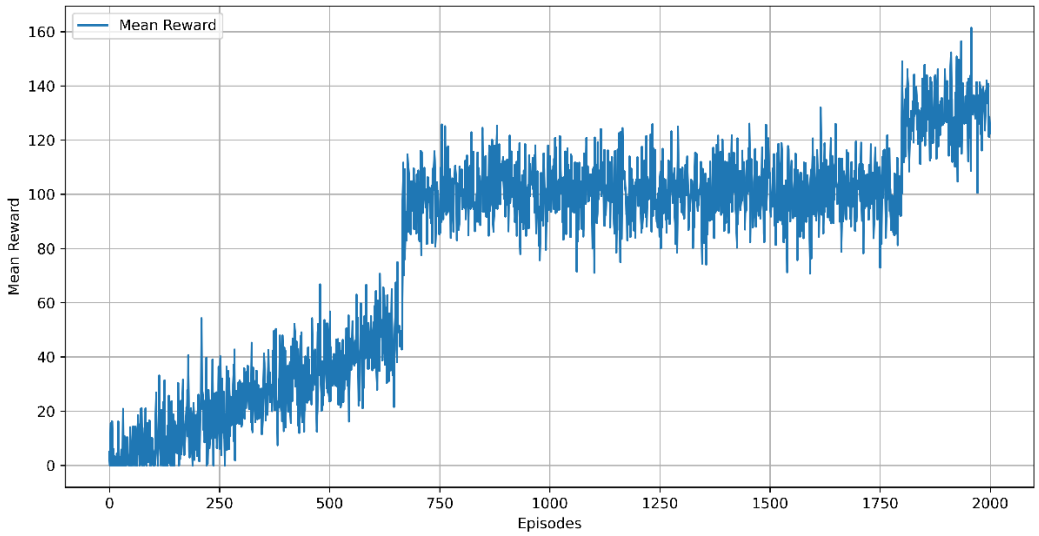

Figure 2 – Rewards vs Episodes – Cartpole - DQL with PQC QF^nApproximators

Post the implementation of the cart pole, we implemented the Mountain Car V0 from the OpenAI gym. This is a different implementation in comparison to the cart pole, given the varied state space and action space. The state space of Mountain Car encompasses both the position and velocity unlike the standard values of the cart pole which deal only with the velocity space. The mountain car scenario is also an interesting one given its need to obtain negative rewards to go against the path of requirement (towards the path against the goal) as in to gain momentum and drive towards the goal position. Hence, the implementation is considerate of this scenario and the same can be found on our GitHub repo.

3. Mountain Car - Policy Gradient RL with PQC Policies

The following graph **(figure 3)** displays the mean reward per episode for training a mountain car agent using policy gradient-based RL (REINFORCE Algorithm) and PQC Policies. The x-axis represents the number of episodes, and the y-axis represents the mean reward. The graph shows that the mean reward starts at around -480 and increases steadily to around -100 over the course of training. This suggests that the agent is successfully learning to control the mountain car.

The rate at which the mean reward increases appears to be constant over time. This suggests that the agent is on an eventual convergence to an optimal policy. It is important to note that the rewards in this graph are negative. This is because the mountain car task is designed to minimize the number of steps taken to reach the goal. So, a lower negative reward indicates better performance. The innate high perturbations are due to the nature of the agent to forgo some fuel to gain momentum by obtaining more negative rewards.

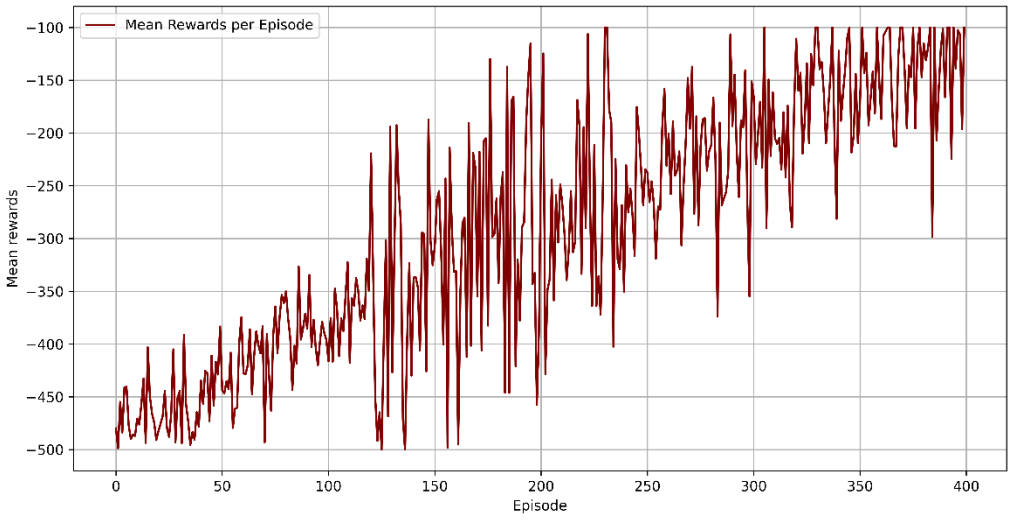

Figure 3 – Rewards vs Episodes – Mountain Car - Policy Gradient-based RL with PQC π

4. Mountain Car - Deep Q-Learning with PQC Q-Function Approximators

The graph **(figure 4)** displays the training performance of the mountain car agent in the OpenAI gym environment using Deep Q-learning with PQC function approximations. The x-axis represents the episode number and the y-axis represents the mean reward per episode.

In the graph, we can see that the mean reward starts around -450 and steadily increases over the course of training episodes. This trend suggests that the agent is successfully learning to control

the mountain car and achieve the goal. The agent appears to converge on a policy that yields a mean reward of around -180 after 800 episodes.

The rate of improvement appears to slow down towards the end of the training, suggesting that the agent may have converged to a near-optimal policy for this task. It's important to note that the rewards in this environment are negative. This is because the goal is to minimize the number of steps needed to reach the goal. So, a higher negative reward represents better performance. Also, the stages of the graph are pretty evident in terms of the exploration, trade-off and exploitation processes.

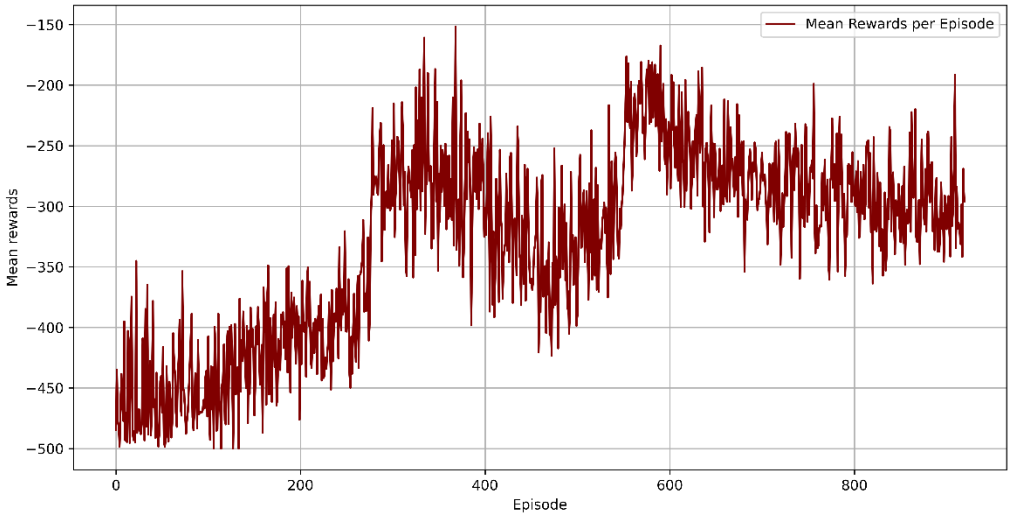

Figure 4 – Rewards vs Episodes – Mountain Car - DQL with PQC QF^n Approximators

IV. FURTHER PLANS

For the final phase of our project, we will develop simulations for the Acrobot environment, a complex scenario characterized by a 6-dimensional state space. Our objective includes implementing all baseline classical reinforcement learning (RL) algorithms across three distinct environments. We will then conduct a detailed comparative analysis between classical and quantum-enhanced RL approaches. This comprehensive evaluation aims to provide a clear understanding of the potential advantages of quantum techniques in reinforcement learning, ultimately identifying the most effective methods. Unveiling the Best Performers!

V. 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