# Abstract

Learning from experience is essential for solving complex challenges: it enables us to recognize patterns, adapt to new situations, and make informed decisions, guiding our actions in both immediate and long-term contexts. This thesis work investigates Reinforcement Learning (RL), a machine learning method where an agent learns by interacting with an environment, using a trial-and-error approach to optimize its actions over time through a reward-based system.

In addition to exploring classical RL, this research examines how quantum computing might enhance RL. Unlike classical computers that represent data in binary bits (0 or 1), quantum computers use qubits, which can exist in a superposition of both 0 and 1 simultaneously. Quantum systems also leverage entanglement, a phenomenon allowing qubits to correlate deeply. These unique properties offer the potential for vastly increased computational capacity compared to classical systems. To evaluate whether quantum machine learning can outperform classical techniques, I implemented an RL algorithm that replaces the traditional neural network with a Parameterized Quantum Circuit (PQC), simulated on a classical computer. This hybrid approach combines classical and quantum processing, leveraging their respective strengths to work within the constraints of current quantum technology. PQC models use a combination of fixed gates (such as controlled-NOT gates) and adjustable gates (such as rotation gates) to encode and optimize complex problems. In this hybrid setup, intensive computational tasks are distributed between classical and quantum processors. For this study, all computations, including quantum simulations, were run on a classical computer: this choice enables to test the quantum algorithm on ideal and foolproof simulated qubits. While actual quantum devices from providers like IBM and Google are available, this study relied on simulation to bypass the limitations of small-scale quantum systems. Current quantum hardware introduces significant noise and operational limitations, including gate and readout errors, decoherence, and limited qubit connectivity, which impact the fidelity of quantum algorithms.

Experiments were conducted in Python using TensorFlow and TensorFlow Quantum libraries, and the Deep Q-Learning algorithm was implemented to estimate state-action values (Q-values) and learn the optimal policy. Both classical and quantum versions of the algorithm were tested within the Acrobot-v1 environment from OpenAI Gym, a widely recognized benchmark for

measuring RL performance in terms of efficiency and accuracy. In this environment, the agent controls a two-link robotic arm connected by a joint and aims to apply torque (positive, negative, or zero) to make the arm swing up and reach a target height. The reward system assigns a -1 penalty for each action that fails to reach the target, while each episode ends either when the arm swings above the target (earning a reward of 0) or after 500 time steps without achieving the goal. The primary goal of this research was to assess whether integrating quantum computing could enhance RL's efficiency and effectiveness compared to classical approaches. To allow this comparison, three parameters were varied: (1) batch size (the number of episodes sampled during training); (2) model expressivity (measured by the number of neurons in hidden layers for the classical model and the number of variational layers in the quantum model); and (3) the learning rate, which controls the speed at which parameters and Q-values are updated.

The results show that the hybrid quantum approach initially learns faster, achieving its peak performance with a reward of approximately -140 within 600 episodes. However, its training was limited by the heavy computational demands of simulating quantum circuits. In comparison, the classical model, trained over 20,000 episodes, converged to a more optimal reward of approximately -70. Over extended training, the classical approach consistently outperformed the quantum model, proving more efficient and producing a more stable policy. Traditional methods remain more effective for practical applications, though quantum computing holds promising future potential. The next phase of the research involves implementing and testing the algorithm on real quantum hardware, using IBM's quantum backend. Testing a quantum reinforcement learning (QRL) algorithm on actual quantum hardware could reduce training time and enable testing over a higher number of episodes and would also let us see how well it performs in real-world conditions. This experiment would help us understand how hardware limitations, like noise and gate errors, affect QRL: by seeing where the algorithm struggles or excels on real devices, we can improve both the algorithm and the hardware, moving us closer to practical and reliable quantum computing systems.