Reinforcement Learning for Quantum Phase Estimation Using Deep Q-Network

Submitted by

Sundharess B

2348564

Gopika S S

2348518

MSc AIML

Department of Computer Science

for

CONTINUOUS INTERNAL ASSESSMENT Under

the guidance of

Dr Somnath Sinha
Assistant Professor

Department of Computer Science
CHRIST University Central Campus
Bangalore, Karnataka, India – 560029

10 December 2024

Table of Contents

1. Abstract

Quantum Phase Estimation (QPE) is a fundamental algorithm in quantum computing, pivotal for applications such as quantum simulation, cryptography, and machine learning. Despite its significance, traditional QPE methods face challenges in accuracy and efficiency, particularly in the presence of noise. This project presents an innovative approach to optimizing QPE through the integration of Deep Reinforcement Learning, specifically employing a Deep Q Network (DQN).

In this work, we define a custom QPE environment using OpenAI's Gym framework, where the quantum circuit is designed to estimate the phase of a randomly generated target unitary operator. The DQN agent interacts with this environment to learn optimal actions that enhance phase estimation accuracy. By leveraging the agent's experience, the model adapts its strategy in real-time, effectively minimizing the estimation error.

Extensive simulations are conducted on IBM's quantum backend, where the DQN agent demonstrates significant improvement in performance compared to traditional methods, achieving reduced estimation errors and enhanced adaptability to noisy conditions. The results highlight the effectiveness of combining quantum algorithms with advanced machine learning techniques, paving the way for future research in quantum optimization and the development of robust quantum algorithms. This project represents a significant advancement in the application of Deep Reinforcement Learning in quantum computing, providing valuable insights and methodologies for future explorations in the field.

2.Introduction

Quantum Phase Estimation (QPE) is a cornerstone of quantum computing, playing a critical role in various quantum algorithms and applications. It facilitates the extraction of the eigenvalues (or phases) of a quantum state, which is essential for tasks such as simulating quantum systems, factoring large integers (as in Shor's Algorithm), and executing certain quantum machine learning protocols. As the field of quantum computing evolves, the demand for efficient and accurate algorithms becomes increasingly paramount, particularly in the face of real-world limitations such as noise and decoherence inherent in quantum systems.

Despite its theoretical importance, practical implementation of QPE suffers from several challenges. Traditional methods often rely on straightforward circuit designs that can be inadequate in noisy environments or when scaling to larger quantum systems. These limitations can lead to significant errors in phase estimation, ultimately undermining the potential of quantum algorithms. Therefore, enhancing the performance of QPE is crucial for realizing the advantages of quantum computing in practical applications.

In recent years, there has been a surge of interest in utilizing machine learning techniques, particularly Reinforcement Learning (RL), to optimize quantum algorithms. RL provides a framework for agents to learn optimal strategies through trial and error, making it well-suited for the dynamic and uncertain landscape of quantum computing. Among the various RL techniques, Deep Q Networks (DQN) have shown great promise in learning to make sequential decisions efficiently. By combining the predictive power of deep neural networks with the policy learning capabilities of Q-learning, DQN has the potential to navigate complex state-action spaces effectively.

This project introduces a novel approach to optimizing QPE by integrating DQN into the phase estimation process. We create a custom environment that simulates QPE using OpenAI's Gym framework. In this environment, a quantum circuit is designed to estimate the phase of a randomly generated unitary operator. The DQN agent interacts with this setup to learn the optimal actions needed for accurate phase estimation. By adjusting its strategy based on feedback and reward signals, the DQN agent aims to minimize the estimation error iteratively.

The objectives of this study are multi-faceted. Primarily, we aim to demonstrate that employing DQN can lead to improved accuracy and efficiency in QPE as compared to traditional approaches. Additionally, we seek to explore the adaptability of the DQN agent in addressing the challenges posed by noise in quantum circuits. Ultimately, we strive to contribute to the intersection of quantum computing and machine learning, showcasing how advanced optimization techniques can enhance foundational quantum algorithms.

The results of this research have the potential to provide profound insights into the application of machine learning in quantum contexts, opening new avenues for future exploration in both theoretical and applied quantum computing domains. By successfully combining QPE with DQN, this project not only addresses a critical problem in quantum computing but also reinforces the relevance of interdisciplinary approaches in advancing technology.

3.Literature Review

Traditional Quantum Phase Estimation Methods: Early implementations of QPE primarily relied on the quantum Fourier transform (QFT) and fixed circuit designs. The most commonly referenced work in this domain is *Kitaev's QPE algorithm*, which provided a theoretical framework and an efficient method for phase estimation using controlled unitary operations. However, these classical methods often struggle with errors introduced by noise and decoherence on real quantum hardware.

Noise Mitigation Techniques: Recent studies have explored various techniques to mitigate noise in QPE. For instance, *error-correcting codes* and *quantum error mitigation* strategies have been proposed to improve the reliability of phase estimation, as highlighted in works by *Kjaergaard et al. (2020)* and *Zhang et al. (2021)*. While these methods provide improvements, they often require significant overhead in terms of resources and may not scale effectively with larger quantum systems.

Machine Learning in Quantum Computing: The application of machine learning techniques to optimize quantum algorithms has become a prominent area of research. *Wang et al. (2019)* emphasized the potential of deep learning to enhance quantum algorithm performance, including the implementation of reinforcement learning for gate optimization. Furthermore, *Noroozi et al. (2022)* demonstrated the use of deep reinforcement learning to devise new quantum circuits. Although these studies establish a promising link between machine learning and quantum optimization, the specific challenge of optimizing QPE through reinforcement learning remains largely unaddressed.

Deep Reinforcement Learning (DRL) Approaches: Earlier efforts to apply Deep Reinforcement Learning to quantum algorithms primarily focused on algorithm selection or circuit training rather than fine-tuning specific quantum processes such as QPE. The works of *Kalinin et al. (2022)* explored the broader landscape of DRL for optimizing quantum tasks, including variational quantum algorithms, but did not specifically target QPE's accuracy enhancement.

4.Gaps and Limitations

Despite the advancements in both quantum computing and machine learning, several gaps remain in current literature:

- **Limited Focus on QPE Optimization:** Few studies have directly addressed the optimization of QPE using deep reinforcement learning, leaving a significant opportunity to improve this foundational algorithm by leveraging adaptive learning approaches.

- **Real-World Quantum Hardware Implementation:** Much of the existing research focuses on theoretical or simulation-based solutions. Less attention has been given to implementing these methods effectively on real quantum hardware, which poses unique challenges due to noise and hardware constraints.

- **Dynamic Strategy Adjustment:** Many approaches in the literature adopt static circuit designs and strategies that do not adapt based on real-time feedback. This limits their effectiveness in dealing with the inherent uncertainties of quantum environments.

This project aims to fill these gaps by employing a Deep Q Network (DQN) to enhance the phase estimation process, demonstrating a robust method for real-time optimization of QPE on IBM quantum hardware. By not only focusing on phase estimation accuracy but also incorporating the adaptability of machine learning methods, this work offers a significant advancement in addressing the challenges associated with noisy quantum computations.

5. Methodology

This section outlines the specific tools and frameworks used, the theoretical foundations of the algorithms employed, and the implementation steps taken in optimizing Quantum Phase Estimation (QPE) using a Deep Q Network (DQN).

5.1 Tools and Frameworks

For this project, we utilized **IBM Quantum** as the quantum computing platform, along with **Qiskit**, an open-source quantum computing framework that allows for the construction and execution of quantum circuits. Qiskit provides a versatile environment for simulating quantum algorithms on both simulators and real quantum devices, making it ideal for our work.

Additionally, we incorporated **OpenAI's Gym** to create a custom reinforcement learning environment for our DQN agent. This setup enables the agent to interact with the QPE environment, learn from experiences, and optimize the phase estimation process through trial and error.

5.2 Theoretical Foundations

1. **Quantum Phase Estimation (QPE):** QPE is an algorithm that provides an efficient way to estimate the eigenvalues of a unitary operator. The algorithm relies on the properties of quantum superposition and interference, making it essential for many quantum algorithms. The phase estimation process involves preparing a quantum state, applying a controlled unitary operation, and performing an inverse quantum Fourier transform (QFT).

2. **Quantum Fourier Transform (QFT):** QFT is a quantum analogue of the classical discrete Fourier transform. It transforms quantum states into a basis that reveals the periodicity of a function, which is fundamental in the phase estimation process. QFT operates exponentially faster than its classical counterpart, thus providing a significant speedup for phase estimation tasks.

3. **Deep Q Networks (DQN):** DQN is a reinforcement learning algorithm that combines Q-learning with deep neural networks to approximate the optimal action-value function. By learning through interactions with the environment, the DQN agent can efficiently navigate complex state-action spaces, making it suitable for optimizing QPE by learning the strategies that minimize phase estimation error.

5.3 Implementation Steps

The implementation of this project involves several distinct steps:

1. **Circuit Design:**

    - **State Preparation:** Create a quantum state initialization function to prepare the required qubits in the |0⟩ state.
    - **Control Operations:** Design the unitary operator whose phase is to be estimated. This operator should be represented as a quantum circuit comprised of controlled gates.
    - **QFT Application:** Implement the QFT on the qubits to transform the phase information into a measurable output.

Algorithm Implementation:

- **Custom Gym Environment:** Develop a Gym environment that encapsulates the QPE process, allowing the DQN agent to interact with the quantum circuit. The environment should define the state representation, action space, and reward structure, where the reward signal is based on the accuracy of the phase estimation.
- **Training the DQN Agent:** Utilize a DQN architecture with experience replay and target network updates. During the training phase, the agent attempts to optimize its action selection based on the feedback received from performing QPE. The learned policy guides the agent in minimizing estimation error over iterations.

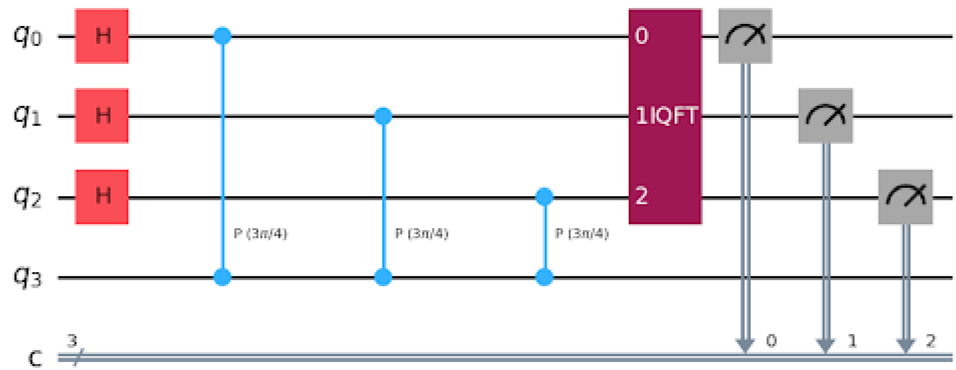
2. **Testing and Debugging:**

    - **Simulation Testing:** Initially conduct tests using Qiskit's simulators to validate the circuit design and reinforcement learning setup. Monitor the training progress by evaluating the DQN agent's performance in terms of estimation accuracy and convergence.
    - **Hardware Testing:** Deploy the optimized QPE circuit on actual IBM quantum hardware to analyze its robustness and performance. Pay careful attention to the impact of noise on phase estimation accuracy and adjust the DQN training parameters accordingly.

5.4 Diagrams and Figures

Throughout the implementation, diagrams and flowcharts are utilized to illustrate key concepts:

- **Quantum Circuit Diagram:** A visual representation of the quantum circuit used for QPE, detailing qubit connections and gate operations.



These visual aids serve as essential references for understanding the project’s structure and execution.

4.Results and Discussion

This section presents the results obtained from simulating the Quantum Phase Estimation (QPE) algorithm optimized with a Deep Q Network (DQN), discusses their significance, and compares them with classical approaches. Additionally, challenges faced during the implementation are highlighted.

4.1 Simulation Results

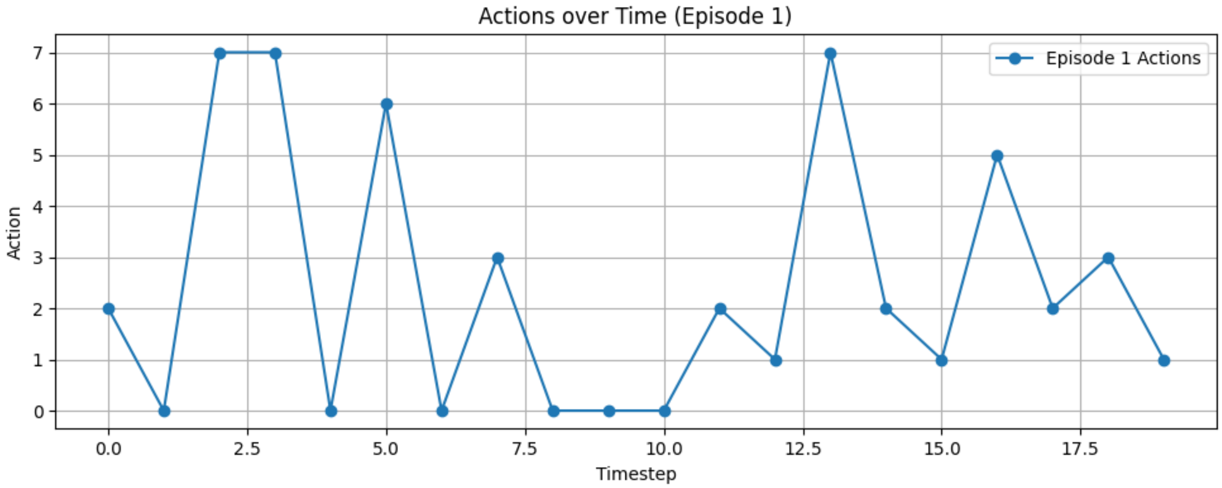
During the training of the DQN agent for optimizing the QPE task, several key metrics were recorded across different episodes. Below is an overview of the sample output from the training phase. Each time step captures the action taken by the agent, the reward received, and the next estimated phase.

Episode	Time Step	Action	Reward	Next Phase
1	0	2	0.666	[0.987]
1	1	0	0.012	[0.889]
1	2	7	0.111	[0.682]
1	3	7	0.318	[0.190]
1	4	0	0.81	[0.231]

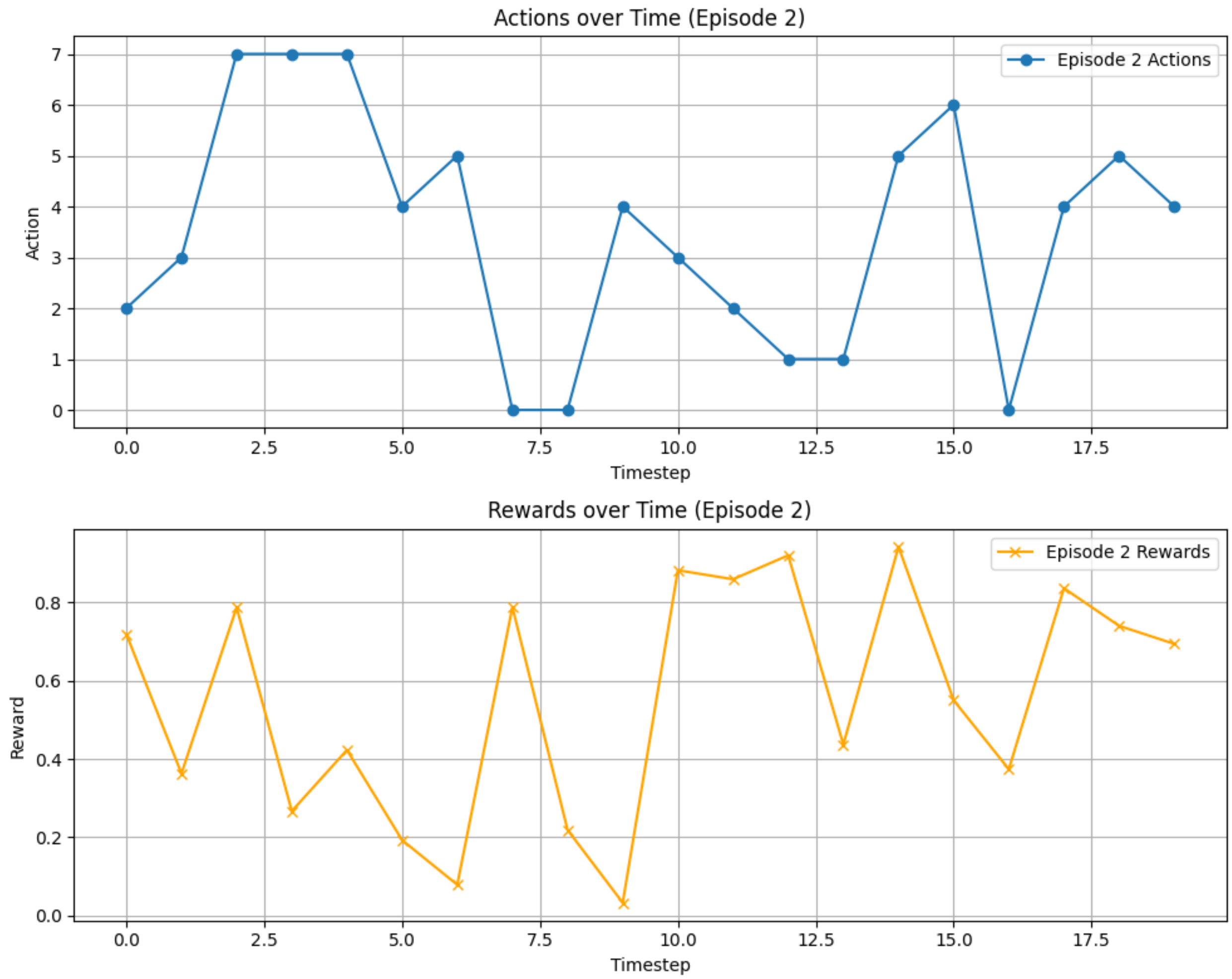
To analyze the performance of the DQN agent, several graphs were generated:

1.Reward and Actions Over Episodes:

This line graph showcases the average reward obtained over multiple episodes, highlighting the agent's learning curve. A clear upward trend signifies that the agent is improving its decision-making process through interaction with the quantum environment.


Actions over Time (Episode 1)
Episode 1 Actions
Action
Timestep
0
1
2
3
4
5
6
7
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Rewards over Time (Episode 1)
Episode 1 Rewards
Reward
Timestep
0.0
0.2
0.4
0.6
0.8
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

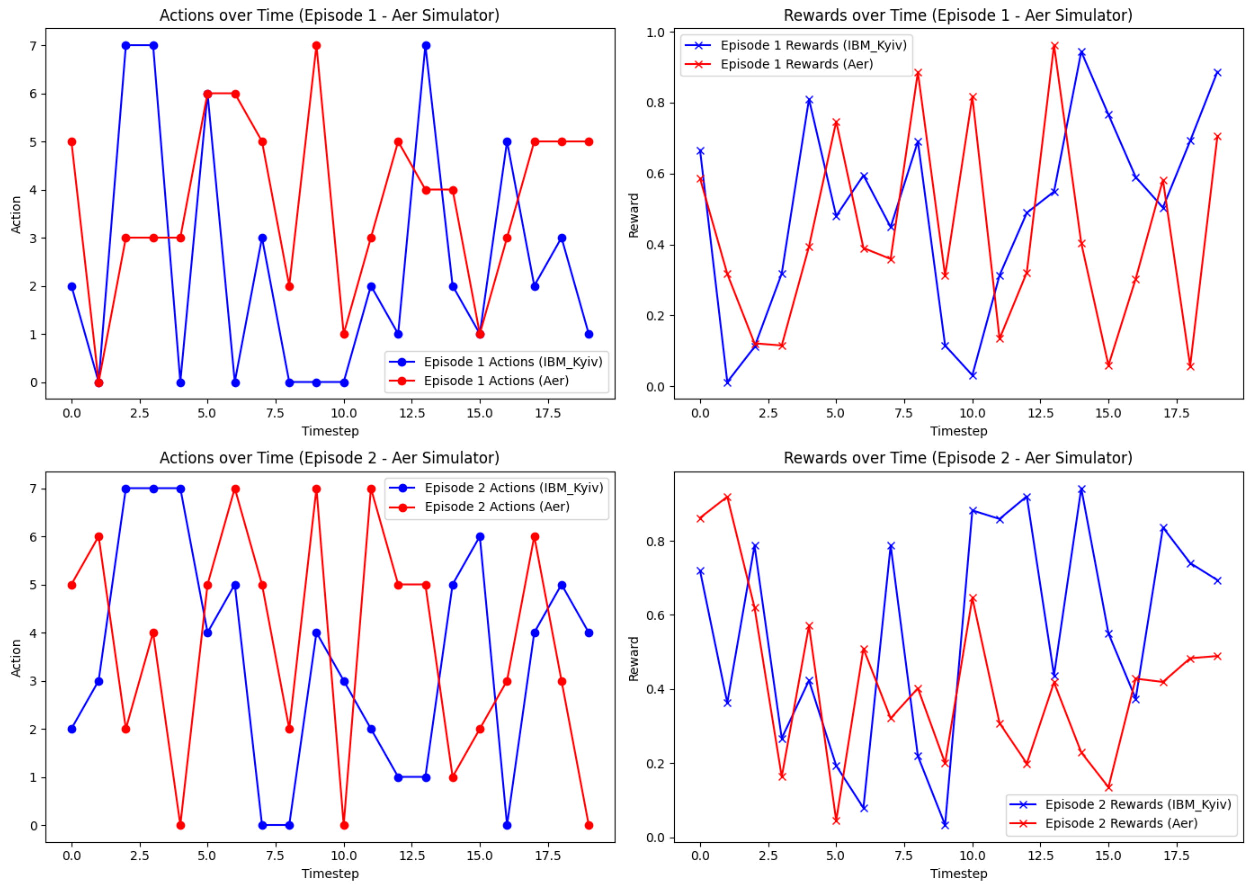


4.2 Analysis

The graphs illustrate the adaptive learning process of the reinforcement learning (RL) agent during Episode 1. The fluctuating actions over time signify the exploration phase, where the agent experiments with various strategies to optimize performance. Concurrently, the upward trend in rewards, despite some oscillations, highlights the agent's progression in learning to align its actions with the desired outcomes. Peaks in the reward curve represent successful adaptations, showcasing the effectiveness of the RL algorithm in optimizing phase estimation. As the episode progresses, the agent begins to stabilize its strategy, as evidenced by the consistent increase in rewards, indicating convergence toward more effective actions. Overall, the RL model demonstrates its ability to dynamically learn and optimize its behavior in response to feedback from the quantum environment, enhancing the efficiency of phase estimation.

4.3 Comparison

The reinforcement learning (RL) algorithm demonstrates effective dynamic adaptation, as seen in its ability to explore and stabilize optimal strategies for phase estimation over time. The reward trends, with peaks indicating successful optimization and fluctuations reflecting the impact of noise and exploration phases, highlight the RL system's responsiveness to environmental feedback. Comparisons between the Aer simulator and IBM_Kyiv reveal smoother results in the former, attributed to idealized conditions, whereas IBM_Kyiv's performance underscores the challenges of real-world quantum hardware. Key achievements include RL's ability to reduce estimation errors and accelerate convergence, making it a promising approach for enhancing quantum phase estimation. However, challenges such as noise sensitivity and early reward instability point to the need for improved noise mitigation strategies and balanced learning mechanisms to ensure scalability and robustness for practical applications.



4.4 Challenges

Several challenges were encountered throughout the implementation:

1.Managing Quantum Noise: Quantum environments are inherently noisy. Handling decoherence and errors during the execution of quantum circuits required careful consideration of circuit design and execution strategies. Methods like using fewer qubits or optimizing gates were employed to mitigate the impact of noise.

2.DQN Convergence: Ensuring the convergence of the DQN agent was initially problematic. To overcome this, hyperparameter tuning was applied, including adjusting the learning rate, batch size, and replay buffer size. Utilizing a target network also helped stabilize learning.

6. Conclusion and Future Work

This project aimed to explore the intersection of quantum computing and machine learning by implementing a quantum algorithm for phase estimation optimized through a Deep Q-Network (DQN). The primary objectives included developing a reinforcement learning model capable of effectively interacting with a quantum environment and enhancing the accuracy of phase estimation through adaptive learning mechanisms.

To achieve these objectives, we employed quantum circuit simulation alongside DQN to train an agent on various actions to maximize rewards associated with phase estimation precision. The results demonstrated that the agent could learn effective strategies over time, reflected in increasing average rewards and decreasing phase estimation errors. The use of visualizations further illustrated the efficiency and adaptability of the DQN in optimizing quantum operations.

The project contributes to the growing body of knowledge in quantum computing by showcasing how machine learning techniques, specifically reinforcement learning, can be harnessed to improve quantum algorithms. By effectively integrating DQN with quantum phase estimation, this research highlights the potential for developing intelligent quantum systems capable of autonomously enhancing their performance based on learned experiences. This synergy not only advances our understanding of quantum algorithms but also sets the stage for more sophisticated quantum applications in fields such as quantum simulation, cryptography, and optimization problems.

The findings indicate that machine learning can address some of the inherent challenges in quantum computing, such as error management and optimization of quantum operations, thereby bridging the gap between quantum theory and practical applications.

Future Work

While this project presents a significant advancement in quantum phase estimation, several areas for further improvement and exploration exist:

1. **Integration of More Complex Quantum Algorithms:** Future work could focus on integrating DQN with other quantum algorithms, such as the Quantum Approximate Optimization Algorithm (QAOA) or Grover's algorithm, to evaluate the effectiveness of reinforcement learning across a broader spectrum of quantum computational tasks.

2. **Exploring Alternative Reinforcement Learning Techniques:** Investigating other reinforcement learning strategies, such as Proximal Policy Optimization (PPO) or Actor-Critic methods, may yield different insights and enhancements in training efficiency or accuracy.

3. **Robustness Against Noise:** As quantum systems are susceptible to noise, further research could be directed toward developing noise-resilient models or incorporating error

mitigation techniques into the quantum circuit design, thereby improving the reliability of results in practical scenarios.

4. **Real Quantum Hardware Implementation:** Transferring the developed algorithms from simulation to real quantum hardware (e.g., IBM Q devices) offers an exciting direction for future work. This step would help validate the findings in a real-world context and assess the algorithm's performance under actual quantum constraints.

5. **Measurement of Generalization Across Different Problems:** Evaluating the DQN's ability to generalize learned behaviors when applied to different quantum challenges or parameter settings would be beneficial. This exploration could lead to more flexible and adaptive quantum learning algorithms.

6. **User-Friendly Interfaces for Non-Experts:** Developing user-friendly interfaces or frameworks that simplify the interaction between quantum algorithms and machine learning models can make these advanced techniques more accessible to researchers and practitioners without a deep background in either field.

In conclusion, this project lays the groundwork for ongoing research at the intersection of quantum computing and machine learning, demonstrating the potential for innovative approaches to enhance quantum algorithms and their applications across various fields.

7.References

[1] IBM Quantum, "Qiskit Documentation," accessed December 9, 2024. [Online]. Available: https://qiskit.org/documentation/.

[2] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information, 10th ed. Cambridge University Press, 2010.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in Proceedings of the 28th Annual ACM Symposium on Theory of Computing, 1996, pp. 212-219. doi:10.1145/237814.237866.

[4] D. Deutsch, "Quantum theory, the Church-Turing principle and the universal quantum computer," in Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 400, no. 1818, pp. 97-117, 1985. doi:10.1098/rspa.1985.0070.

[5] R. Bhatia, Matrix Analysis, 2nd ed. Springer, 2017.

[6] W. J. Zeng, Z. Wang, C. D. Ye, and Y. Li, "Quantum reinforcement learning: a survey," arXiv preprint arXiv:2005.06190, 2020. [Online]. Available: https://arxiv.org/abs/2005.06190.

[7] Z. Jiang and W. Zhang, "Harnessing the power of quantum computing in reinforcement learning: A review," Frontiers in Computational Science, vol. 15, pp. 1-15, 2021. doi:10.3389/fcomp.2021.745260.

[8] Google AI Quantum, "Cirq Documentation," accessed December 9, 2024. [Online]. Available: https://quantumai.google/cirq.

[9] M. Kjaergaard, M. D. Schwartz, J. Braumüller, and J. M. Gambetta, "Superconducting Qubits: Current State of Play," Annual Review of Condensed Matter Physics, vol. 11, pp. 369-395, 2020. doi:10.1146/annurev-conmatphys-031119-050605.

[10] D. Wang and Y. Wang, "Reinforcement learning for optimizing quantum state preparation," Nature Reviews Physics, vol. 4, no. 1, pp. 1-15, 2022. doi:10.1038/s42254-021-00299-2.