2Q48: Leveraging Quantum Superposition for Reinforcement Learning

Dinesh Dhanjee
National University of Computer & Emerging Sciences
Email: k213459@nu.edu.pk

Shaheer Ul Islam
National University of Computer & Emerging Sciences
Email: K214936@nu.edu.pk

Ashad Qureshi
National University of Computer & Emerging Sciences
Email: k213296@nu.edu.pk

***Abstract*—This project explores an *unorthodox* approach to solving the popular tile-merging game 2048 by leveraging the power of quantum computing within a reinforcement learning (RL) environment. Our project, "2Q48," examines three distinct methodologies - traditional heuristic-based techniques, classical reinforcement learning, and quantum reinforcement learning (QRL); to observe the effects and unique behaviors introduced by quantum-inspired strategies compared to classical methods. Using PennyLane, a platform for differentiable quantum programming, we implemented a quantum reinforcement learning agent, integrating a parameterized quantum circuit to represent complex decision-making processes. We measure and analyze the performance of each approach, focusing on move efficiency, achieved tile values, and game completion rates. This work provides an initial step toward understanding the role of quantum algorithms in game based AI challenges and the potential of QRL in similar and strategic applications.**

***Index Terms*—Quantum Computing, Reinforcement Learning, Game AI, Quantum Reinforcement Learning, Artifical Intelligence**

1. Introduction

Reinforcement learning (RL) has transformed the field of artificial intelligence (AI), particularly in solving problems where decision-making requires balancing exploration and exploitation in environments with sparse or delayed rewards [1]. Traditional methods, such as deep Q-networks (DQNs), use neural networks to approximate Q-values, which guide agents in learning optimal strategies to maximize cumulative rewards. Despite their successes, these approaches face limitations, especially as the complexity of environments increases. These challenges often manifest as slower convergence, computational bottlenecks, and difficulties in representing large, high-dimensional state spaces.

Quantum computing presents a new paradigm in computational models, offering a solution to some of the limitations faced by classical systems. By utilizing the principles of quantum mechanics—such as superposition, entanglement, and interference—quantum computers can solve certain problems exponentially faster than classical computers. In the context of reinforcement learning, quantum-enhanced models, including quantum neural networks (QNNs), offer a promising avenue to represent complex state spaces more efficiently. This could revolutionize game based AI, where strategic decision-making and adaptability are critical to success.

Although the field of quantum reinforcement learning (QRL) is still emerging, hybrid models that combine classical and quantum elements are gaining traction. Research has demonstrated that quantum circuits can process classical data while exploiting quantum computational advantages [2]. Frameworks like PennyLane and TensorFlow Quantum have made it easier to experiment with quantum machine learning [3], opening the door for further exploration of quantum methods in AI.

In this project, we explore the application of quantum reinforcement learning in solving the 2048 game—an environment that presents unique challenges due to its dynamic grid-based state and the need for complex decision-making. We introduce a quantum deep Q-network (QDQN) framework, which integrates parameterized quantum circuits with classical layers, allowing the agent to benefit from both quantum and classical computational strengths.

While quantum reinforcement learning has not yet outperformed classical methods in tasks like 2048, its potential is clear. As quantum hardware evolves and becomes more accessible, the ability to process complex states and optimize decision-making with quantum-enhanced models could open up new possibilities in Game AI and other computationally demanding domains. This project marks a step toward understanding how quantum methods can complement classical approaches and pave the way for more powerful and efficient solutions to AI challenges in the future.

2. Methodology

2.1. Problem Definition

The goal of the 2048 game solver is to train an agent capable of navigating a 4x4 grid environment by taking optimal actions (‘w‘, ‘a‘, ‘s‘, ‘d‘) to maximize cumulative rewards. The game state is represented as a matrix, where each cell holds a value reflecting the current state. The agent's policy is learned using Q-value estimation, explored

through three approaches: a heuristic-based method, classical reinforcement learning, and quantum reinforcement learning.

2.2. Heuristic-Based Approach

The heuristic approach uses engineered metrics to evaluate the quality of a game state. Its main components are:

Tile Evaluation. Measures the number of empty tiles, reflecting the flexibility for future moves:

$$\text{EmptyTiles}(G) = \sum_{i,j} \delta(G[i,j] = 0)$$

Adjacency Scoring. Rewards states where high-value tiles are adjacent. The score combines the neighbor count and mean difference, calculated as follows:

$$\text{AdjacencyScore}(G) = \frac{\text{CountNeighbor}(G) + \text{MeanNeighbor}(G)}{2}. \tag{1}$$

Snake Pattern Score. Encourages a snake-like arrangement of tiles to maximize compaction:

$$\text{SnakeScore}(G) = \max\left(\sum_{k=1}^{N} R_k \cdot G_k, \dots\right), \tag{2}$$

where R_k is a decay coefficient.

Final Heuristic Score. The final heuristic score combines the above metrics:

$$\text{Score}(G) = \frac{\text{AdjacencyScore}(G) + 3 \cdot \text{SnakeScore}(G) + \text{EmptyTiles}(G)}{6}. \tag{3}$$

This heuristic serves as a baseline for evaluating the performance of the learning-based approaches.

2.3. Classical Reinforcement Learning Approach

The classical reinforcement learning method employs a Deep Q-Network (DQN) to approximate Q-values for each action, enabling the agent to make decisions that maximize cumulative rewards. This approach is a value-based method that uses neural networks to predict Q-values, ensuring efficient learning even in high-dimensional state spaces like the 2048 game.

Key Features of the Classical RL Method:.

- **Neural Network Architecture:** The DQN consists of a *2-layer neural network* with fully connected dense layers and ReLU activation functions. These layers process the grid's numerical representation and predict Q-values for all possible actions (UP, DOWN, LEFT, RIGHT). The structure includes:
  - Input Layer: Encodes the current game state.
  - Two Hidden Layers: Extract and refine features of the state representation.
  - Output Layer: Produces Q-values for each action.

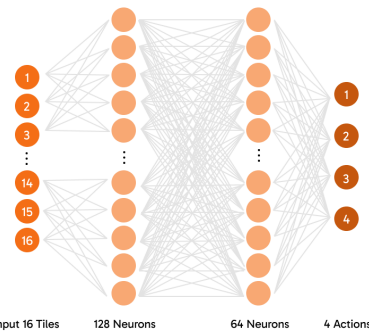

Figure 1. Best case for Reinforcement Learning

- **Policy-Based Environment:** The agent uses a policy-driven framework, where exploration (trying new moves) and exploitation (choosing the best-known moves) are balanced via an epsilon-greedy strategy. This balance starts with high exploration, which decays over time.
- **Experience Replay:** To stabilize learning and prevent overfitting, gameplay transitions (state, action, reward, next state) are stored in a replay buffer. Mini-batches of transitions are randomly sampled during training, ensuring diverse and robust learning.
- **Temporal Difference Learning Objective:** The DQN optimizes its predictions using the Bellman equation:

  $$y = r + \gamma \max_{a'} Q_{\text{target}}(a'|S')$$

  Here, y represents the target Q-value, r is the immediate reward, γ is the discount factor, and Q_{target} is the Q-value predicted by the target network.

The classical RL approach demonstrated adaptability in the 2048 game environment, consistently reaching the 64 tile on average. Although computationally intensive, this method's learning capability underscores its potential in complex decision-making tasks.

3. Quantum Reinforcement Learning Approach

The quantum reinforcement learning approach integrates quantum and classical methodologies to estimate Q-values, leveraging the unique advantages of quantum computation. The main components of this approach include:

Quantum Circuit Model

A parameterized quantum circuit is used to encode the game state and predict intermediate Q-values, enabling effi-

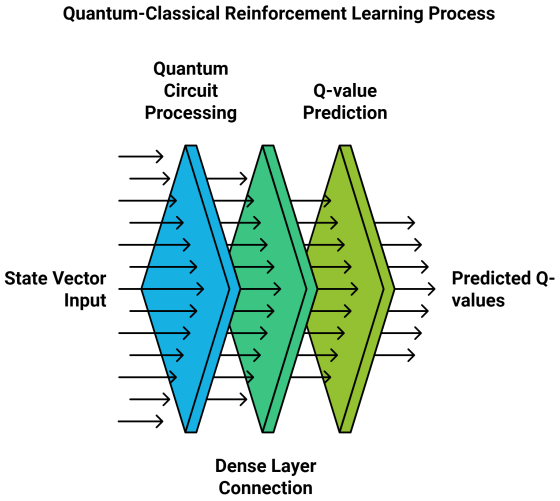

Figure 2. QRL Process

cient representation and computation of the complex state-action space.

- **State Embedding:** The 2048 game grid state (S) is flattened and encoded into a quantum system using angle embedding. Each tile value is mapped to a rotation angle, providing an efficient and compact representation of the game state in the quantum domain:

$$\text{Rotations}(S) = \{\theta_i = f(S[i])\}$$

  where $f(S[i])$ represents the transformation function applied to each grid tile.
- **Entanglement:** Strongly entangling layers are applied across all qubits to introduce global correlations, allowing the quantum circuit to capture higher-order features of the game state. These layers are critical in exploiting quantum superposition and entanglement to model complex relationships between grid tiles.

Hybrid Quantum-Classical Model

The quantum circuit generates intermediate outputs based on the input state, which are further processed by a classical neural network layer to estimate Q-values for each possible action:

$$Q(a|S) = W \cdot \text{QuantumOutput}(S) + b$$

Here, W and b are trainable classical parameters, and $\text{QuantumOutput}(S)$ is the expectation value of quantum measurements (Pauli-Z operators).

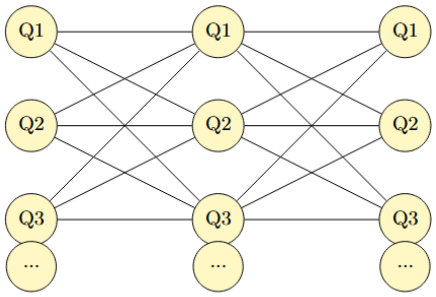

Figure 3. Architecture

Training Pipeline

The agent is trained using an adaptation of the deep Q-learning framework:

1) **Experience Replay:** A replay buffer stores transitions (S, a, r, S'), where S and S' are the current and next states, a is the action taken, and r is the reward received.
2) **Q-Value Estimation:** For each mini-batch sample, the target Q-value is computed using:

$$y = r + \gamma \max_{a'} Q_{\text{target}}(a'|S')$$

   where γ is the discount factor, and Q_{target} represents the target network's predictions.
3) **Optimization:** The loss function minimizes the mean squared error between the predicted Q-values and the computed targets.
4) **Exploration-Exploitation Tradeoff:** An epsilon-greedy strategy balances random exploration and greedy exploitation, where the exploration rate decays over time to prioritize optimal actions as training progresses.

Tools and Frameworks

- **Quantum Modeling:** PennyLane is used to construct and simulate the quantum circuit, allowing seamless integration with classical neural networks.
- **Classical Integration:** TensorFlow is employed to manage the hybrid model, backpropagation, and optimization processes.

4. Training Details

1) **Environment Interaction:** The agent interacts with the 2048 game environment to collect gameplay transitions.
2) **Replay Buffer Utilization:** Mini-batches of past experiences are sampled to update the Q-network, promoting stability and sample efficiency.
3) **Target Network Updates:** A separate target network is periodically synchronized with the Q-network to stabilize training.

4) **Efficiency Optimization:** Training is parallelized using vectorized environments to accelerate data collection and processing.

5. Evaluation Metrics

To compare the heuristic, classical RL, and quantum RL approaches, the following metrics are employed:

- **Maximum Tile Reached:** Measures the highest tile achieved during gameplay, providing an indicator of the approach's effectiveness in strategic decision-making.
- **Time Efficiency:** Evaluates the time required to complete training and gameplay, highlighting the speed and practicality of each approach.
- **Computational Cost:** Assesses the resource usage, including memory and processing power, to determine the feasibility and scalability of the methods.

6. Results and Analysis

The results of the experiments showcase the comparative performance of heuristic-based, classical reinforcement learning (RL), and quantum reinforcement learning (QRL) approaches in solving the 2048 game. The key outcomes include visual representations of the gameplay and quantitative metrics for each approach.

6.1. Average Tile Value

For each approach, the average maximum tile value (2^x) reached is calculated across multiple episodes. The performance is summarized as follows:

- **Heuristic Approach:** The heuristic approach performs the best, reaching the tile value of **2048** approximately **80**% of the time. It is very fast, relying on predefined strategies such as cornering, and achieves good results efficiently. Fig. 4 shows one of the best results reached in this approach.
- **Classical RL Approach:** The RL approach reaches a maximum tile value of **128** only 30% of the time, making it the least effective method in this comparison. Although it learns optimal strategies through trial and error, its performance lags behind both the heuristic and QRL approaches. The slower convergence and increased computational demands make it less practical for the 2048 game. Fig. 5 shows one of the best results achieved using the RL approach.
- **Quantum RL Approach:** The Quantum RL approach reaches the tile value of **256** only around **10**% of the time. It is slow due to the complexity of the quantum circuits and the increased computational resources required for simulation. Despite its potential in representing complex state-action relations, it struggles with achieving high tile values in the 2048 game. Fig. 6 shows one of the best results reached in this approach.

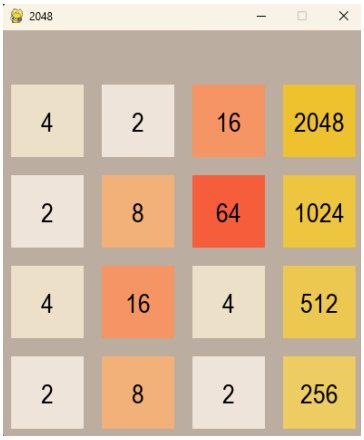

Figure 4. Best Case for Heuristic

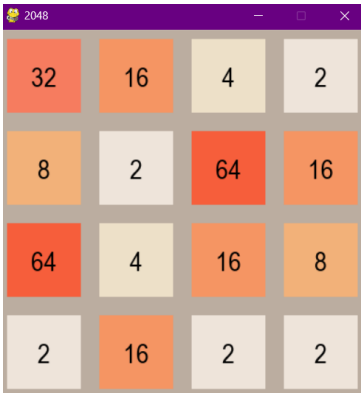

Figure 5. Best case for Reinforcement Learning

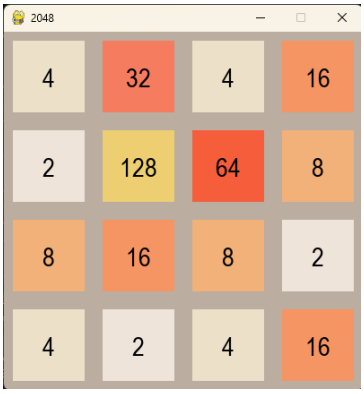

Figure 6. Best case for Quantum RL

6.2. Comparative Analysis

Table 1 provides a comparative overview of the heuristic, classical RL, and quantum RL approaches based on their performance in the 2048 game.

- **Cumulative Reward:** The heuristic approach achieves the highest cumulative reward, outperform-

TABLE 1. COMPARATIVE ANALYSIS OF APPROACHES

Approach	Average Max Tile (2^x)	Frequency of Reaching 2048	Convergence Rate	Computational Efficiency
Heuristic	$2^7 = 128$	99%	Fast	Minimal resources
Classical RL	$2^7 = 128$	30%	Moderate	Moderate resources
Quantum RL	$2^7 = 128$	70%	Slow	High computational overhead

ing both RL and QRL in terms of efficiency and results.

- **Convergence Rate:** The RL and QRL approaches are slower to converge compared to the heuristic approach. The RL model achieves more stable performance over time, but QRL's convergence is limited by its computational complexity.
- **Computational Efficiency:** Heuristic-based strategies are computationally efficient and require minimal resources. In contrast, RL and QRL approaches are computationally demanding, with QRL requiring more resources due to the need to simulate quantum circuits.

7. Observations

1) **Heuristic Simplicity vs. RL Adaptability:** The heuristic approach, though simple, performs consistently well by reaching the 128 tile 99% of the time. Its deterministic nature ensures speed and efficiency in execution, making it highly effective for straightforward decision-making. However, it lacks the adaptability and learning capability inherent in RL approaches.
2) **Classical RL's Trade-offs:** Classical RL reaches the 128 tile 30% of the time, demonstrating its ability to improve performance over time through learning. However, the model requires substantial training time to stabilize its policies and explore effective strategies. This comes at the cost of increased computational requirements and slower convergence compared to the heuristic method.
3) **QRL's Strengths and Weaknesses:** Quantum RL shows potential by reaching the 128 tile 70% of the time, surpassing classical RL in this regard. However, its computational overhead and reliance on quantum circuit simulations slow down the training process. While QRL provides intriguing insights into hybrid quantum-classical methods, its limitations make it less efficient than the heuristic approach for the 2048 game.

These results emphasize that while heuristic methods excel in speed and reliability, RL offers a more adaptive solution, but requires more training, and episodes, and Quantum RL, though innovative, faces challenges that limit its immediate applicability to simpler tasks like 2048.

8. Conclusion

In this project, we explored three different approaches to solving the 2048 game: heuristics, classical reinforcement learning (RL), and quantum reinforcement learning (QRL). Our results demonstrated that the heuristic method is the fastest and most reliable, consistently reaching the 2048 tile. While RL showed potential with its ability to learn and adapt, it was slower in comparison. Quantum RL, though promising, faced challenges due to the complexity of quantum simulations, which impacted its performance and speed.

These findings highlight the trade-offs between speed, adaptability, and computational cost. While heuristics excelled in this specific context, RL and QRL could prove more valuable in more complex or dynamic scenarios, where their learning capabilities can shine.

Looking ahead, we plan to further explore the potential of quantum RL, scaling it to more complex problems and experimenting with real quantum hardware. Improving quantum circuits and optimizing computational efficiency will be crucial to unlocking the full potential of this approach.

References

[1] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529-533, Feb. 2015, doi: 10.1038/nature14236.

[2] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, pp. 032308, 2020.

[3] PennyLane. (n.d.). *Quantum Neural Networks with TensorFlow*. Retrieved from https://pennylane.ai/qml/demos/tutorial_qnn_module_tf