Cyber Defence in the Realm of Quantum Computing and Reinforcement Learning

Evan Moore, Tim Xia

April 7, 2025

1 Introduction

1.1 Context

This project explores the applications of quantum reinforcement learning (QRL) in cybersecurity, specifically using a hybrid classical-quantum reinforcement learning (RL) approach to prevent two trains from colliding. This QRL example problem is an extension on the work of Barbeau and Garcia-Alfaro [1], and has been expanded in various ways. RL is a computational technique in which a system learns to perform a specific task by interacting with an environment. An agent receives rewards for its actions and optimizes future actions to maximize its reward [2]. As such, QRL leverages quantum circuits for a wide range of benefits over classical RL [1].

We leverage the power of variational circuits (VCs), commonly known as variational quantum circuits (VQCs), a quantum computing approach that uses tunable parameters which are applied to the circuits. These parameters are optimized through a cost function, which evaluates how effectively the quantum state encodes the agent's prediction of the correct action based on classical data. The cost function guides the gate adjustment, enabling the VQC to iteratively improve its prediction [3]. Quantum properties can be exploited for a variety of benefits. Examples of this include entanglement, which can be used to allow the agent to comprehend more states, or superposition to allow the system to represent a large amount of information simultaneously.

1.2 Problem Statement

With the increasing frequency of security breaches in network controlled systems (NCSs), the future of cyber-security is in question. The rise of quantum computing and the threats it poses to security only heighten these concerns. Barbeau and Garcia-Alfaro assert that integrating quantum tools into security strategies is crucial for effectively defending against quantum-powered adversaries [1].

The primary goal of this project is to explore the practical applications of QRL in addressing emerging challenges. As mentioned above, the objective is to achieve this by creating an adapted version of the two-train simulation from [1]. The paper seeks to develop a quantum-assisted RL system that trains one of the agents, represented by a train in the simulation, to avoid collision with the other train, which serves as the adversary.

The two trains move at the same speed, so they must avoid collisions by selecting different paths along the track. The paper uses the track design proposed by Barbeau and Garcia-Alfaro in [1], which features a crossroad with two paths, and also develops a more complex version incorporating a third path. This addition necessitated an expansion of both our reward system and VQC. Specific implementation details will be discussed in later sections.

By using a simple illustration like a two-train collision avoidance system, this paper seeks to demonstrate the effectiveness of QRL. Similar to the original paper found in [1], that this approach should be explored further as part of a potential defense against quantum-enabled attackers targeting NCSs.

1.3 Result

This paper has successfully implemented a version of the two-train system outlined in Barbeau and Garcia-Alfaro's paper [1], with several additions and modifications. It has incorporated multiple modes for the adversary, such as fixed behavior (always taking the pass or loop), random behavior, and even QRL for the adversary in an attempt to cause a collision. Additionally, it expands on the system by introducing a track with three paths, resulting in three different states. This required modifying the VQC to accommodate two qubits and entanglement. The simulation is capable of running under the same three modes as the original system. In modes where only the agent employs QRL, we observe that the distance between the trains increases over time, indicating the agent's success in avoiding the adversary.

This paper begins with general research on RL, along with related quantum concepts. Starting with Barbeau and Garcia-Alfaro's paper to gain a broad understanding [1], we eventually turned to the PennyLane documentation for information on variational quantum circuits [4]. It is important to note that we used content from the README file of the GitHub repository mentioned in the original paper to get an initial idea of how our future implementation should resemble [5]. However, the end result deviates greatly, which will be shown in future sections. The code of this project uses Python for the general coding language, PennyLane for quantum content, and Q-Learning as the classical aspect of our hybrid classical quantum approach.

1.4 Outline

The remainder of this report is structured as follows. Section 2 provides a more in-depth dive into the background information and relevant concepts to this project. Section 3 presents the results obtained from our implementation of the system in much more detail. Section 4 discusses the results and performance of the system, including a performance analysis. Finally, Section 5 offers a conclusion and suggestions for future work.

2 Background Information

Building a QRL model requires an understanding of several complex concepts. The following section provides a detailed breakdown of these practices.

2.1 Reinforcement Learning

RL is a branch of machine learning where an agent learns to make decisions by interacting with an environment. The agent receives feedback in the form of rewards and aims to maximize its cumulative reward over time [2].

An RL problem can be modeled as a Markov Decision Process (MDP) as described in [1], which consists of the following components:

- **State:** A specific situation the agent encounters in the environment.
- **Actions:** The set of possible moves the agent can make from a given state.

- **Transition Probability:** The likelihood of moving from one state to another after taking a specific action.
- **Reward:** The feedback received after performing an action in a particular state.

2.2 Quantum Computing and Variational Circuits (VCs)

VCs are quantum circuits with trainable parameters, typically optimized to solve problems [3]. In the context of RL, VCs can model action probabilities by encoding classical information into quantum states using the following gates:

- **RX Gate:** Rotates a qubit around the X-axis.
- **RY Gate:** Rotates a qubit around the Y-axis.

Additionally, a CNOT gate is applied to entangle qubits if the circuit allows more than a single-qubit input, allowing the system to represent and explore a broader solution space simultaneously.

2.3 Scenarios and Variational Circuit (VQC) Design

In the QRL implementation, two main scenarios were explored. The initial goal was to replicate the work of Barbeau and Garcia-Alfaro [1], which used a single-qubit VQC. After successfully reproducing this model, the work was extended to a more complex system involving a multi-qubit VQC.

2.3.1 Scenario 1: Single-Qubit VQC

As shown in Figure 1, the first model uses a single qubit encoded through rotation gates, without any entanglement. The qubit is rotated around the X and Y axes, and a measurement is performed to derive action probabilities. This model is limited to environments where the agent has only two possible actions.

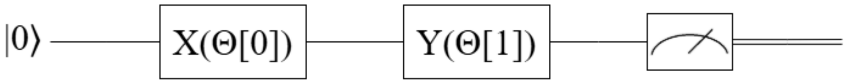


Figure 1: Single-qubit Variational Quantum Circuit (VQC) without entanglement.

2.3.2 Scenario 2: Multi-Qubit VQC

Figure 2 shows the improved multi-qubit model. After encoding the classical data, CNOT gates are applied to entangle the qubits. This design enables the model to handle more complex environments, where an agent may have three possible actions available in any given state.

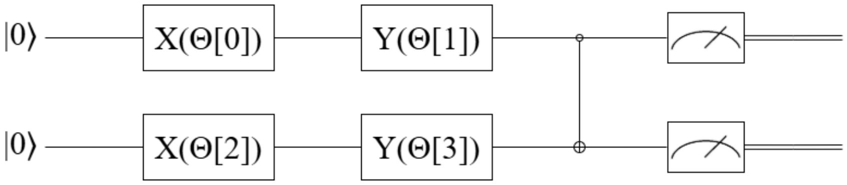


Figure 2: Multi-qubit Variational Quantum Circuit (VQC) with entanglement.

2.4 Q-learning and Its Integration with VCs

Q-learning is a RL algorithm that learns an optimal policy by updating a Q-table. Each entry in the Q-table, $Q(s, a)$, represents the expected reward for taking action a in state s. The Q-values are updated using Bellman's equation:

$$Q(s, a) = Q(s, a) + \alpha \left[r + \gamma \max_{a'} Q(s', a') - Q(s, a) \right], \tag{1}$$

where:

- α is the learning rate,
- γ is the discount factor,
- r is the reward received,
- s and s' are the current and next state,
- a is the action taken.

In QRL, the classical Q-table is used to encode state-action values, which are then used to train the VQC to approximate Q-values.

2.5 Training the Variational Circuit

The VQC is trained by minimizing the Mean Squared Error (MSE) between its predicted outputs and the target Q-values:

$$\text{MSE} = \frac{1}{n} \sum (X_i - Y_i)^2, \tag{2}$$

where:

- X_i are the predicted Q-values,
- Y_i are the target Q-values,
- n is the number of samples.

Gradient descent is used to optimize the VQC's parameters, iteratively adjusting them to minimize the prediction error. Squaring the values amplifies larger errors, allowing the system to prioritize correcting them.

3 Result

3.1 Core Quantum Machine Learning (QML) System

To understand the results of the project, one must start with the core QML implementation. All required background knowledge is listed in the previous section. Initially, the system consisted of a single-qubit VQC and a classical component in the form of expected rewards and state-action pairs from a Q-table.

The process begins with hyperparameter tuning, the process of defining arrays of commonly used Q-learning parameters, and running the simulation across multiple epochs with all possible combinations. The combination that yields the best performance, measured by the largest average distance between the two trains, is considered the most optimal [6].

Using the most optimized parameters set, the simulation is run over a finite number of epochs. In the initial epoch, the agent train is guaranteed to engage in exploration, the process of selecting random actions and recording the rewards in an effort to find optimal strategies. As training progresses, the probability of exploration gradually decreases, favoring exploitation instead, the act of selecting actions based on the agent's current knowledge to maximize expected rewards. This balance between exploration and exploitation is fundamental in reinforcement learning, as described by Wang et al. in [7]. As our QML approach is a hybrid between quantum and classical implementations, this idea remains valuable [1].

After updating the Q-table with Bellman's equation, it effectively acts as a growing library of the estimated desirability of each action explored. The VQC's parameters are then updated to ensure that its output probabilities mimic the target distribution provided by the Q-table. Over time, the VQC learns to imitate the Q-table's policy, allowing the model to leverage the power of quantum computing in RL such as superposition to evaluate multiple states, which in theory would increase training time massively, and allow the representation of more complex policy spaces [8].

3.2 Two-State Track Implementation and Results

Using this QML model, this paper first addresses the original agent-versus-adversary problem as defined in [1]. As aforementioned, two trains travel at the same speed on a one-way track. The agent's goal is to avoid the adversary by selecting different routes at an intersection. The track layout is illustrated in Figure 3. The agent starts at node 0, while the adversary begins at node 7. At node 2, the trains must choose between two paths: the bypass, which leads down to node 8, or the loop, which continues straight through to node 3. After the agent reaches node 0, the distance between the two trains are recorded to update the Q-table, and consequently the VQC.

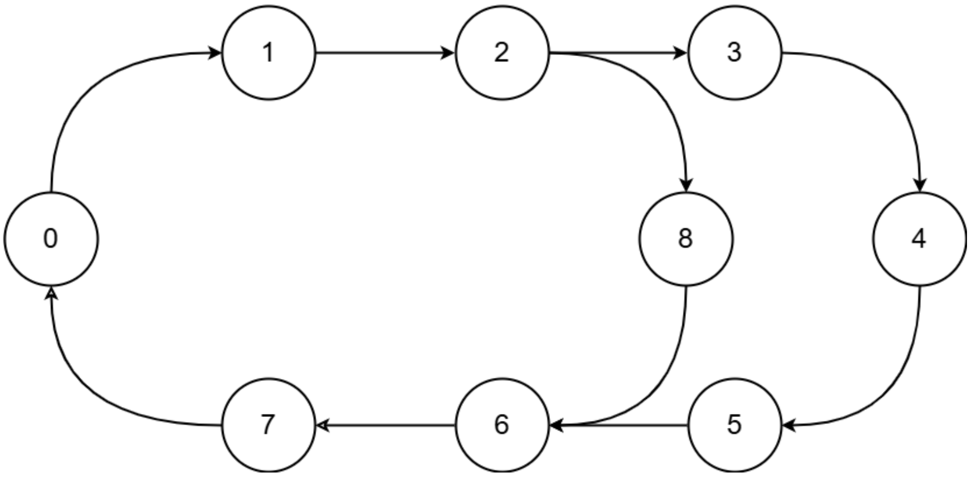
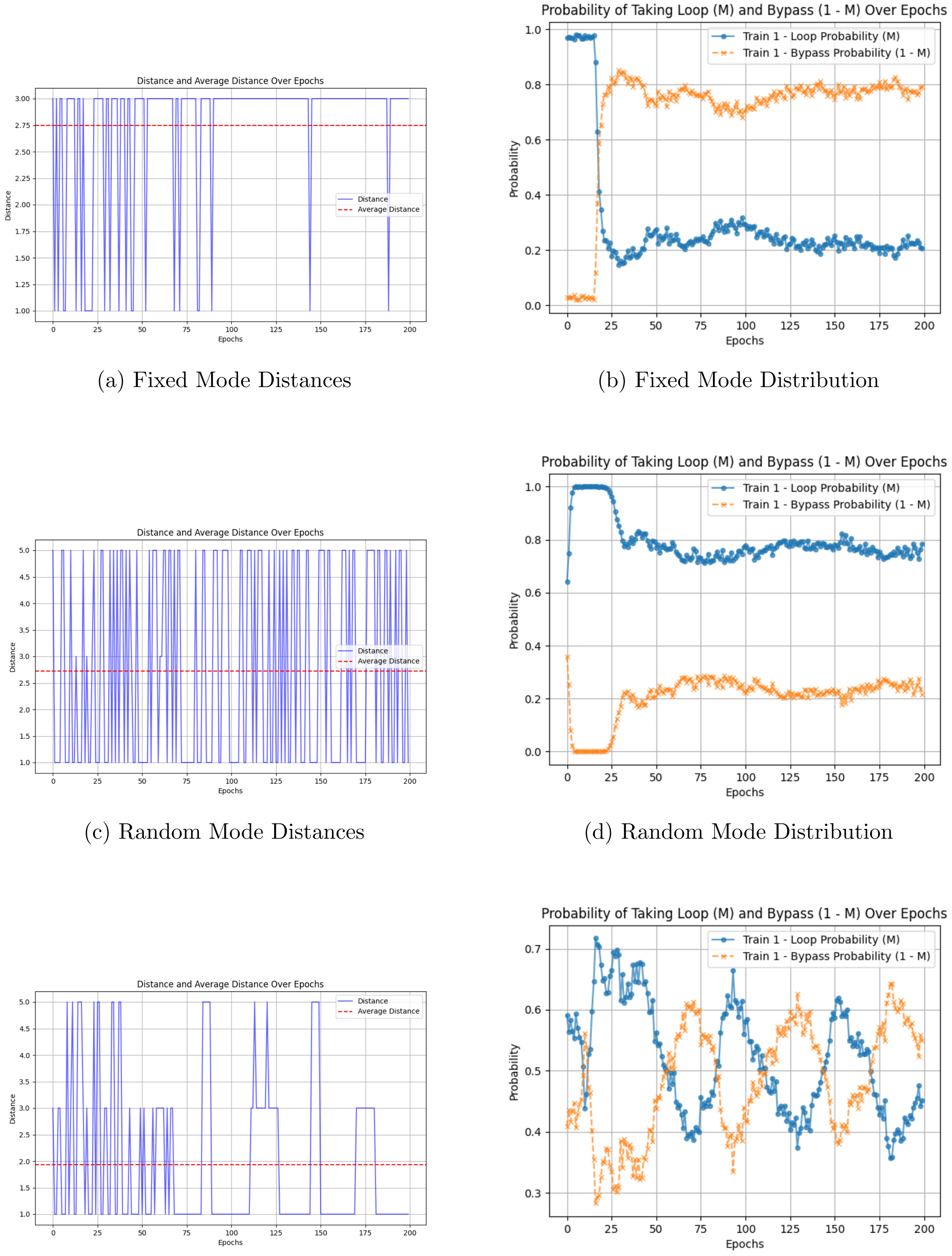


Figure 3: The two-state track



(a) Fixed Mode Distances

(b) Fixed Mode Distribution

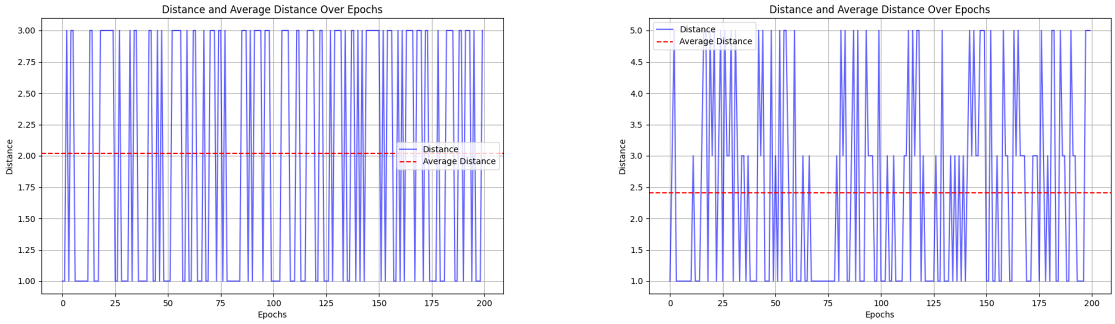
(c) Random Mode Distances

(d) Random Mode Distribution

(e) QRL Mode Distances

(f) QRL Mode Distribution

Figure 4: Choice Distribution and Distances for Fixed, Random, and QRL Modes



(a) Fixed Control Mode Distances

(b) Random Control Mode Distances

Figure 5: Control Results for Distances in Fixed and Random Modes

3.3 Three-State Track Implementation and Results

So far, this paper has extended the simulation from Barbeau and Garcia-Alfaro [1] by incorporating a QRL mode. To further expand on the original problem, this paper introduces a new path branching off from node 2, creating a three-state system. This new path, referred to as the outer-loop, is a longer route compared to the previous loop. The updated track is shown in Figure 6.

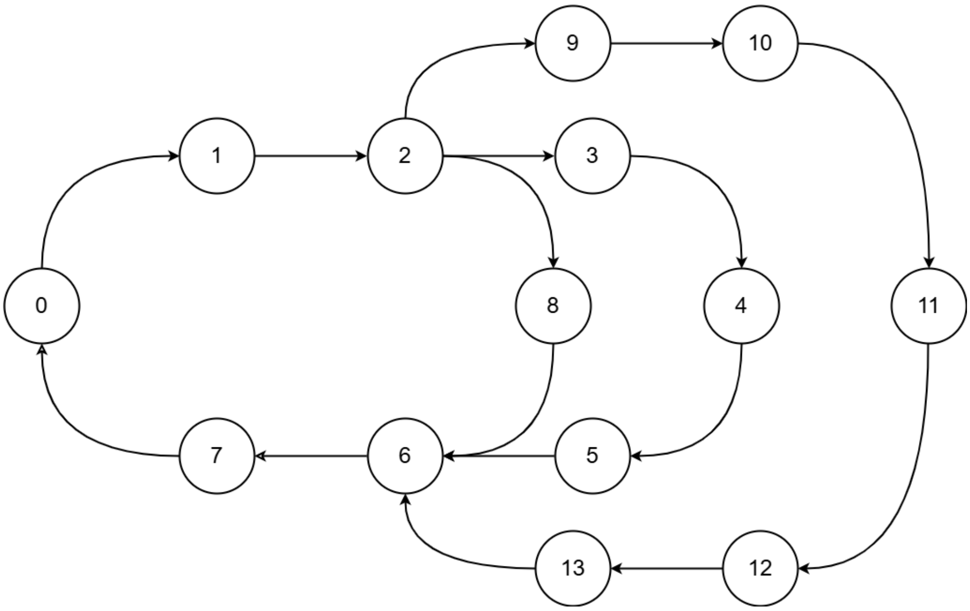
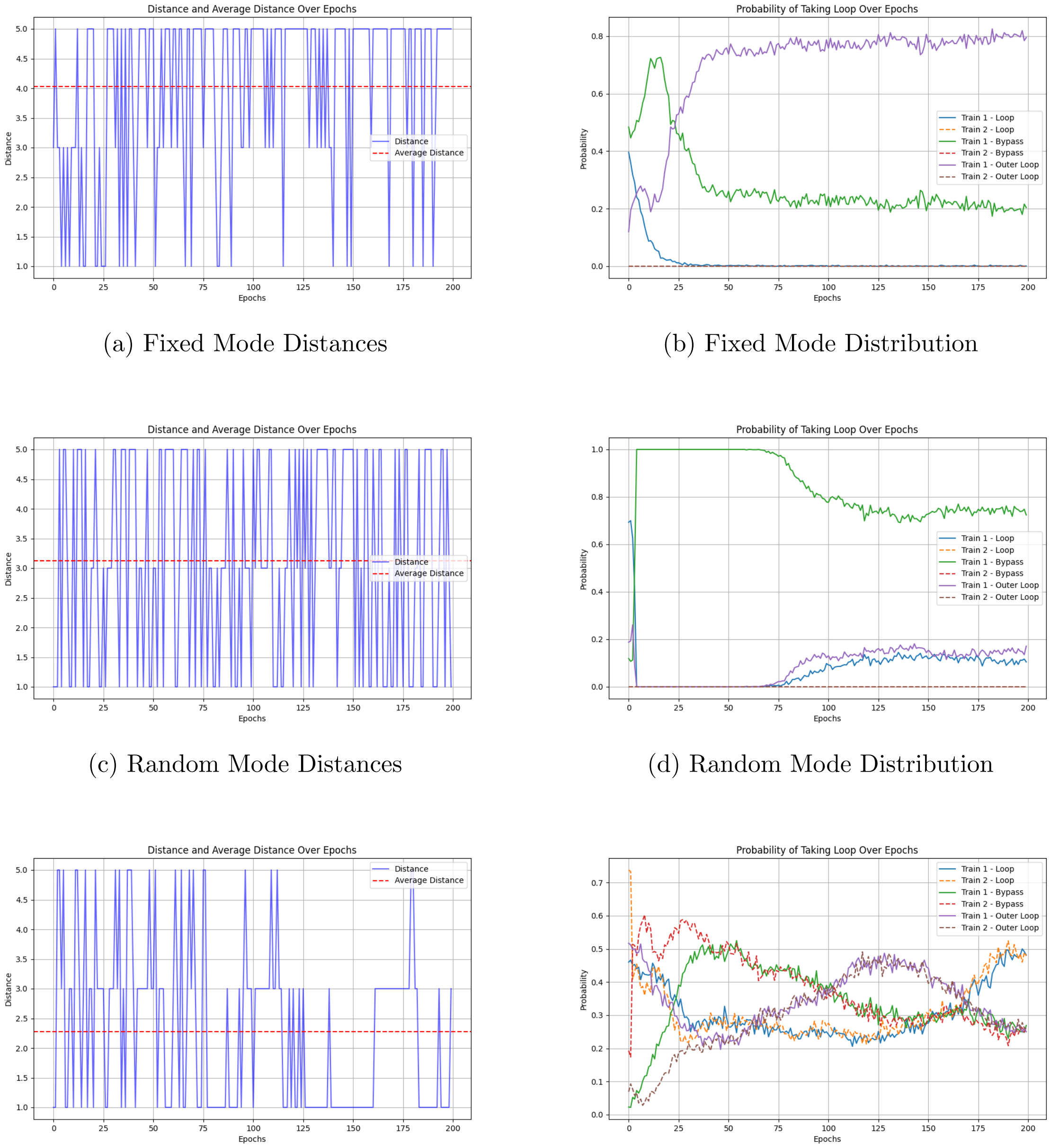


Figure 6: The three-state track

Although our core QML implementation remained the same, this scenario requires us to use a 2-qubit VQC, presented in section 2.5.2. The results for the same three modes from before: fixed, random, and QRL mode are shown in Figure 7, with similar controls in Figure 8 on the following page.



(a) Fixed Mode Distances

(b) Fixed Mode Distribution

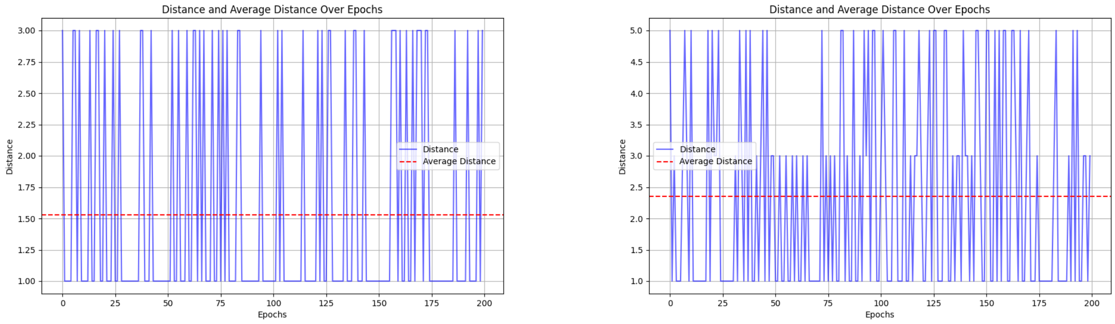
(c) Random Mode Distances

(d) Random Mode Distribution

(e) QRL Mode Distances

(f) QRL Mode Distribution

Figure 7: Choice Distribution and Distances for Fixed, Random, and QRL Modes, three-state track



(a) Fixed Control Mode Distances

(b) Random Control Mode Distances

Figure 8: Control Results for Distances in Fixed and Random Modes, three-state track

4 Evaluation

4.1 Two-State Track Evaluation

In the two-state track configuration (Figure 3), the simulation was run using three adversary modes: fixed, random, and QRL adversary. The goal of the agent (Train One) was to maintain a safe distance from the adversary (Train Two) by selecting paths that minimized collision risks.

In the fixed adversary mode (Figures 5a and 5b), Train Two followed a deterministic path by always taking the loop. Train One successfully learned an optimal policy, by being able to maintain an average distance of 2.75 by selecting the bypass path approximately 80% of the time. This mode achieved the highest average distance across the three adversary modes.

In the random adversary mode (Figures 4c and 4d), Train Two selected paths randomly, introducing more uncertainty. Here, Train One adapted by favoring the loop path approximately 80% of the time, the opposite of its behavior in the fixed mode. Although the average distance slightly decreased compared to the fixed mode, the agent demonstrated its ability to adjust its policy based on the adversary's actions which reflects the effectiveness of this model.

In the QRL adversary mode (Figures 4e and 4f), the adversary used QRL to minimize the distance to Train One, creating the most challenging environment. As expected, the average distance dropped to just below 2, which is the lowest among all the modes. Figure 4f also shows frequent fluctuations in action preferences almost every 25 epochs, demonstrating the learning process occuring between Train One and Train Two.

To ensure the QRL model's effectiveness, a performance study was done where the results of the previous three scenarios were compared with a control which is where both trains acted randomly (Figures 5a and 5b). In these cases, the average distances were lower than where Train One was trained with QRL as the distance for fixed adversary mode went from 2.75 to 2.00 and the random adversary mode went from just less then 2.75 to just less then 2.50. This demonstrates the Train One learned effective collision avoidance policies compared to random behavior by using this QRL model.

4.2 Three-State Track Evaluation

The three-state track configuration (Figure 6) introduced additional complexity by adding a third path called the outer-loop. To accommodate the larger action space, the agent's quantum system was expanded from a single-qubit to a two-qubit variational quantum circuit (VQC).

Across the different adversary modes (fixed, random, and QRL), the trends are similar to those shown in the two-state track results. In the fixed adversary mode, the agent maintained an average distance of approximately 4 as shown in 8a. The distribution of the action probabilities showed that in this case the outer loop was the best action to take as increasingly over epochs it trended towards that move.

For the random adversary mode the average distance maintened between Train One and Train Two was approximately 3.25 as shown in 7c. The action that the agent learned to be its best action was the bypass (7d).

Similarly to the QRL mode in a two state track configuration, the average distance was the lowest among the various adversary modes, with an average distance of approximately 2.25 (see 7e). The probability distribution shown in 7f shows again the fluctuations between which move is correct which again just shows both trains trying to adapt based off the action of the other train.

There was a performance study also done for a three-state track which reiterates the effectiveness of QRL, with random agents achieving significantly lower distances of around 1.00 to 2.50 (see 8a). These results highlight the agent's ability to learn and adapt even in more challenging environments when leveraging quantum-enhanced strategies.

4.3 General Observations

Overall, the results validate QRL as a promising approach for collision avoidance in dynamic multi-agent environments [1]. Against fixed or random adversaries, the QRL agent consistently learned effective policies, outperforming random baselines and maintaining safe distances. Even when facing a learning QRL adversary, the agent was able to maintain a minimum distance of at least two, showing that it learned an effective policy.

Despite the increased complexity of the three-state track, the agent successfully adapted to the complex scenario. This suggests that QRL can extend beyond simple environments to handle more realistic scenarios, such as those encountered in real-world NCSs.

Through our train simulation, we illustrate the potential of integrating QRL into the control and security layers of NCSs. Such integration could produce agents capable of learning effective policies, even in the presence of quantum-enabled adversaries.

5 Conclusion

5.1 Summary

This project demonstrates the strong potential of quantum reinforcement learning as a powerful tool for addressing dynamic challenges, particularly in systems vulnerable to quantum-powered adversaries like NCSs. By applying a hybrid classical-quantum learning approach to the two-train problem introduced by Barbeau and Garcia-Alfaro [1], this paper explores how QRL can aid real-time decision-making in the face of threats to security and safety. Building on their original work, this paper expanded the action space by introducing a third track and increased the complexity of the simulation with adaptive QRL adversary behavior [1].

While the system exhibited some imperfections, the overall results from both the two-track and three-track scenarios demonstrate that the QRL agent successfully learned safe policies in a variety of scenarios. This conclusion is supported by the consistently greater average distance between trains in our system compared to the control scenarios. Although the QML-enabled adversary was able to reduce the gap more effectively, the distance never dropped below 1.00, the minimum following distance, highlighting both the strength and necessity of quantum-powered defense mechanisms against quantum-powered advanced adversaries.

5.2 Relevance

The assignment outline specifies that the project must incorporate topics in machine learning, communications, or security, enhanced by quantum techniques. In alignment with this objective, this work focuses on a machine learning system enhanced through quantum computing. Specifically, the use of a variational quantum circuit to support RL. The quantum component enables parallel evaluation of multiple state-action spaces with quantum superposition, which in theory, could reduce training time. Moreover, VQCs allows for efficient function approximation, enabling QML systems to generalize better in complex environments, even with relatively simple Q-table representations [1].

The project also contributes to strengthening the team's quantum knowledge, particularly through the application of entanglement in designing a two-qubit variational quantum circuit (VQC). Additionally, it has presented an opportunity to work on a practical application of quantum computing in security.

5.3 Future Work

There are several ways this current system can be improved. The primary direction is to further refine the agent's learning process to ensure it consistently selects the most optimal action. For instance, in the two-state track under the random mode, the agent correctly identified that the loop yielded a higher average reward. However, over 200 epochs, it did not fully converge on always choosing the loop, opting for it only the majority of the time. Enhancing the system's learning consistency would undoubtedly lead to greater train separation, increasing the integrity and security of the system.

Another direction for expanding this system is to enable the agent to handle more complex and dynamic problems. At present, the environment is relatively simple—even a human can easily determine the optimal track, as the correct choice remains constant. To increase complexity, the plan is to incorporate the adversary's position from the previous iteration into the current state. This means the optimal choice will vary based on context, leading to a more chaotic state and reward landscape. Exploration of this approach has already begun, and while the system has struggled in most simulations, there have been instances where the agent successfully learned an optimal policy that outperformed the controls. The next step is to analyze these successful cases to understand what led to effective learning and how we can replicate and improve those conditions.

Lastly, the aim is to improve the system by increasing the number of states and testing its scalability. This would involve designing more complex VQCs to handle a richer set of inputs and decisions. For instance, we could introduce tracks with multiple intersections and diverging paths, making collisions possible even in the first iteration. Such scenarios would require the agent to react to immediate threats and be heavily penalized for early mistakes. This level of complexity would likely follow the enhancements discussed earlier, where the agent is first trained to handle dynamic and context-sensitive environments. Overall, the progress made so far is a source of pride, and there remains optimism about the potential for future development.

Contributions of Team Members

Evan Moore contributed equally to the core QML system, developed the QML mode, and implemented the result visualization. Their work includes VQC updates via gradient descent, hyperparameter tuning, and system refinement. They wrote Sections 2 and 4 of the report.
Tim Xia also contributed equally to the core QML system, developed the train simulation, control scenarios, the 3-state track systems, and implemented loss, cost, and update functions for the VQC. They wrote Sections 1, 3, and 5 of the report.

References

[1] M. Barbeau and J. Garcia-Alfaro, "Cyber-physical defense in the quantum era," *Scientific Reports*, February 2022. Available: `https://doi.org/10.1038/s41598-022-05690-1`.

[2] MATLAB, "What is reinforcement learning?," 2024. Available: `https://www.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html`.

[3] PennyLane, "Variational circuits," 2025. Available: `https://pennylane.ai/qml/glossary/variational_circuit`.

[4] P. Developers, "Pennylane," 2025. Available: `https://pennylane.ai/`.

[5] M. Barbeau and J. Garcia-Alfaro, "Dl-poc: Readme," 2022. Available: `https://github.com/jgalfaro/DL-PoC?tab=readme-ov-file`.

[6] I. Belcic and C. Stryker, "What is hyperparameter tuning?," 2016. Available: `https://www.ibm.com/think/topics/hyperparametertuning`.

[7] H. Wang, T. Zariphopoulou, and X. Zhou, "Exploration versus exploitation in reinforcement learning: a stochastic control approach," 2019. Available: `https://arxiv.org/abs/1812.01552`.

[8] N. Schetakis, D. Aghamalyan, P. Griffin, *et al.*, "Review of some existing qml frameworks and novel hybrid classical–quantum neural networks realising binary classification for the noisy datasets," *Scientific Reports*, vol. 12, no. 1, p. 11927, 2022. Available: `https://doi.org/10.1038/s41598-022-14876-6`.