Autonomous Cyber Defence by Quantum-Inspired Deep Reinforcement Learning

Wenbo Feng, Sanyam Vyas and Tingting Li
School of Computer Science and Informatics, Cardiff University, United Kingdom
{*fengw4, vyass3, lit29*}*@cardiff.ac.uk*

Keywords: Autonomous Cyber Defence, Reinforcement Learning, Quantum Computing

Abstract: With the rapid advancement of computing technologies, the frequency and complexity of cyber-attacks have escalated. Autonomous Cyber Defence (ACD) has emerged to combat these threats, aiming to train defensive agents that can autonomously respond to cyber incidents at machine speed and scale, similar to human defenders. One of the main challenges in ACD is enhancing the training efficiency of defensive agents in complex network environments, typically using Deep Reinforcement Learning (DRL). This work addresses this challenge by employing quantum-inspired methods. When coupled with Quantum-Inspired Experience Replay (QER) buffers and the Quantum Approximate Optimization Algorithm (QAOA), we demonstrate an improvement in training the defence agents against attacking agents in real-world scenarios. While QER and QAOA show great potential for enhancing agent performance, they introduce substantial computational demands and complexity, particularly during the training phase. To address this, we also explore a more practical and efficient approach by using QAOA with Prioritised Experience Replay (PER), achieving a balance between computational feasibility and performance.

1 INTRODUCTION

Within the field of cybersecurity, the interaction between defenders and attackers is fundamentally imbalanced. Defenders must remain in a constant state of vigilance, identifying and responding to every potential threat, while attackers need only to succeed once to achieve their objectives. This significant disparity highlights the urgent need for sophisticated and adaptable defences that can promptly and comprehensively counter attacks. AI offers promising opportunities to develop such defences, particularly through Autonomous Cyber Defence (ACD) using Reinforcement Learning (RL) and Game Theory. The aim of ACD is to train defensive agents which can autonomously react to cyber incidents like human defenders. These agents are expected to not only detect malicious behaviours in real-time but also execute advanced defensive actions such as system hardening, isolating, deploying decoys and recovery at machine speed and scale.

Deep Reinforcement Learning (DRL) has been widely used to design and train such defensive agents (Vyas et al., 2023; Shen et al., 2024) to learn optimal policies for strategic response in dynamic and adversarial environments. However, while facing a complex network environment, it is very challenging to train defensive agents with traditional DRL efficiently. Based on that, this work aims to enhance the performance of DRL with quantum computing methods in order to further accelerate the training of defensive agents. Specifically, we use Quantum-Inspired Experience Replay (QER) to optimize exploration and empirical replay techniques in DRL, and we utilize the Quantum Approximate Optimization Algorithm (QAOA) to improve the training efficiency of defensive agents.

We demonstrate our approach using a set of realistic scenarios built in the OpenAI Gym interface from the well-known autonomous defence competition CAGE Challenge (Standen et al., 2022). It allows us to rigorously assess and analyse the proposed quantum-inspired approach. The key innovative contributions of this work are summarised as follows:

- Optimizing Experience Replay: This work enhances traditional DRL algorithms in ACD by introducing Quantum-inspired Experience Replay, improving storage and retrieval efficiency using quantum computing features.
- QAOA in Defence Training: The QAOA is used to train and test defensive agents in a quantum computing environment, integrating it with CybORG,

a research platform by OpenAI Gym for training autonomous agents. We optimize QAOA parameters to boost the effectiveness of defensive strategies in cybersecurity.

- QAOA and MDP Integration: QAOA is combined with Markov Decision Processes to improve decision-making for defensive agents. Quantum states represent MDP states, parameter adjustments are actions, and optimization outcomes serve as rewards, achieving synergy between quantum and classical computing.

In the following sections, we begin with a discussion of underpinning technologies in Section 2, followed by the integration of quantum-inspired methods into ACD in Section 3. Relevant results of the proposed approach are presented in Section 4. The paper concludes with a discussion of the limitations and potential directions for further research in this area.

2 Related Work

In this section, we discuss the key technologies and methodologies relevant to our work, focusing on Deep Reinforcement Learning (DRL) in game-theoretic contexts for autonomous network defence and the role of Replay Buffer techniques.

2.1 DRL Based Game Theory for Autonomous Network Defence

Intelligent game countermeasure technology plays a critical role in ACD particularly through the use of DRL algorithms to tackle sequential decision-making problems in adversarial environments. One of the earliest approaches to applying DRL in game-theoretic models is the Least Squares Policy Iteration (LSPI) algorithm which was expanded by (Lagoudakis and Parr, 2012) to include zero-sum Markov games. This work demonstrated the effectiveness of this method in various scenarios, and illustrated the challenges and advantages of using value function approximation in Markov games, which induced further exploration into applying DRL in competitive environments.

Markov Games have been utilised in several domains of cyber security operations to provide a framework for modelling adversarial scenarios. For instance, Benaddi *et al.* (Benaddi et al., 2022) developed a stochastic game model that incorporates Markov Decision Processes (MDP) to improve decision-making in intrusion detection systems (IDS) and to analyse the behaviour of IDS. Using a Partially Observable Markov Decision Process (POMDP) and recurrent-aided DQN, Liu *et al.* (Liu et al., 2021) introduced a network defence framework that dynamically converges to optimal defence tactics in the presence of partial rationality and imperfect knowledge. The applications demonstrate the adaptability and efficacy of Markov Games in modelling and resolving network defence problems.

The integration of DRL with Markov Games has demonstrated promising results in autonomous network defence, with DRL methods such as Double DQN, PPO, and A3C, showing significant improvement in policy optimization for adversarial settings. Double DQN, enhanced with experience replay and target networks has effectively addressed training instability, rendering it well-suited for dynamic network defence scenarios. Similarly, PPO is the preferred option for continuous control tasks for security applications due to its resilience and effectiveness in policy optimization, ensuring more effective defence mechanisms in complex, adversarial environments.

2.2 Replay Buffer

Experience Replay, commonly referred to as Replay Buffer, is an essential element in many DRL architectures. An agent's experiences (state, action, reward, next state and done flag) during interactions with the environment are stored in the replay buffer. A significant benefit of using a replay buffer is its ability to break temporal connections between successive interactions, which ensures the stability of the learning process and improves sample efficiency. Several variations of the replay buffer, including Prioritized Experience Replay (PER) (Schaul et al., 2015), Hindsight Experience Replay (HER)(Andrychowicz et al., 2017b), and Quantum-Inspired Replay (QER) (Wei et al., 2021), have been developed to further optimize the learning process.

Schaul and colleagues (Andrychowicz et al., 2017a) proved the efficacy of Prioritized Experience Replay (PER) using the Atari 2600 benchmark suite. The implementation demonstrated notable improvements in learning efficiency and performance compared to the traditional uniform sampling approach through prioritising more informative transitions. This mechanism directs the learning process towards the most valuable experiences resulting in faster convergence.

Andrychowicz *et al.*(Andrychowicz et al., 2017a) introduced the HER algorithm in robotic manipulation tasks, including block stacking and fetch reach. In these tasks, the robot acquires knowledge to accomplish objectives by considering unsuccessful attempts as successes in attaining other goals. Imple-

menting this method greatly improved sample efficiency and success rates in tasks with few rewards, demonstrating its practical usefulness in solving complex tasks with limited feedback.

In our work, we utilised QER to enhance the performance of DRL to train the defensive agents by manipulating quantum information. More details are provided in Section 3.1

2.3 Exploration-Exploitation Policy

The Exploration-Exploitation Policy is key to building an effective DDQN algorithm. This work primarily uses the ε-greedy policy and Boltzmann strategy (Cercignani and Cercignani, 1988).

The ε-greedy policy selects the action with the highest Q-value most of the time but explores by choosing a random action with probability ε. This balances exploiting known optimal actions and exploring new ones, addressing the risk of being trapped in suboptimal solutions due to inaccurate Q-value estimations (Hasselt et al., 2016):

$$a = \begin{cases} arg\max_a Q(s,a) & \textit{with prob. } 1-\varepsilon \\ \text{random action} & \textit{with prob. } \varepsilon \end{cases}$$

Here, ε controls the exploration probability but does not consider the relative Q-values of actions, leading to equally random choices even for slightly suboptimal actions.

The Boltzmann strategy improves exploration by using a probability distribution proportional to Q-values, introducing more informed action selection. It incorporates a temperature parameter τ to control randomness, with the selection probability given by: $P(a|s) = \frac{\exp\left(\frac{Q(s,a)}{\tau}\right)}{\sum_b \exp\left(\frac{Q(s,b)}{\tau}\right)}$. Higher τ increases randomness, while lower τ approaches greedy behaviour. This method prioritizes higher Q-value actions and smoothens the probability mapping.

We found that combining ε-greedy with Boltzmann yields better results. ε-greedy alone struggles when ε drops to 0.01, as exploration becomes insufficient and score optimization becomes inconsistent. A hybrid approach, where Boltzmann is applied with a predefined probability when ε is minimal, enhances performance and ensures more consistent updates.

3 Quantum-Inspired Autonomous Defence Agents

In this section, we first discuss the Quantum-inspired Experience Replay (QER), which is followed by the other important component of our approach – Quantum Approximate Optimization Algorithm (QAOA). We then discuss how they were integrated with Markov Game and contribute to our final approach.

3.1 QER

Quantum-inspired experience replay (QER) combines ideas from quantum computing with DRL to make better use of experience samples and boost the performance of traditional experience replay buffers by manipulating quantum information. By representing and manipulating quantum states, QER allows RL models to select and process training samples more efficiently, thereby accelerating convergence and improving policy performance (Wei et al., 2021).

In QER, each empirical sample is represented as a quantum state. Specifically, an empirical e_k can be represented by the state of a quantum bit (*qubit*) which is: $|\psi^{(k)}\rangle = b_0^{(k)}|0\rangle + b_1^{(k)}|1\rangle$ Here, $b_0^{(k)}$ and $b_1^{(k)}$ are two probability magnitudes indicating the likelihood of the empirical sample being rejected or accepted, respectively. This quantum state satisfies the normalization condition: $|b_0^{(k)}|^2 + |b_1^{(k)}|^2 = 1$, where $b_0^{(k)}$ and $b_1^{(k)}$ can be initialized and adjusted based on the quality of the experience, e.g., Temporal Differential (TD) error, making the selection of experience quantum-inspired.

QER introduces two key quantum operations to dynamically adjust the probability magnitude of empirical samples: *the preparation operation* and *the depreciation operation*.

The preparation operation aims to increase the selection probability of empirical data, targeting samples with higher TD errors. Specifically, the Grover iteration algorithm, a classical method, is used to sample the quantum state of experience. This quantum algorithm effectively amplifies the probability amplitude of the target state. Each iteration updates the quantum state of the empirical sample by using the following rotation matrix:

$$U_\sigma = \begin{pmatrix} \cos(\sigma) & -\sin(\sigma) \\ \sin(\sigma) & \cos(\sigma) \end{pmatrix}$$

where σ is a rotation angle usually dynamically adjusted according to the empirical TD error. Through multiple iterations, the probability magnitude $b_1^{(k)}$ of empirical samples with higher TD errors will be significantly increased, prioritizing these samples for selection during playback.

The depreciation operation aims to prevent certain empirical samples from being overused in training, i.e., to prevent overfitting caused by excessive re-

play. Whenever an empirical sample is used, the depreciation operation reduces its selection probability through a quantum rotation operation, allowing other samples to be selected. The devaluation operation is realized by the following rotation matrix:

$$U_\omega = \begin{pmatrix} \cos(\omega) & -\sin(\omega) \\ \sin(\omega) & \cos(\omega) \end{pmatrix}$$

where ω is a depreciation factor that decreases as the empirical samples are replayed more frequently. This operation ensures that the selection of empirical samples is diversified and representative, preventing the model from falling into local optima.

In practice, QER integrates these quantum operations into the experience replay buffer. Whenever an empirical sample is selected for training, a depreciation operation adjusts its quantum state to reduce the probability of future selection. Conversely, if an empirical sample has a high TD error, the preparation operation increases its probability of being selected. The introduction of these quantum operations enhances the utilization of empirical samples, ensuring that the model can fully explore the environment while efficiently leveraging important samples during training, thereby improving overall learning efficiency and policy performance. The specific implementation of QER is shown in Figure 1. This framework integrates quantum principles into DRL by means of QER. Starting with raw Experience data, the experiences are encoded into a Quantum representation of experience (Step 1), followed by a *Preparation operation* that generates a Superposition state with an amplitude (Step 2). In Step 3, a Mini-batch of these quantum experiences is sampled from Quantum Composite Systems as a Buffer. After interaction with the Environment (Step 4), the agent computes a new TD-error, which is used to update the amplitudes of the quantum state through a second *Preparation operation* (Step 5). This quantum-enhanced replay mechanism improves the efficiency and effectiveness of agent's learning in DRL systems.

3.2 QAOA

QAOA is a hybrid quantum-classical variational optimization method designed to solve combinatorial optimization problems. It combines quantum state evolution with classical optimization algorithms to approximate the optimal solution by tuning a series of parameters. Current noisy intermediate-scale quantum (NISQ) devices work well with the algorithm, and it can provide effective approximate optimization solutions in complicated quantum systems. The main idea behind QAOA is to use a set of controlled

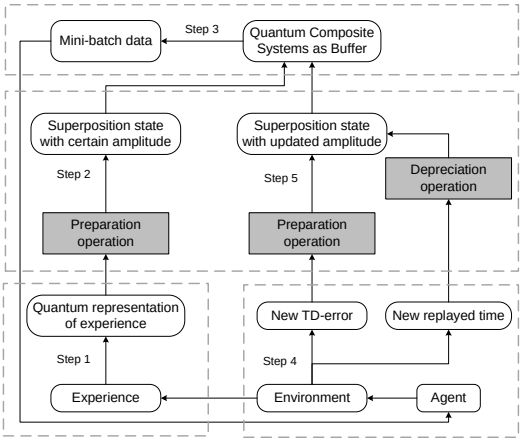


Figure 1: Framework of Deep Reinforcement Learning with Quantum Experience Replay(QER) (Wei et al., 2021)

quantum gate operations on a quantum state and then change the parameters of these operations to use a classical optimization algorithm to find the smallest objective function (Zhou et al., 2020). Specifically, the **Problem Hamiltonian Volume** is defined as $\hat{H}_z = -J\sum_{j=1}^{N}\sigma_j^z\sigma_{j+1}^z$ where σ_j^z is the **Pauli-Z** operator of the Jth quantum bit. $\hat{H}_x$ denotes **Yokohama Hamiltonian Volume** $\hat{H}_x = -\sum_{j=1}^{N}\sigma_j^x$ where σ_j^x is the textitPauli-X operator of the Jth quantum bit. Based on that, the **Quantum State Evolution** can be defined as below

$$|\psi_P(\gamma,\beta)\rangle = \left(\prod_{t=1}^{P} e^{-i\beta_t\hat{H}_x}e^{-i\gamma_t\hat{H}_z}\right)|+\rangle$$

Amongst them, $\gamma = (\gamma_1,\gamma_2,\ldots,\gamma_P)$ and $\beta = (\beta_1,\beta_2,\ldots,\beta_P)$ is a $2P$ real number parameter. $\hat{H}_z$ is the problem Hamiltonian, whose ground state is the solution sought. $\hat{H}_x$ is the transverse field term used to drive the quantum state evolution. As a result, the **variational energy** is denoted as:

$$E_P(\gamma,\beta) = \langle\psi_P(\gamma,\beta)|\hat{H}_z|\psi_P(\gamma,\beta)\rangle$$

where $|\psi_P(\gamma,\beta)$ is the quantum state after the P-round operation. The optimization is carried out using QAOA. As shown in Algorithm 1, the quantum state evolves through successive applications of the problem and mixer Hamiltonians. γ and β are iteratively adjusted to minimise the cost function $E_P(\gamma,\beta)$.

3.3 Combination of QAOA and MDP

In the combination of the QAOA and MDP, the parameter optimization problem of QAOA is naturally integrated into the MDP framework, enhancing the decision-making capability of the defensive agent in

Data: Initial parameters γ_0, β_0, number of steps P
Result: Optimized quantum state minimizing $E_P(\gamma,\beta)$
Initialize the quantum state $\psi_0(\gamma_0,\beta_0)$;
for $t = 1$ *to* P **do**
  Update state with $\hat{H}_z$: $|\psi_t'\rangle = e^{-i\gamma_t\hat{H}_z}|\psi_{t-1}\rangle$;
  Update state with $\hat{H}_x$: $|\psi_t\rangle = e^{-i\beta_t\hat{H}_x}|\psi_t'\rangle$;
  if $E_P(\gamma,\beta)$ *not minimized* **then**
    Adjust parameters γ, β;
    Optimize $E_P(\gamma,\beta)$;
  end
  else
    Continue to the next iteration;
  end
end

Algorithm 1: Quantum State Evolution & Optimization

complex network environments. Through this combination, QAOA not only relies on a fixed quantum computational process but also flexibly utilizes classical RL algorithms for adaptive optimization, thereby enabling more effective defence strategies.

In this framework, the quantum states in the QAOA are considered as "states" in the MDP. These quantum states can usually be described by amplitude or probability distributions. These states carry all the current information about the system, and each state reflects the configuration and evolutionary outcome of QAOA at a particular step.

Actions in MDP are represented in QAOA as the adjustment of QAOA parameters γ_t and β_t at each step. Each action involves choosing specific values for γ_t and β_t in a given state, guiding the quantum state's evolution towards a more optimal state. The action space is therefore a multidimensional continuous space, where each dimension represents a degree of freedom in the QAOA parameters.

State transfer in QAOA is realised through specific quantum operations. These operations correspond to applications of classical quantum gate operations or Hamiltonian terms. In the MDP framework, each action (i.e., the choice of γ_t and β_t) leads to the current quantum state $|\psi_P(\gamma,\beta)$ to evolve to the next quantum state. This evolution follows the Schrödinger equation and is guided by the design principles of QAOA.

In the combination of QAOA and MDP, the reward function is typically related to the optimization result of the objective function. Specifically, the reward in the MDP is designed as the negative objective function value $-C(\gamma,\beta)$ where $C(\gamma,\beta)$ is the distance or difference between the quantum state and the target state. This design transforms optimizing the QAOA parameters into maximizing the cumulative reward.

By combining this with MDP, we can view the parameter optimisation process of QAOA as a strategy optimisation problem. We can use classical reinforcement learning algorithms like Q-learning or Proximal Policy Optimisation (PPO) to learn and optimise these strategies. Through continuous iteration, the system can select the optimal values in each state, maximizing the cumulative reward and enabling the effective use of quantum computing.

In the MDP, the termination condition of QAOA can be set to occur after all steps are completed or when the objective function reaches a preset optimal solution. By stopping the QAOA after achieving the optimal quantum state, this termination condition enhances the computing efficiency and effectiveness.

The algorithmic overview is provided in Algorithm 2 to illustrate the structured approach to optimise policy and state value parameters iteratively.

Input: Initial policy parameters θ_0, initial state value parameters $V^{\Pi}_{\theta_0}$
Output: Optimized policy parameters θ^*, optimized state value parameters $V^{\Pi}_{\theta^*}$
for $k = 0,1,2,\ldots$ **do**
  Sampling N_{epi} **episodes:**
  for $i = 1$ **to** N_{epi} **do**
    Initialize state S_1
    for $t = 1$ **to** P **do**
      Sample action a_t from policy $\Pi_{\theta_k}(a_t|O_{t-1})$
      Observe new state S_t and reward r_t
      Update state $S_t = |\psi_{t-1}\rangle$ using environment dynamics $e^{-i\beta_t\hat{H}_x}e^{-i\gamma_t\hat{H}_z}|\psi_{t-1}\rangle$
    end
    Store episode $(S_1,a_1)^i,\ldots,(S_P,a_P)^i$ and rewards $r_1^i,\ldots,r_P^i$
  end
  Policy update:
  Update policy parameters θ_{k+1} using collected episodes
  State value (SV) update:
  Update state value parameters $V^{\Pi}_{\theta_{k+1}}$ using collected episodes
end

Algorithm 2: Quantum Reinforcement Learning

4 Results and Evaluation

4.1 Experimental Settings

We used CAGE challenge 2 scenarios (Kiely et al., 2023) to evaluate our improved models in CybORG (Baillie et al., 2020). This challenge requires developing a blue agent to autonomously defend a network against a red agent. A typical network is constructed with three subnets: **Subnet 1** (non-critical user hosts), **Subnet 2** (enterprise servers), and **Subnet 3** (a critical

operational server plus three user hosts).

Each episode has a fixed number of steps. At each step, both red and blue agents choose actions from a high-level list. CybORG then instantiates and executes these actions, determining their real-world effects. High-level attack actions include *Discover Remote Systems, Discover Network Services, Exploit Network Services, Privilege Escalation*, and *Impact*, each instantiated with details like IPs, ports, and sessions.

As shown in Figure 2, each run starts with the red agent controlling a host in Subnet 1. The red agent then performs reconnaissance, exploits enterprise servers (Subnet 2), escalates privileges, and finally tries to impact the operational server (Subnet 3). Two red agents were used: *B_line*, which randomly selects attacks, and *Meander*, which methodically compromises each subnet in turn.

The blue agent monitors hosts and can terminate red access or restore systems. Restoration halts the red agent but disrupts users, and the red agent remains on the foothold host, simulating persistent threats. The blue agent can also deploy decoys; if the red agent escalates on a decoy, it fails and is removed, forcing it to exploit a real service.

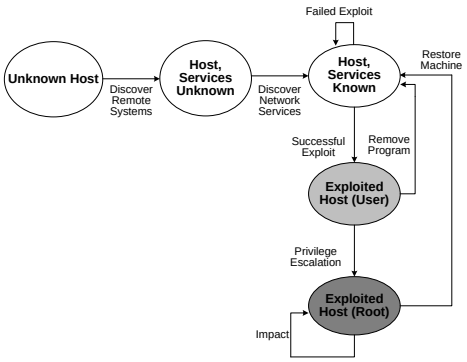


Figure 2: Effect of high-level actions on host state. Contextual information instantiates each high-level action, determining the impact of each attack (Kiely et al., 2023)

The reward function penalizes the blue agent based on the red agent's access level, with the heaviest penalty if the operational server is impacted. There is also a penalty for restoring hosts, discouraging simple recovery strategies and encouraging more strategic and stable defense.

The evaluation method is based on the criteria provided by the CAGE challenge in terms of trial lengths and red agents. We ran experiments with various trail lengths: 30, 50, and 100 steps. Different types of red agents were implemented: *Meander* (explore randomly), *B-line* (moves directly to the operational server), *Sleep* (no action). For each combination of trial length and red agent, CybORG is executed over 10 episodes, leading to a total of 1,000 episodes, with the blue agent's total reward recorded and presented in Table 1. We used Intel Core i7-8750H with 16 GB RAM for all our experiments.

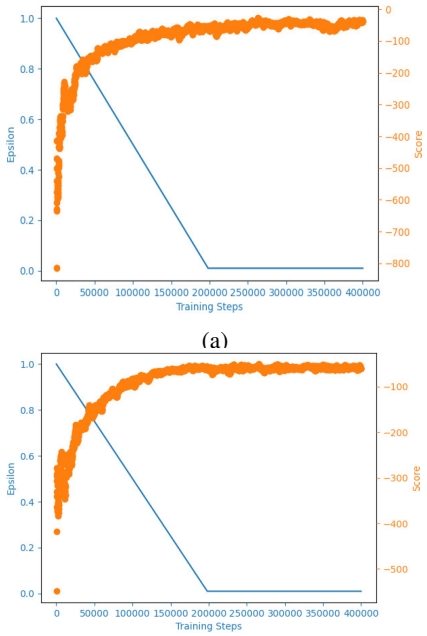
4.2 Evaluation Results

In this section, we present the results of our experiments to evaluate the improved learning efficiency of defensive agents in large-scale network environments using the proposed optimisation strategies. We first present the baseline results with DDQN in Section 4.2.1. We then investigate how the QER buffers improve the efficiency of storing and retrieving experience samples, thereby enhancing the performance of defensive agents in Section 4.2.2. With the combination of QAOA, DRL further improves the decision-making ability of defensive agents in complex network environments, discussed in Section 4.2.3.

4.2.1 DDQN Algorithm

In the initial strategy, the red agent "*Meander*" and "*B_line*" were trained independently. While this approach showed some effectiveness in their respective environments, the trained *blue agent* exhibited limitations in more complex environments, particularly a lack of robustness when dealing with a wide range of scenarios. To address these limitations, some adjustments were made to the original training strategy: (i) increasing the number of epochs to enhance the learning depth and adaptability of the model, and (ii) combining the agents "*Meander*" and "*b_line*" for hybrid training by using random functions to select the two strategies. This improved approach not only retains the strengths of their respective strategy, but also can enable more flexible responses to dynamic environments, which provides a better balance between exploration and exploitation, and an improved robustness of the model.

As a baseline model, DDQN demonstrated stable performance in standard environments. However, test results showed that DDQN performed poorly in large-scale network environments with varying trial lengths. Its singular exploration strategy restricts learning efficiency and hinders the agent's ability to adapt to evolving threats. To address this, we introduced the Boltzmann strategy (Cercignani and Cercignani, 1988) into the model, which provides the agent with more flexibility in action selection through a probability distribution. It encourages the agent to explore actions with lower Q-values that might be overlooked under the traditional ε-greedy. Experi-



(a)

(b)

Figure 3: DDQN Training with standard strategy (a) vs. with Boltzmann Strategy (b)

mental results show that after the ε-greedy strategy drops to its lowest value (min = 0.01), the Boltzmann strategy smooths the transition and avoids a drop in agent performance. In the trials of *"Meander"* and *"B-line"* as shown in Figure 3b, the scores improved compared to the standard DDQN as in Figure 3a, enhancing the overall learning efficiency.

4.2.2 DDQN with QER

We introduced a QER in this set of experiments. QER leverages the superposition and probability distribution properties of quantum states to enable more efficient selection and replay of experience samples. Experimental results (as shown in Figure 4) indicate that QER provides better performance in high-dimensional policy spaces, particularly in the B-line and Meander trials, with outcomes comparable to those achieved by the baseline DDQN model.

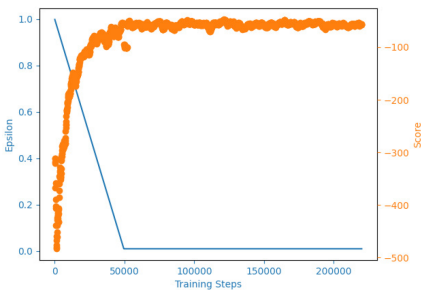
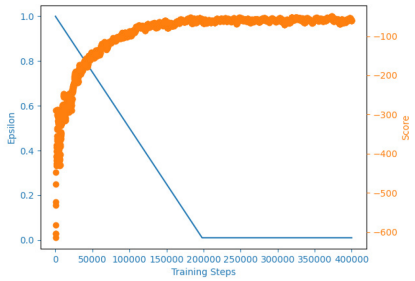


Figure 4: The training of DDQN with QER

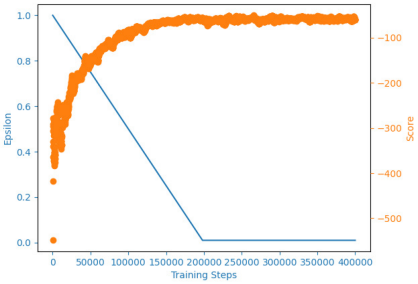
4.2.3 DDQN with QAOA

In this set of experiments, we combine QAOA with DDQN. Although the QER buffer demonstrates good performance, its complexity and high demand for computational resources make it challenging to integrate with QAOA. Due to resource constraints, we were unable to fully explore the potential of combining the QER buffer and QAOA optimization. As a result, we opted to use the priority experience replay (PER) buffer instead for this set of experiments.

The training records are shown in Figure 5a and 5b. Experimental results indicate that this combination offers better performance in both *B-line* and *Meander* trials. The integration of QAOA further enhances the system's robustness, allowing the defensive agent to make more accurate and efficient decisions when confronting complex threats.Finally, a detailed comparison of the performance of these different strategy combinations is presented in Table 1.



(a)



(b)

Figure 5: The training record of DDQN algorithm with PER (a) and with QAOA (b)

5 Conclusion and Future Work

In conclusion, this work demonstrated the potential of quantum-inspired techniques—QAOA and QER—to improve the training efficiency of defensive agents in Autonomous Cyber Defence. As cyber-attacks grow increasingly complex, the integration of these methods with DRL can enhance decision-making and

	30 steps trial length		50 steps trial length		100 steps trial length	
Blue Agent	B-line	Meander	B-line	Meander	B-line	Meander
DDQN	-14.67±6.52	-16.38±5.25	-29.50±13.34	56.97±14.50	-71.43±30.63	100.32±50.3
DDQN + Boltzmann	-12.43±5.00	-12.07±2.07	-25.00±13.12	-24.43±5.80	-65.26±30.33	-60.88±17.10
DDQN + QER	-11.32±5.93	-10.32±4.67	-20.22±14.28	-17.49±6.23	-62.95±28.68	-44.50±22.95
DDQN + QAOA	-6.91±4.38	-5.77±1.79	-12.95±6.07	-10.69±3.88	-31.19±14.09	-22.30±7.66

Table 1: Evaluation individual average rewards

responsiveness against threats, including APTs and zero-day exploits (Li and Hankin, 2017).

QER buffers represent a substantial improvement in experience sampling, leveraging quantum-inspired principles to produce more diverse and representative memory retrieval. This leads to more effective learning and enhanced defensive capabilities. Meanwhile, integrating QAOA with DRL helps solve complex optimization tasks, enabling agents to navigate intricate decision spaces and yield globally optimized solutions. This combined approach strengthens agent adaptability, producing more robust strategies for managing sophisticated cyber threats.

However, quantum-inspired methods impose computational demands and complexity. Although employing DDQN, Boltzmann strategies, and PER yielded a balance between performance and feasibility, current quantum resources are limited. Simulating quantum computing on classical hardware can introduce bottlenecks that affect scalability and realism.

Future research should focus on larger, more complex environments and real-world scenarios. Validating these techniques outside simulated settings will help identify challenges and guide practical deployments. Further exploration of QER's underlying mechanisms, dynamic parameter tuning, and efficient resource management can refine these quantum-inspired approaches. Ultimately, these methods offer promising avenues for advancing cyber defence strategies and resilience.

REFERENCES

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, O. P., and Zaremba, W. (2017a). Hindsight experience replay. In *Advances in neural information processing systems*, volume 30.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Abbeel, P., and Zaremba, W. (2017b). Hindsight experience replay. *CoRR*, abs/1707.01495.

Baillie, C., Standen, M., Schwartz, J., Docking, M., Bowman, D., and Kim, J. (2020). Cyborg: An autonomous cyber operations research gym. *arXiv preprint arXiv:2002.10667*.

Benaddi, H., Elhajji, S., Benaddi, A., Amzazi, S., Benaddi, H., and Oudani, H. (2022). Robust enhancement of intrusion detection systems using deep reinforcement learning and stochastic game. *IEEE Transactions on Vehicular Technology*, 71(10):11089–11102.

Cercignani, C. and Cercignani, C. (1988). *The Boltzmann Equation*. Springer New York.

Hasselt, H. V., Guez, A., and Silver, D. (2016). Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Kiely, M., Bowman, D., Standen, M., and Moir, C. (2023). On autonomous agents in a cyber defence environment. *arXiv preprint arXiv:2309.07388*.

Lagoudakis, M. and Parr, R. (2012). Value function approximation in zero-sum markov games. *arXiv preprint arXiv:1301.0580*.

Li, T. and Hankin, C. (2017). Effective defence against zero-day exploits using bayesian networks. In *Critical Information Infrastructures Security: 11th International Conference*, pages 123–136. Springer.

Liu, X., Zhang, H., Dong, S., and Zhang, Y. (2021). Network defense decision-making based on a stochastic game system and a deep recurrent q-network. *Computers Security*, 111:102480.

Schaul, T., Quan, J., Antonoglou, I., and Silver, D. (2015). Prioritized experience replay.

Shen, Y., Shepherd, C., Ahmed, C. M., Yu, S., and Li, T. (2024). Comparative dqn-improved algorithms for stochastic games-based automated edge intelligence-enabled iot malware spread-suppression strategies. *IEEE Internet of Things Journal*, 11(12):22550–22561.

Standen, M., Bowman, D., Son Hoang, T. R., Lucas, M., Tassel, R. V., Vu, P., Kiely, M., Konschnik, K. C. N., and Collyer, J. (2022). Cyber operations research gym. https://github.com/cage-challenge/CybORG.

Vyas, S., Hannay, J., Bolton, A., and Burnap, P. P. (2023). Automated cyber defence: A review. *arXiv preprint arXiv:2303.04926*.

Wei, Q., Ma, H., Chen, C., and Dong, D. (2021). Deep reinforcement learning with quantum-inspired experience replay. *IEEE Transactions on Cybernetics*, 52(9):9326–9338.

Zhou, L., Wang, S. T., Choi, S., Pichler, H., and Lukin, M. D. (2020). Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10(2):021067.