Efficient Quantum Soft Actor-Critic Model for Dynamic Spectrum Sharing in Intelligent O-RAN

Vu-Hai Nguyen*, Yared Abera Ergu*, Ren-Hung Hwang†, Trung Q Duong‡§, Van-Linh Nguyen*
*Dept. of Computer Science and Information Engineering, National Chung Cheng University (CCU), Taiwan
†College of Artificial Intelligence, National Yang Ming Chiao Tung University (NYCU), Taiwan
‡Faculty of Engineering and Applied Science, Memorial University, Canada
§School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, UK
g13410151@ccu.edu.tw, {yared111p, rhhwang}@cs.ccu.edu.tw, tduong@mun.ca, nvlinh@cs.ccu.edu.tw

***Abstract*—The Open Radio Access Network (O-RAN) architecture enables intelligent Dynamic Spectrum Sharing (DSS), although state-of-the-art deep reinforcement learning (DRL) methods have demonstrated remarkable practical results, their integration with quantum computing brings greater potential for enhancements in performance. In this study, we present Quantum Soft Actor-Critic (QSAC), a novel hybrid quantum-classical methodology aimed at tackling barren plateaus and demonstrating enhanced performance compared to conventional DRL benchmarks, including Soft Actor-Critic (SAC) and Twin Delayed Deep Deterministic Policy Gradient (TD3), achieving a higher demand satisfaction ratio and markedly improved ϵ-band compliance rates under demanding conditions. Additionally, we present a comprehensive architectural design for the effective integration of our framework into the O-RAN non-real-time and near-real-time RICs. Our quantum-enhanced actor significantly reduces the number of trainable actor parameters by over 99% compared to its conventional SAC equivalent (from around 73,000 to merely 377), indicating an important improvement in model efficiency. To promote repeatability and facilitate future research, we offer the open-source implementation of our study.**

***Index Terms*—Quantum machine learning, Quantum Soft Actor-Critic, Dynamic spectrum sharing, O-RAN, reinforcement learning**

I. Introduction

The rapid progress of wireless communications has resulted in a rising demand for spectrum utilization, leading in an increasing shortage of radio spectrum resources. To address this issue, two main approaches have emerged: (1) extending into higher frequency bands (which often involves high auction and deployment costs) or (2) reusing and utilizing the existing granted spectrum more efficiently. Due to the cost, the second approach is more reasonable for temporary [1]. Although 4G Long Term Evolution (LTE) remains widely utilized, we may assume that NR devices will soon become widely used as LTE devices nowadays. Cognitive radio (CR) has been recognized as an important solution to spectrum shortage due to its capacity to sense, learn, and adapt dynamically to the communication environment [2].

Recently, the Open Radio Access Network (O-RAN) [3] has developed as an open architecture for the integration of artificial intelligence (AI) and machine learning (ML) applications for effective network resource management. This convergence presents a promising opportunity to integrate CR technology with end devices and address the spectrum shortage issue . Dynamic spectrum sharing (DSS) has emerged as an essential technique for enhancing spectrum effectiveness, based on the concepts of CR. DSS facilitates more efficient and collaborative spectrum utilization, providing a pragmatic and adaptable solution to these issues. However, effective implementation of DSS demands an adaptable and intelligent network architecture, presenting a distinct difficulty in opti-

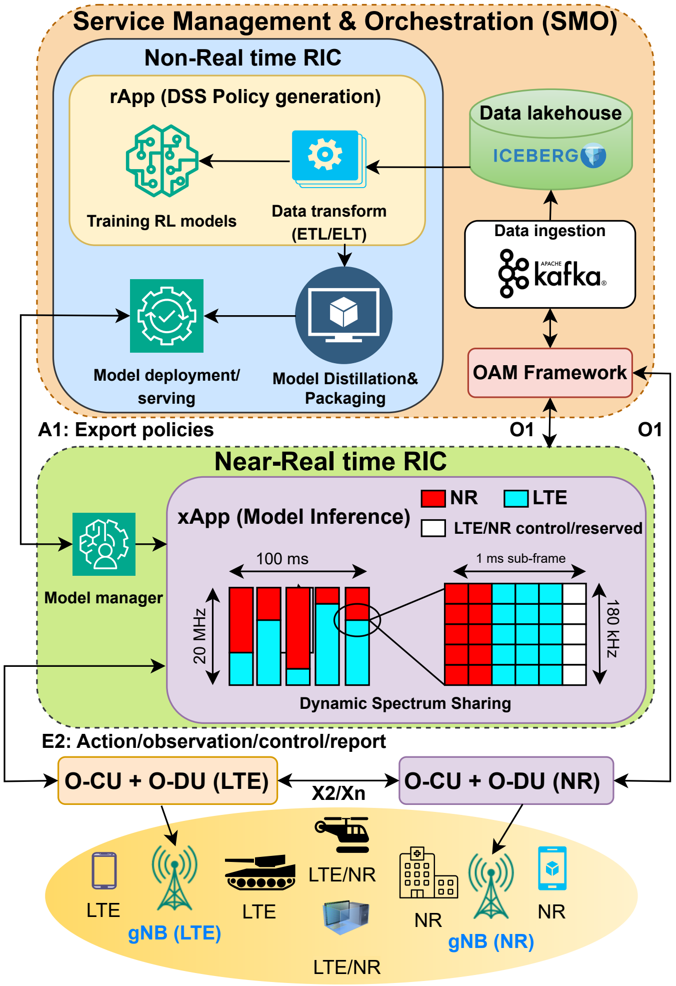


Fig. 1: Conceptual diagram of Intelligent O-RAN for Dynamic Spectrum Sharing between LTE and NR

mizing resource allocation within dynamic and complicated environments-thereby heightening the demand for a more robust computational model.

A. Related Works

Dynamic Spectrum Sharing (DSS) is emerging as a key enabler for beyond-5G and O-RAN networks, allowing heterogeneous radio technologies (e.g., LTE, NR, and Wi-Fi) to dynamically coexist within shared spectrum. In such environments, the radio landscape is highly non-stationary and multi-operator, demanding autonomous coordination under near-real-time (near-RT) constraints. Traditional optimization and game-theoretic schemes, though effective in static scenarios, are inadequate for this dynamic setting as they rely on fixed models and centralized control, which cannot adapt to time-varying interference and distributed decision loops.

Recent surveys highlight that data-driven and AI/ML-based DSS frameworks are increasingly essential for adaptive spectrum management and real-time decision making in open and intelligent RAN architectures [4]. Existing research, summarized in **Table I**, can be broadly categorized into:

a) Spectrum perception and classification via deep learning: These approaches focus on identifying spectrum usage patterns and interference conditions using DNN-based classifiers. For example, *ChARM* [5] supports LTE–Wi-Fi coexistence in unlicensed bands with low-latency spectrum identification. However, such methods are primarily perception-oriented and lack autonomous decision-making capabilities for spectrum allocation or policy adaptation.

b) Spectrum allocation and policy optimization via reinforcement learning: This category includes both model-free DRL and hybrid RL frameworks aimed at dynamic spectrum access and sharing. *AdapShare* [6] and actorcritic models [7] utilize TD3/DDPG for LTE–NR coordination, showing promise in policy convergence but facing scalability and exploration challenges. Hybrid methods such as DRL+MCTS [8] and DEQN [9] enhance temporal reasoning and stability but lack near-RT scalability and full O-RAN integration.

Across both categories, existing solutions rely on classical neural approximators and struggle to balance real-time adaptation with distributed coordination. The application of ML, particularly DRL, has shown enhanced efficacy relative to conventional methods, especially in adapting to real-world dynamics such as fluctuating traffic, noise, and incomplete information [7].

To the best of our knowledge, this study is the first to introduce Quantum Reinforcement Learning (QRL) for DSS in O-RAN. The proposed framework leverages Variational Quantum Circuits (VQCs) chained with DRL to enhance policy expressiveness, exploration efficiency, and robustness under stochastic traffic and interference. Quantum Machine Learning (QML) opens up revolutionary opportunities for enhancing AI with unforeseen capabilities [10], especially in designing next-generation network architectures. In this context, we present an efficient QSAC method to address the DSS problem, demonstrating superior performance and integration potential within the O-RAN architecture, as illustrated in **Fig. 1**. Further operational details and practical significance are discussed in Section III-A.

TABLE I: Summary–DSS studies in O-RAN/Wireless NWs

Study	Method	Objective	O -RAN	RT	Scal.	Limitations
[6]	DRL (TD3, DDPG)	LTE–NR alloc.	✓	✓	✗	Low exploration effi.
[5]	DNN classifier	LTE–WiFi coexist.	✓	✓	✗	Classifi. only
[11]	RL + AI/ML	Spectrum optim.	✓	✓	✗	No DSS-speci-eval.
[9]	DEQN (RNN+DRL)	Adaptive DSS	✗	✓	✗	Limited scalability
[7]	Actor–Critic RL	Access + sharing	✗	✓	✓	Action-space-compl.
[8]	DRL + MCTS	4G–5G alloc.	✗	✗	✗	limited RT adapt.
Ours	QML+DRL	DSS in O -RAN	✓	✓	✓	RT, robust, scalable

B. Contributions

Compared to AdapShare [6], which formulates DSS as a contextual bandit problem (where the agent makes decisions based on current context without considering long-term consequences or state transitions) and uses classical actor-critic RL with synthetic data, our approach models the task as a full Markov Decision Process (MDP) and applies quantum reinforcement learning (QRL) for improved convergence, scalability, and continuous control. These improvements in results show enhanced resource efficiency and fairness, highlighting QRL's potential for future integration into O-RAN-based DSS.

This work presents the following key contributions

- We introduce QSAC, the first quantum-enhanced reinforcement learning model tailored for dynamic spectrum sharing (DSS) in O-RAN. Our custom Q-SAC architecture is benchmarked against classical RL models, including AdapShare, to provide a comparative evaluation.
- Experimental results demonstrate that QSAC achieves a spectrum efficiency of 0.761 and a DSR of 0.989 for NR and 0.987 for LTE, outperforming TD3, DDPG, and SAC under constrained resource conditions. QSAC also achieves a 91% LTE-5% ϵ-band compliance rate, significantly higher than TD3 (12%), DDPG (29%), and SAC (53%).
- QSAC exhibits a clear quantum advantage by achieving faster convergence and more stable learning dynamics across varying demand profiles. Its reward trajectory shows reduced variance and consistent improvement, indicating robust policy learning in highly dynamic environments. To support reproducibility and future research, we release the full source code at Github repository.

II. System Model and Problem Formulation

We extend the AdapShare formulation [6] to an intelligent O-RAN architecture for dynamic spectrum sharing between LTE and NR, as illustrated in Fig. 1. Both networks use a compatible numerology with 15 kHz subcarrier spacing, enabling shared scheduling over physical resource blocks (PRBs). Our system leverages the O-RAN framework, where the Non-Real-Time RIC trains reinforcement learning models and the Near-Real-Time RIC performs inference using a Quantum Soft Actor-Critic (SAC) algorithm. The agent learns traffic patterns, channel states, and interference dynamics from time-series data to optimize PRB allocation in real time, minimizing surplus and deficit to improve spectrum efficiency and reduce packet loss.

A. Efficient resource sharing

The resource pool of PRBs is shared when LTE and NR networks operate within the same frequency band. The premise is that NR functions partially within the existing LTE frequency, with demand fluctuations occurring at consistent time intervals. These intervals represent time frames in the LTE system: 1 ms subframes, or durations of several seconds, minutes, or hours depending on the practical environment. At each time step t, each network requires a specific quantity of PRBs: we denote the LTE demand as $D_{\mathbb{L},t}$ and the NR demand as $D_{\mathbb{N},t}$. The main aim of efficient resource sharing is to mitigate both surplus and deficit of the two networks through the goal function J_{ers}. The optimization problem is defined [6] as follows (1)

$$\begin{aligned} \textbf{Optimization:} \quad & \min_{A_{\mathbb{L},t}, A_{\mathbb{N},t}} J_{ers}(A_{\mathbb{L},t}, A_{\mathbb{N},t}, D_{\mathbb{L},t}, D_{\mathbb{N},t}), \\ \textbf{s.t.} \quad & A_{\mathbb{L},t} + A_{\mathbb{N},t} \leq N_{r,t}, \\ & 0 \leq A_{\mathbb{L},t} \leq N_{r,t}, \\ & 0 \leq A_{\mathbb{N},t} \leq N_{r,t}, \end{aligned} \tag{1}$$

where

$$J_{ers} = \zeta(\frac{A_{\mathbb{L},t} - D_{\mathbb{L},t}}{D_{\mathbb{L},t}})^2 + (\zeta - 1)(\frac{A_{\mathbb{N},t} - D_{\mathbb{N},t}}{D_{\mathbb{N},t}})^2 \tag{2}$$

$N_{r,t}$ is defined as the total number of resources available in the pool for distribution to both LTE and NR, letting $(A_{\mathbb{L},t}, A_{\mathbb{N},t})$ represent the allocation pair for LTE and NR at time t. Therefore, $\frac{N_{\mathbb{L},t} - D_{\mathbb{L},t}}{D_{\mathbb{L},t}}$ and $\frac{N_{\mathbb{N},t} - D_{\mathbb{N},t}}{D_{\mathbb{N},t}}$ denote the surplus and deficit components, respectively. The weighting factor $\zeta \in [0, 1]$ facilitates intent-driven spectrum management by allowing the controller to prioritize one network above another: $\zeta \rightarrow 1$ favors LTE, while $\zeta \rightarrow 0$ favors NR. Consequently, the optimization problem 1 can be restructured as an RL strategy based on this formulation.

B. Soft Actor-Critic (SAC) Problem Formulation

This subsection presents the RL approach and provides a detailed justification of using the SAC algorithm for solving the DRL-based DSS [12]. A key advantage of SAC is its capacity to motivate the actor to pursue substantial rewards (as assessed by the critic) while simultaneously demonstrating diversity in its actions, hence fostering exploration alongside the maintenance of exploitation balance, serves as a foundational principle for proposing QSAC. The details are stated as follows:

State and Observation: The state space s_t consists of the demands of LTE and NR at each time step t, where $s_t = \{D_{\mathbb{L},t},\ D_{\mathbb{N},t}\}$. Furthermore, at time t, the network has an observation comprising the past information of prior demands, which is defined as $o_t = \big[(D_{\mathbb{L},\kappa}, D_{\mathbb{N},\kappa}) \bigm| \kappa = t, t-1, \ldots, t-n\big]$.

Action: At each time step t, the action (μ_t, υ_t) is a continuous vector that defines the proportion of total resources $N_{r,t}$ to be allocated. The action space A is normalized and defined by: $a_t = (\mu_t, \upsilon_t)$ s.t. $\mu_t, \upsilon_t \in [0, 1]$ and $\mu_t, \upsilon_t \leq 1$ The optimal allocation of PRBs is $(A_{\mathbb{L},t}, A_{\mathbb{L},t}) = (\mu_t Nr, t, \upsilon_t Nr, t)$, which must comply with the limitations outlined in Eq. (1). Normalisation offers a scalable framework and enhances training stability.

Reward: The aim of the reward function is to identify the pair $A_{LTE,t}, A_{NR,t}$ that minimizes J or maximizes -J. we denote D_t total demand and A_t total allocation at time t and the reward function is defined as

$$r_t = \begin{cases} -J - 10J, & \text{if } A_t \text{ and } D_t - \gtrless N_{r,t}, \\ -5J, & \text{if } D_t > N_{r,t} \text{ and } \big(A_{\mathbb{L},t} = 0 \text{or} A_{\mathbb{N},t} = 0\big), \\ -J + 10, & \text{if } D_t > N_t \text{ and } A_t > N_{r,t} - 0.03, \\ -J + 5, & \text{if } D_t > N_t \text{ (all other cases)}, \\ -J, & \text{if } D_t \leq N_t \text{ and } A_t \leq N_{r,t}. \end{cases} \tag{3}$$

To solve the formulated optimization problem, we employ the Soft Actor-Critic (SAC) algorithm [12], a state-of-the-art off-policy actor-critic method. The agent acquires a stochastic policy π_ϕ (actor) and two Q-functions $Q_{\theta_1}, Q_{\theta_2}$ (critics) by optimising their corresponding objective functions through gradient-based techniques, as outlined in [12].

C. Variational Quantum Actor for SAC

This subsection presents the proposed QSAC architecture, based on the concepts of the SAC algorithm stated in subsection II-B, we also provide the comprehensive processing algorithm of the proposed QSAC in Algorithm 1. The QSAC model consists of two primary components that reflect the actor-critic architecture of the SAC: an actor network and a critic network. Specifically, we propose a variational quantum actor π_θ for the actor network, which employs a hybrid quantum-classical policy network. In contrast, the critic network is implemented using a classical value network consisting of Q_{ϕ_1}, Q_{ϕ_2}. When implementing QSAC, we strictly follow the SAC algorithm. However, line 7 applies a delayed policy update condition. This approach enables the Q-value, as evaluated by the critics, to stabilize prior to being utilized in policy updates, thus avoiding the actor from learning from temporal errors in the critics and fostering a more stable training process overall.

In the actor network, the hybrid quantum-classical policy network consists of three main components, as illustrated in **Fig. 2**. The data processing flow proceeds from 1. Pre-processing layer to 2. Data Re-uploading and Variational Quantum Circuit, and finally to 3. Post-processing layer.

Initially, at the pre-processing layer, the input is the state $s \in \mathbb{R}^S$ with an input qubit count of N_q. A linear projection layer is employed to align the input dimension with the number of qubits, with the objective of preserving and transferring maximal information into the space specified by the available N_q qubits: $z = Ws + b \in \mathbb{R}^{N_q}$. Following that, the data undergoes a layer normalization process with $\tilde{z} = LN(z)$, ensuring that the mean $\mathbb{E}[\tilde{z}] = 0$ and the standard deviation $\text{Var}[\tilde{z}] = 1$. Upon normalization, the data is subjected to a data

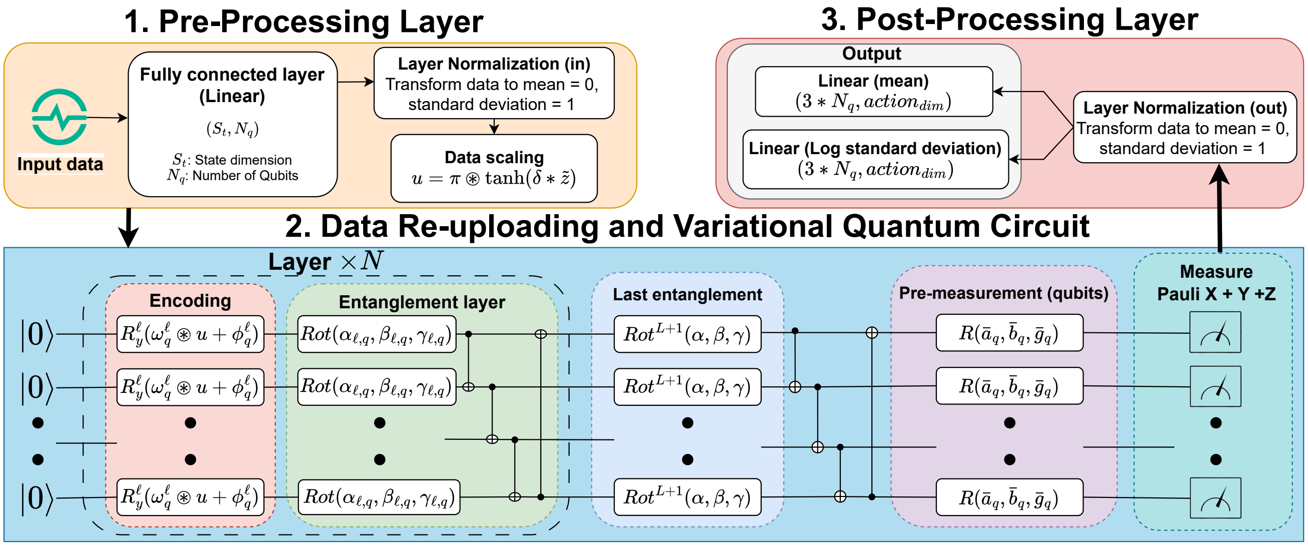

1. Pre-Processing Layer
Input data
Fully connected layer (Linear)
(S_t, N_q)
S_t: State dimension
N_q: Number of Qubits
Layer Normalization (in)
Transform data to mean = 0, standard deviation = 1
Data scaling
$u = \pi \circledast \tanh(\delta * \tilde{z})$
3. Post-Processing Layer
Output
Linear (mean)
$(3 * N_q, action_{dim})$
Linear (Log standard deviation)
$(3 * N_q, action_{dim})$
Layer Normalization (out)
Transform data to mean = 0, standard deviation = 1
2. Data Re-uploading and Variational Quantum Circuit
Layer $\times N$
Encoding
Entanglement layer
Last entanglement
Pre-measurement (qubits)
Measure
Pauli X + Y +Z
$|0\rangle$
$R_y^\ell(\omega_q^\ell \circledast u + \phi_q^\ell)$
$Rot(\alpha_{\ell,q}, \beta_{\ell,q}, \gamma_{\ell,q})$
$Rot^{L+1}(\alpha, \beta, \gamma)$
$R(\bar{a}_q, \bar{b}_q, \bar{g}_q)$

Fig. 2: The Quantum Actor Architecture for the QSAC Agent in Intelligent O-RAN

scaling process that maps it into rotation angles appropriate for quantum data embedding, employing the following equation:

$$u = \pi\ \tanh(\delta \odot \tilde{z}) \quad \text{with} \quad \delta \in \mathbb{R}^{N_q} \tag{4}$$

The goal is to use $u_i \in (-\pi, \pi)$ as the encoding parameter for each qubit, preparing the data for quantum encoding. Secondly, in the re-uploading Variational Quantum Circuit, we propose the data re-uploading technique based on the principles established by Adrin et al. [13]. The data re-uploading block comprises N layers, integrated with a variational quantum circuit, and involves various supplementary sub-blocks prior to advancing to the subsequent level. We initialize all N_q as $|0\rangle$. In each encoding layer ℓ, each qubit q is encoded via the Y-gate as equation 5 follows

$$U_{\text{enc}}^{(l)}(\omega_q^\ell, u, \phi_q^\ell) = \bigotimes_{q=1}^{N} R_Y^\ell(\omega_q^\ell u + \phi_q^\ell) \tag{5}$$

with $R_Y(\varphi) = exp(-i\frac{\varphi}{2}Y)$ and ω_i, ϕ_i are learnable parameters, deep nonlinear modeling is achieved by re-uploading the input u at every layer. Following this, the Entanglement layer in the data reuploading block, located downstream of the encoding block, utilises entangling processes, mathematically expressed by the equation 6

$$U_{\text{ent}}^{(\ell)}(\Theta_q^{(\ell)}) = \left(\prod_{q=1}^{n} \text{Rot}(\Theta_q^{(\ell)})_{(q)}\right) \cdot \left(\prod_{q\in\mathcal{E}^{(\ell)}} CNOT_q\right) \tag{6}$$

with $\Theta_n^{(\ell)} = \{\alpha_{\ell,q}, \beta_{\ell,q}, \gamma_{\ell,q}\} \in \mathbb{R}^{n\times 3}$, they are also learnable parameters. $\text{Rot}(\Theta_q^{(\ell)})$ is a set of quantum gates consisting of the sequence $R_y(\alpha_{\ell,q}), R_z(\beta_{\ell,q}), R_y(\gamma_{\ell,q})$, with $Ry(\varphi)$ is mentioned above and $R_Z(\varphi) = exp(-i\frac{\varphi}{2}Z)$. Subsequent to the rotation gates, an entanglement block, denoted by the operator $\prod_{q\in\mathcal{E}^{(\ell)}} \text{CNOT}_q$ is executed. This operator comprises a series of CNOT gates, each functioning on a control qubit c and a target qubit t as per the mapping $|c,t\rangle \longmapsto |c, t \oplus c\rangle$ where $\oplus$ denotes addition modulo 2. The qubit pairings for these operations are predetermined in the set $E^{(\ell)}$ for each layer, thereby establishing entanglement across the quantum register. After completing the Data re-uploading block, an extra entanglement block is incorporated to enhance the expressive capacity of the data: $U_{\text{ent}}^{(L+1)}(\Theta_q^{(L+1)})$. Prior measurement, we implement an additional quantum gate called pre-measurement rotations, with the critical objective of optimizing the measurement process. The rotation gate is defined by a set of learnable parameters represented by $\bar{\Theta} = \{\bar{a}_q, \bar{b}_q, \bar{g}_q\}$, which we initialize and sample from a uniform distribution over the interval $[\frac{\pi}{2} - 0, 1, \frac{\pi}{2} + 0, 1]$. This initialization pushes the qubits to start rotating in a nearly equal superposition state, aligning with the area of maximum gradient magnitude.

$$U_{\text{pre}}(\bar{\Theta}) = \bigotimes_{q=1}^{n} \text{rot}(\bar{a}_q, \bar{b}_q, \bar{g}_q)_{(q)} \tag{7}$$

The resulting final state is therefore given by $|\psi_{\text{out}}(\mathbf{x})\rangle = U_{\text{pre}}(\bar{\Theta})\, U_{\text{ent}}^{(L+1)}(\Theta^{(L+1)}) \left[\prod_{\ell=1}^{L} U_{\text{ent}}^{(\ell)}(\Theta^{(\ell)})\, U_{\text{enc}}^{(\ell)}(\mathbf{x})\right] |\psi_0\rangle$. Finally, after the application of the pre-measurement rotation gates, the quantum data is measured and projected into a classical state, prepared for processing in the subsequent layer. The result of the quantum circuit is a classical vector $\mathbf{y} \in \mathbb{R}^{3n}$, derived by calculating the expectation values of Pauli operators $P \in \{X, Y, Z\}$ concerning the quantum state $|\psi_{\text{out}}(\mathbf{x})\rangle$. The output vector $\mathbf{y}$ can be defined as follows: $y = [\langle X_1\rangle, \ldots, \langle X_n\rangle, \langle Y_1\rangle, \ldots, \langle Y_n\rangle, \langle Z_1\rangle, \ldots, \langle Z_n\rangle]^T$ and each component of the output vector is measured according to the Born rule: $\langle P_j\rangle = \langle\psi_{\text{out}}(\mathbf{x})|P_j|\psi_{\text{out}}(\mathbf{x})\rangle$.

In the third phase, the post–processing layer, we implement a normalization layer similar to that utilized in the pre-processing stage. Furthermore, we incorporate two linear projector blocks to generate the mean μ and standard deviation σ, which act as inputs for the following critic network.

III. Results and Discussions

We used a public dataset [6] and the following evaluation measures to quantitatively evaluate the problem objectives. We

Algorithm 1: QSAC Training

```
Input: Variational quantum actor $\pi_\theta$, classical critics
       $Q_{\phi_1}, Q_{\phi_2}$, replay buffer $\mathcal{D}$
Initialize: Parameters $\theta, \phi_1, \phi_2$; Target networks
       $\phi_{\text{targ},i} \leftarrow \phi_i$; Temperature $\alpha$
for each training step do
    Observe state $s_t$ and select action $a_t \sim \pi_\theta(\cdot|s_t)$
      Execute $a_t$, observe reward $r_t$, next state $s_{t+1}$,
      and done signal $d$;
    Store transition $(s_t, a_t, r_t, s_{t+1}, d)$ in replay buffer
      $\mathcal{D}$;
    Sample a minibatch $\mathcal{B} = \{(s, a, r, s', d)\}$ from $\mathcal{D}$;
    Compute target values: $y \leftarrow r + \gamma(1 -
      d)(\min_{i=1,2} Q_{\phi_{\text{targ},i}}(s', a') - \alpha \log \pi_\theta(a'|s'))$,
      where $a' \sim \pi_\theta(\cdot|s')$;
    Update critic parameters $\phi_i$ for $i = 1, 2$ by
      minimizing the loss:
      $\mathcal{L}_\phi = \mathbb{E}_{(s,a,y)\sim\mathcal{B}}[\sum_{i=1,2}(Q_{\phi_i}(s, a) - y)^2]$;
    if delayed policy update then
        Update actor parameters $\theta$ by maximizing the
          objective: $\mathcal{J}_\pi(\theta) =$
          $\mathbb{E}_{s\sim\mathcal{B},a\sim\pi_\theta}[\min_{i=1,2} Q_{\phi_i}(s, a) - \alpha \log \pi_\theta(a|s)]$;
        Update temperature $\alpha$ by minimizing the
          objective (with target entropy $\mathcal{H}$):
          $\mathcal{J}(\alpha) = \mathbb{E}_{s\sim\mathcal{B},a\sim\pi_\theta}[-\alpha(\log \pi_\theta(a|s) + \mathcal{H})]$;
        Update target critic parameters for $i = 1, 2$:
          $\phi_{\text{targ},i} \leftarrow \tau\phi_i + (1 - \tau)\phi_{\text{targ},i}$;
    end
end
```

TABLE II: Hyper-parameters used in the experiments

Parameter	DDPG	TD3	SAC	QSAC
Exploration std σ_{act}	0-0.09	0-0.09	–	–
Policy-smoothing std σ_{targ}	0.1	0.1	–	–
Smoothing clip C	0.2	0.2	–	–
Entropy coef α	–	–	1e−4	1e−4
Actor architecture	MLP	MLP	MLP	Q-actor
Critic architecture	MLP	MLP	MLP	MLP
Obs norm	Yes	Yes	Yes	Yes
Actor Parameters	72,962	72,962	73,476	**377**
Critic Parameters	73,217	146,434	146,434	146,434

present three main metrics as follows:

1) The Demand-satisfaction ratio (DSR) measures how well the allocated resources meet the demand at each step t. For LTE and NR as $\Lambda = \{\mathbb{L}, \mathbb{N}\}$ as in subsection II-A and the total number of data samples is T and ϱ is a small constant to avoid division by zero, it is defined as

$$\text{DSR}_{A,\text{mean}} = \frac{1}{T}\sum_{t=1}^{T} \frac{\min(A_{\Lambda,t}, D_{\Lambda,t})}{\max(D_{\Lambda,t}, \varrho)} \tag{8}$$

(2) This metric quantifies the efficiency of the overall PRB usage

$$U_{\text{mean}} = \frac{1}{T}\sum_{t=1}^{T} \frac{A_{\Lambda,t} + A_{\Lambda,t}}{N_{r,t}}. \tag{9}$$

3) ϵ-band compliance rate ($\epsilon -$BCR) measures the proportion of time steps where the allocation is sufficiently close to the demand. For a tolerance factor $\varepsilon \in \{\ 5\%, 10\%, 15\%\}$, the condition is

$$\epsilon - \text{BCR}_{\Lambda,\varepsilon} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{1}(|A_{\Lambda,t} - D_{\Lambda,t}| \leq \varepsilon \cdot D_{A,t})\,, \tag{10}$$

where $\mathbf{1}(\cdot)$ is the indicator function. We employ the implementation and hyperparameters from Adapshare [6] for the TD3 and DDPG models, which act as benchmarks for comparison with our proposed SAC and QSAC models. We use the following shared hyperparameters across all models: Actor learning rate = 1e-4, Critic learning rate = 2e-3, batch size = 48, time step = 1, target update rate τ = 1e-4, policy update frequency = 2, and replay buffer size = 1e6. The rest of hyper-parameters settings are in the Table II.

A. Results and Discussion

We used the total number of PRB resources: $N_{r,t} = 60$ and total samples of 860 for each LTE and NR networks for training procedures and performed simulations throughout 100 timesteps of the dataset and achieved some significant positive outcomes. Table III illustrates that QSAC achieved the highest average DSR for NR, although the LTE DSR of 0.987 was also extremely competitivemarginally following TD3 and markedly surpassing DDPG and SAC. While TD3 achieved the highest DSR for LTE, it incurred a comparatively substantial excess comparing to other models, indicating that TD3 often favours short-term benefits above global optimality. Significantly, under the stringent LTE-5% epsilon band compliance rate, QSAC attained an overall result of 0.91, while SAC, DDPG, and TD3 achieved rates of 0.53, 0.29, and 0.12, respectivelya considerable gap. At LTE-10%, QSAC consistently outperformed other models, achieving a result of 0.98. **Fig. 3** illustrates the reward curve in training, calculated with a 100-step moving

TABLE III: Quantitative Performance Comparison of RL models

Model	DSR (mean)		Util	$\epsilon -$ band compliance rate					
	LTE	NR	mean	LTE-5%	NR-5%	LTE-10%	NR-10%	LTE-15%	NR-15%
QSAC	0.987	**0.989**	0.761	**0.91**	0.71	**0.98**	0.86	0.99	0.94
SAC	0.948	0.956	0.718	0.53	0.55	0.80	1.00	0.99	1.00
TD3 [6]	**0.999**	0.966	0.770	0.12	0.74	0.94	1.00	0.99	1.00
DDPG [6]	0.977	0.980	0.767	0.29	**0.92**	0.48	1.00	0.92	1.00

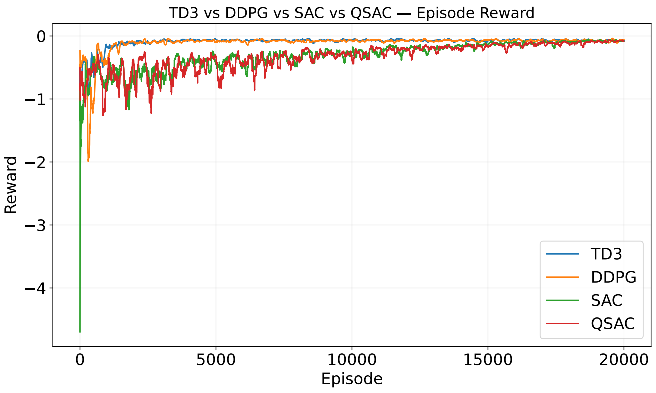
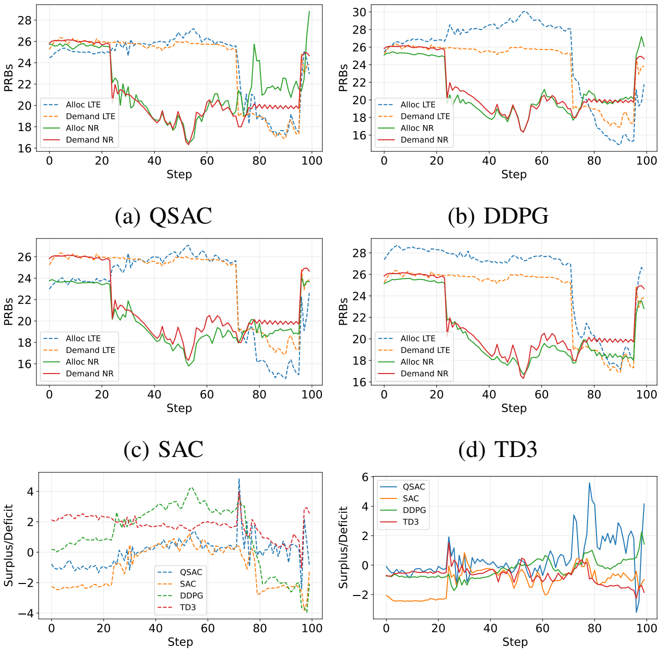


Fig. 3: Episode reward comparison of the RL models



(a) QSAC

(b) DDPG

(c) SAC

(d) TD3

(e) LTE Surplus/Deficit

(f) NR Surplus/Deficit

Fig. 4: Analysis of the dynamic resource allocation performance: Allocation vs Demand

average. During the early phase, SAC and QSAC demonstrate lower and variable rewards owing to their stochastic policies that prioritize entropy maximizationthis reflects a calculated trade-off to promote exploration of the action space. In the intermediate phase, both QSAC and SAC exhibit consistent reward development with substantially reduced variance. In contrast, TD3 and DDPG exhibit a more rapid initial growth, followed by plateau phases and oscillations around local optima, indicative of their reliance on policies that emphasize short-term optimization with elevated risk. In the concluding phase, QSAC meticulously monitors a consistently elevated reward level, exhibiting minimal profound negative outliers. SAC continues to exhibit minor noise, but TD3/DDPG maintain their prior values with gradual enhancement.

Fig. 4a–Fig. 4d illustrate all model performance, while Figs. 4e and 4f show the corresponding surplus/deficit patterns (near 0 is better). QSAC Fig. 4a maintains LTE allocations with minimal variance and minor deviations following load changes around steps 30 and 75, quickly re-stabilizing within a few steps; SAC Fig. 4c exhibits long fluctuations, while DDPG Fig. 4b and TD3 Fig.4d remain in surplus or deficit areas, when the allocation deviates by about 24 PRBs from the actual demands, conventional models reveal larger negative spikes and persistent bias in Fig. 4e. In NR, although QSAC has not yet maintained stability in the final timesteps, it exhibits the most stable trajectory initially and demonstrates the quickest recovery after shocks, with the surplus/deficit traces in Fig. 4f centered around zero and showing the least spread and extremes. QSAC achieves the minimal allocation variance, the shortest recovery time step, and the lowest worst-case deficit/surplus, demonstrating its superiority for more stable PRB utilization and better demand satisfaction consistency compared to DDPG, SAC, and TD3.

IV. Conclusion

This paper introduced QSAC, a quantum-enhanced reinforcement learning model for dynamic spectrum sharing (DSS) in O-RAN. Through simulation over 100 time-steps and training on 860 samples per LTE and NR network, QSAC achieved strong performance, including a DSR of 0.989 for NR and 0.987 for LTE. It also demonstrated superior compliance under tight constraints, achieving 91% LTE-5% ϵ-band compliance-significantly outperforming TD3, DDPG, and SAC. Compared to conventional RL models, QSAC maintained a stable reward trajectory with reduced variance, indicating robust policy learning and long-term optimization. These results validate a promise of integrating variational quantum circuits into DRL for spectrum coordination. Future work will explore scaling QSAC to multi-agent settings, optimizing quantum circuit depth for real-time inference, and extending the framework to other intelligent RAN functions such as traffic steering and energy-efficient scheduling.

References

[1] A. Damnjanovic, D. Knisley, A. Saurabh, R. Prakash, X. Zhang, and S. Chen, "Spectrum sharing with o-ran architecture," in *2024 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2024, pp. 108–113.

[2] F. Hu, B. Chen, and K. Zhu, "Full spectrum sharing in cognitive radio networks toward 5g: A survey," *IEEE Access*, vol. 6, pp. 15 754–15 776, 2018.

[3] M. Polese, L. Bonati, S. DOro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[4] F. R. V. Guimares, J. M. B. da Silva Jr., C. Casimiro Cavalcante, G. Fodor, M. Bengtsson, and C. Fischione, "Machine learning for spectrum sharing: A survey," *Foundations and Trends in Networking*, vol. 14, no. 12, p. 1159, 2024. [Online]. Available: http://dx.doi.org/10.1561/1300000073

[5] L. Baldesi, F. Restuccia, and T. Melodia, "Charm: Nextg spectrum sharing through data-driven real-time o-ran dynamic control," in *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, 2022, pp. 240–249.

[6] S. Gopal, D. Griffith, R. A. Rouil, and C. Liu, "Adapshare: An rl-based dynamic spectrum sharing solution for o-ran," in *2025 IEEE 22nd Consumer Commun. & Networking Conf. (CCNC)*, 2025, pp. 1–7.

[7] L. Dong, Y. Qian, and Y. Xing, "Dynamic spectrum access and sharing through actor-critic deep reinforcement learning," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, no. 1, Jun. 2022. [Online]. Available: http://dx.doi.org/10.1186/s13638-022-02124-4

[8] U. Challita and D. Sandberg, "Deep reinforcement learning for dynamic spectrum sharing of lte and nr," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[9] H.-H. Chang, L. Liu, and Y. Yi, "Deep echo state q-network (deqn) and its application in dynamic spectrum sharing for 5g and beyond," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 929–939, 2022.

[10] V.-L. Nguyen, L.-H. Nguyen, R.-H. Hwang, B. Canberk, and T. Q. Duong, "Quantum machine learning for 6g network intelligence and adversarial threats," *IEEE Communications Standards Magazine*, vol. 9, no. 3, pp. 40–48, 2025.

[11] R. Barker, "From deepsense to open ran: Ai/ml advancements in dynamic spectrum sensing and their applications," 2025. [Online]. Available: https://arxiv.org/abs/2502.02889

[12] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018. [Online]. Available: https://arxiv.org/abs/1801.01290

[13] A. Prez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, Feb. 2020. [Online]. Available: http://dx.doi.org/10.22331/q-2020-02-06-226