Generative Resource Allocation for 6G O-RAN with Diffusion Policies

Salar Nouri*, Mojdeh Karbalaee Motalleb*, Vahid Shah-Mansouri*, and Tarik Taleb†
*School of Electrical and Computer Engineering, University of Tehran, Tehran, Iran
Email: {salar.nouri, mojdeh.karbalee, vmansouri}@ut.ac.ir
†Ruhr University Bochum, Bochum, Germany
Email: tarik.taleb@rub.de

***Abstract*—Dynamic resource allocation in O-RAN is critical for managing the conflicting QoS requirements of 6G network slices. Conventional reinforcement learning agents often fail in this domain because their unimodal policy structures cannot model the multi-modal nature of optimal allocation strategies. This paper introduces Diffusion Q-Learning (Diffusion-QL), a novel framework that represents the policy as a conditional diffusion model. Our approach generates resource allocation actions by iteratively reversing a noising process, with each step guided by the gradient of a learned Q-function. This method enables the policy to learn and sample from the complex distribution of near-optimal actions. Simulations demonstrate that the Diffusion-QL approach consistently outperforms state-of-the-art DRL baselines, offering a robust solution for the intricate resource management challenges in next-generation wireless networks.**

***Index Terms*—Resource allocation, Network slicing, Reinforcement Learning, Diffusion Model, Generative AI**

I. Introduction

THE vision for sixth generation (6G) wireless networks, enabled by flexible architectures like Open Radio Access Network (O-RAN), promises transformative applications such as holographic communications and the tactile internet [1], [2]. Achieving this vision requires addressing a critical resource management challenge: supporting network slices with conflicting quality of service (QoS) requirements. Enhanced Mobile Broadband (eMBB) demands peak rates above 10 Gbps, Ultra-Reliable Low Latency Communications (URLLC) requires sub-millisecond latency with near-perfect reliability, and massive massive Machine-Type Communications (mMTC) must serve extremely high device densities [3], [4]. This creates a complex, high-dimensional, dynamic optimization problem beyond traditional allocation methods, motivating intelligent network control.

The deep reinforcement learning (DRL) emerged as a promising paradigm for this challenge, with recent efforts exploring various architectures. Advanced methods leveraged graph neural networks to capture topological complexities in V2X systems [5], multi-agent frameworks for distributed control [6], [7], and federated learning to enhance privacy and scalability [8]. However, a critical research gap persists: these approaches commonly rely on simple, unimodal policy parameterizations, such as Gaussian distributions. Such policies are ill-equipped to capture the complex, often multi-modal, nature of the optimal resource allocation solution space, where multiple distinct allocation strategies can yield similar performance [9]. This fundamental limitation in policy expressiveness leads to suboptimal performance and poor generalization in the dynamic O-RAN environment [10].

Recognizing this fundamental gap in policy expressiveness, generative models have emerged as a powerful alternative for constructing sophisticated control policies. While our prior work introducing a Semi-Supervised Variational Autoencoder (SS-VAE) [11] demonstrated the potential of generative approaches, variational autoencoders (VAEs) can suffer from training instability and limited expressiveness in the latent space. To overcome these limitations, this paper introduces a more robust framework, a novel Unified Joint Model (UJM) termed **Diffusion-QL**, for joint resource allocation and network slicing in O-RAN. Our approach uses a single deep neural architecture that learns to jointly detect complex network states and allocate resources across slices by mapping the policy directly to the environment's state space. We circumvent the limitations of prior methods by leveraging a conditional diffusion model, a generative paradigm renowned for its stability and ability to learn complex distributions [12]. We formulate the slicing problem for three distinct services—eMBB, URLLC, and mMTC—and deploy the Diffusion Q-Learning (Diffusion-QL) agent as an xApp within the near-real-time Radio Access Network Intelligent Controller (RIC). Unlike conventional methods that learn a single action, our framework generates a distribution of near-optimal actions guided by a learned Q-function, enabling more expressive and adaptive decision-making. The main contributions of this paper are summarized as follows:

- *Joint O-RAN Slicing Model:* We formally model the joint power and Physical Resource Block (PRB) allocation problem for eMBB, URLLC, and mMTC services, capturing their diverse QoS requirements.
- *Diffusion-QL Framework:* We design Diffusion-QL, a generative DRL agent acting as a UJM that uses a Q-guided diffusion process to overcome the policy expressiveness limitations of prior unimodal DRL approaches.
- *Validation and Robustness:* Through extensive system-level simulations, we demonstrate that Diffusion-QL significantly outperforms state-of-the-art baselines, exhibiting its better adaptability and reliable performance across

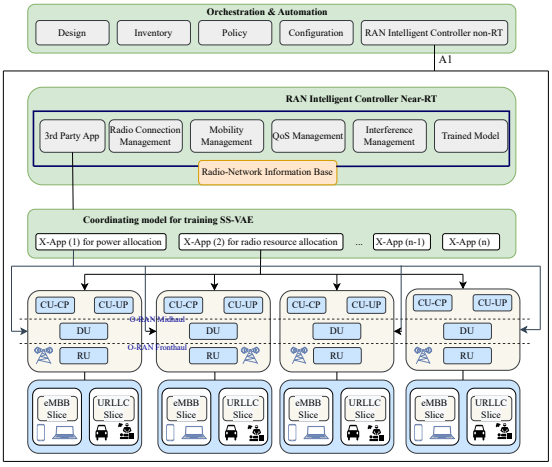


Fig. 1: Architectural overview of the O-RAN system [11].

diverse network scenarios.

- *Computational Feasibility:* We provide a theoretical analysis of our framework, confirming its feasibility for real-time operation within the stringent latency constraints of the O-RAN near-RT RIC.

II. System Model and Problem Formulation

A. System Model

We consider a joint power and PRBs allocation framework within an O-RAN architecture supporting three primary network slices: eMBB, URLLC, and mMTC, each with distinct QoS requirements for latency and throughput. Our model assumes each Radio Unit (RU) serves multiple User Equipments (UEs) distributed across these slices. We consider S_1, S_2, and S_3 slices for eMBB, URLLC, and mMTC, respectively. Each service type $j \in \{e, u, m\}$ comprises S_j slices, and each slice s_j serves U_{s_j} UEs. Consistent with [11], we jointly optimize resources for single-antenna UEs and RUs (Fig. 1).

The system spectrum consists of K PRBs, shared among R single-antenna RUs. Each slice s is allocated $\bar{K}_s$ PRBs, satisfying $\sum_s \bar{K}_s \leq K$. The binary variable $\alpha^r_{u_s} = 1$ indicates the association of UE u in slice s to RU r, where each UE connects to exactly one RU which is shown as $\sum_r \alpha^r_{u_s} = 1, \quad \forall u, s$. The PRB allocation is indicated by the binary variable $\beta^r_{u_s,k} = 1$ if PRB k of RU r is assigned to UE u in slice s. This assignment is valid only if the UE is associated with the RU, and each PRB is exclusively allocated to at most one UE per RU, as enforced by:

$$\beta^r_{u_s,k} \leq \alpha^r_{u_s}, \quad \forall r \in \mathcal{R}, \ \ k \in \mathcal{K}_f, \ \ u \in \mathcal{U}, \ \ s \in \mathcal{S}, \tag{1}$$

$$\sum_{u \in \mathcal{U}} \alpha^r_{u_s} \beta^r_{u_s,k} \leq 1, \quad \forall r \in \mathcal{R}, k \in \mathcal{K}_f, \ \ s \in \mathcal{S}. \tag{2}$$

This exclusive allocation eliminates intra-cell interference. The Signal to Interference & Noise Ratio (SINR) of the u^{th} UE in slice s on PRB k is $\rho^r_{u_s,k} = \frac{\alpha^r_{u_s} \beta^r_{u_s,k} p^r_{u_s,k} |h^r_{u_s,k}|^2}{BN_0 + I^r_{u_s,k}}$, where $I^r_{u_s,k}$ represents the inter-cell interference:

$$I^r_{u_s,k} = \sum_{j \neq r}^{R} \sum_{l=1, l \neq s}^{S} \sum_{i=1, i \neq u}^{U} \alpha^j_{i_l} \beta^j_{i_l,k} p^j_{i_l,k} |h^j_{i_l,k}|^2. \tag{3}$$

Here, $|h^r_{u_s,k}|^2$ is the channel power gain between UE u in slice s and O-RU r on PRB k, and $p^r_{u_s,k}$ is the transmission power allocated to UE u in slice s by O-RU r on PRB k. B is the bandwidth, and N_0 is the Gaussian noise power spectral density, so BN_0 is the noise power of the system. The achievable data rate of the u^{th} UE in slice s served by O-RU r on PRB k is given by $R^r_{u_s,k} = \alpha^r_{u_s} \beta^r_{u_s,k} B \log_2\left(1 + \rho^r_{u_s,k}\right)$. The total achievable rate of UE u in slice s is $R_{u_s} = \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}_s} \alpha^r_{u_s} \beta^r_{u_s,k} B \log_2\left(1 + \rho^r_{u_s,k}\right)$. Accordingly, the total throughput of slices s and the overall system throughput are, respectively, $R_s = \sum_{u \in \mathcal{U}_s} R_{u_s}$ and $R_{\text{tot}} = \sum_{s \in \mathcal{S}} R_s$.

We model the transmission delay D_{u_s} for a packet of average size L_{u_s} bits as a function of the achievable rate, ignoring queuing and processing delays for simplicity, as $D_{u_s} = \frac{L_{u_s}}{R_{u_s}}$. The instantaneous capacity of O-RU r, denoted C_r, is the aggregate data rate of all UEs it serves, given by $C_r = \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \sum_{k \in \mathcal{K}_s} \alpha^r_{u_s} \beta^r_{u_s,k} B \log_2\left(1 + \rho^r_{u_s,k}\right)$.

B. Problem Formulation

Given that the joint resource allocation problem is NP-hard, we formulate an optimization to find a near-optimal solution. Our objective is to maximize the aggregate throughput (Equation (4)), while satisfying strict QoS constraints (Equations (5) – (7)). The problem is formulated as:

$$\max_{\{\alpha, \beta, p\}} \quad \sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \sum_{r \in \mathcal{R}} \sum_{k \in \mathcal{K}_s} \alpha^r_{u_s} \beta^r_{u_s,k} B \log_2\left(1 + \rho^r_{u_s,k}\right), \tag{4}$$

$$\text{s.t.} \quad D_{u_s} = \frac{L_{u_s}}{R_{u_s}} \leq D^{\max}_{u_s}, \quad \forall u, s, \tag{5}$$

$$\sum_{s \in \mathcal{S}} \sum_{u \in \mathcal{U}_s} \sum_{k \in \mathcal{K}_s} \alpha^r_{u_s} \beta^r_{u_s,k} p^r_{u_s,k} \leq P^{\max}_r, \quad \forall r, \tag{6}$$

$$C_r \leq C^{\max}_r, \quad \forall r, \tag{7}$$

$$\sum_r \alpha^r_{u_s} = 1, \quad \forall u, s, \tag{8}$$

$$\sum_u \alpha^r_{u_s} \beta^r_{u_s,k} \leq 1, \quad \forall r, k, s, \tag{9}$$

$$\sum_s \bar{K}_s \leq K, \quad \alpha^r_{u_s}, \beta^r_{u_s,k} \in \{0, 1\}. \tag{10}$$

III. Methodology

We propose a Diffusion-based Reinforcement Learning (Diffusion-RL) model that serves as a UJM for resource allocation in the O-RAN, chosen for its ability to efficiently explore high-dimensional state spaces while providing better generalization and robustness compared to traditional Reinforcement Learning (RL) paradigms [9]. By employing a generative model as the policy, our approach optimizes the exploration-exploitation balance, effectively addressing data sparsity and

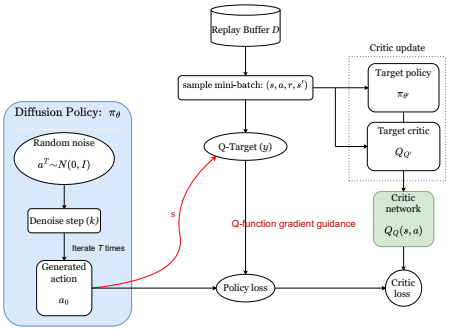


Fig. 2: Overview of the Diffusion-QL training loop.

mitigating the high computational costs associated with conventional online RL training. The Diffusion-QL training loop is illustrated in Fig. 2. Performance of the Diffusion-QL is evaluated against three benchmarks: 1) the exhaustive search algorithm (ESA), which provides a theoretical optimum but is computationally prohibitive for real-time deployment; 2) our prior SS-VAE work, which uses a generative model but requires labeled datasets; and 3) the Deep Q-Network (DQN), a standard DRL baseline that can struggle with generalization.

A. Diffusion-QL Theory

To address the complex resource allocation problem in O-RAN, Diffusion-QL represents the RL policy as a conditional diffusion model. Unlike conventional unimodal policies (e.g., Gaussian), which fail to capture the multimodal distribution of optimal resource allocation, diffusion models offer a highly expressive paradigm that excels at learning complex distributions and provides better training stability than VAEs [9], [12], [13].

The policy ($\pi_\theta(a|s)$) learns to generate actions by iteratively reversing a gradual noising process. The training objective is twofold: it performs implicit behavior cloning by learning to denoise samples, ensuring generated actions remain close to the data distribution, while simultaneously utilizing the gradient of a learned Q-function to maximize long-term returns. Formally, $\pi_\theta(a|s)$ follows the reverse process of a conditional diffusion model. The forward process q is a fixed Markov chain that gradually adds Gaussian noise over T steps according to a variance schedule $\{\beta_t\}_{t=1}^T$. The distribution of a noised action a_t at any step t can be sampled directly from the original action a_0 [13] which is shown as $q(a_t|a_0) = \mathcal{N}(a_t; \sqrt{\bar{\alpha}_t}a_0, (1-\bar{\alpha}_t)\mathbf{I})$, where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$.

The reverse process is parameterized by a neural network $\epsilon_\theta(a_t, t, s)$ that predicts the noise ϵ added at step t, conditioned on the network state s. The network minimizes the denoising score matching loss, serving as an implicit behavior cloning objective [12]: $\mathcal{L}_{diffusion}(\theta) = \mathbb{E}_{t,s,a_0,\epsilon}\left[||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}a_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t, s)||^2\right]$.

To guide the policy towards high-value actions, we integrate a critic network $Q_\phi(s, a)$ trained via the standard Temporal Difference (TD) learning objective from a replay buffer $\mathcal{D}$ [9], [12] which is represented as $\mathcal{L}_{critic}(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - (r + \gamma Q_{\phi_{tgt}}(s', a'))\right)^2\right]$.

During the reverse denoising process, the Q-function gradient steers actions toward high-reward regions by perturbing the sampling steps. The mean for the next denoising step a_{t-1} is adjusted as $\hat{\mu}_\theta(a_t, t, s) = \mu_\theta(a_t, t, s) + w \cdot \Sigma_t \nabla_{a_t} Q_\phi(s, a_t)$ [12], where μ_θ is the predicted mean from the denoising network, Σ_t is the reverse step covariance, and w is a guidance scale hyperparameter. This balances generative exploration with Q-guided exploitation, as summarized in Algorithm 1.

Diffusion-QL stability is rooted in the denoising $\mathcal{L}_{diffusion}(\theta)$, which avoids the mode collapse typical of Generative Adversarial Networks (GANs), and the latent constraints of VAEs. The Policy Gradient Theorem further supports theoretical convergence; as critic Q_ϕ converges to the optimal action-value function via TD-learning, the gradient guidance $w \cdot \Sigma_t \nabla_{a_t} Q_\phi$ ensures reverse sampling shifts generated actions toward high-reward regions of the action space.

Algorithm 1 Diffusion-QL for O-RAN Resource Allocation

Initialize: Diffusion policy π_θ, critic networks Q_{ϕ_1}, Q_{ϕ_2}.
Initialize: Target networks $\pi_{\theta'} \leftarrow \pi_\theta$, $Q_{\phi'_1} \leftarrow Q_{\phi_1}$, $Q_{\phi'_2} \leftarrow Q_{\phi_2}$.
Initialize: Replay buffer $\mathcal{D}$.
for each training iteration **do**
  Sample a mini-batch $B = \{(s, a, r, s')\} \sim \mathcal{D}$.

  // Critic Update
  Sample next action $a' \sim \pi_{\theta'}(\cdot|s')$ via the reverse diffusion process.
  Compute target Q-value:
  $y \leftarrow r + \gamma \min_{i=1,2} Q_{\phi'_i}(s', a')$
  Update critics ϕ_1, ϕ_2 by minimizing:
  $\mathcal{L}_{critic}(\phi_i) = \mathbb{E}_{(s,a)\in B}\left[(Q_{\phi_i}(s,a) - y)^2\right]$

  // Policy Update
  Sample action $a_0 \sim \pi_\theta(\cdot|s)$ via the reverse diffusion process.
  Compute the combined policy loss:
  $\mathcal{L}_{policy} = \mathcal{L}_{diffusion}(\theta) - \lambda \cdot \mathbb{E}_{s\in B, a^0\sim\pi_\theta}[Q_{\phi_1}(s, a^0)]$
  Update policy θ by minimizing $\mathcal{L}_{policy}$.

  // Target Network Update
  $\theta' \leftarrow \rho\theta' + (1-\rho)\theta$
  $\phi'_i \leftarrow \rho\phi'_i + (1-\rho)\phi_i$ for $i \in \{1, 2\}$
end for

B. Markov Decision Process (MDP) Formulation for Diffusion-QL

We formulate the problem as an MDP, enabling the O-RAN orchestrator to act as an intelligent agent:

State: State $\mathfrak{s}(t) \in \mathfrak{S}$, is defined by $\{\mathfrak{s}_u(t)\}_{u=1}^U$, where $\mathfrak{s}_u(t)$ is a binary indicator that equals one if the data rate requirement of UE u is satisfied and zero otherwise. This

TABLE I: Simulation Environment and Hyperparameter Settings.

Parameter	Value
O-RAN Environment	
Cell Radius	400 m
Number of PRBs	50
gNB Transmit Power, P_{max}	46 dBm
Noise Power Spectral Density	-174 dBm/Hz
Channel Model	3GPP TR 38.901
Path Loss Model	Urban Macro (path loss exponent: 3.76)
User Distribution	40/40/20% (eMBB/URLLC/mMTC)
QoS Requirements	
eMBB Min. Rate Req.	10 Mbps
URLLC Min. Rate Req.	2 Mbps
URLLC Max. Delay Req.	1 ms
RL Training	
Optimizer	Adam
Critic Learning Rate	3e-4
Policy Learning Rate	1e-4
Discount Factor, γ	0.98
Target Network Update Rate, ρ	0.005
Replay Buffer Size	200,000
Batch Size	128
Diffusion-QL Model	
Network Architecture	3-Layer MLP
Neurons per Layer	128
Diffusion Timesteps, T	20
Guidance Scale, w	1.2
Noise Schedule, β_t	Linear, 1e-4 to 2e-2

TABLE II: Generalization performance against the ESA benchmark on test data.

Algorithm	MAE ↓	R^2 ↑	Cosine Sim. ↑	BAEP (%) ↓
SS-VAE	0.1407	0.7237	0.9745	5.45
DQN	0.2156	0.6985	0.8915	9.72
Diffusion-QL	**0.1381**	**0.7461**	**0.9808**	**4.19**

state, combined with quantized transmission power levels, enables the model to jointly detect environmental conditions and network demands.

Action: Action $\mathfrak{a} \in \mathfrak{A}$ represents a unified resource allocation decision. This mechanism allows the agent to separate radio resources and power levels among the slices. It contains the UE-RU association indicators and the PRB assignments: $\mathfrak{a} = \{\alpha_{u,b,s}, \{\beta_{u,b,m,s}\}_{m=1}^{M}\}_{b=1}^{B}$. By outputting this unified vector, the model ensures power and frequency are optimized simultaneously rather than in decoupled stages.

Reward: The reward $\mathfrak{R}(\mathfrak{s}, \mathfrak{a})$ is a weighted combination of data rates objective and system constraints as $\mathfrak{R}(\mathfrak{s}, \mathfrak{a}) = \Theta_r R_{u_s} + \Theta_{\text{const}} C^b_{u_s,m} + \Theta_{\text{bias}}$, where Θ_r, Θ_{const}, and Θ_{bias} are the respective weights assigned to the data rate, the constraints, and a bias value. This guides the agent toward actions that maximize throughput while maintaining strict network compliance.

IV. Simulation Results

We evaluate the Diffusion-QL algorithm against three benchmarks: ESA, DQN, and SS-VAE [11]. All experiments were performed using PyTorch [14] on an NVIDIA Volta V100 GPU. Simulation parameters for the O-RAN environment and the Diffusion-QL model are detailed in Table I. To ensure a fair comparison, configurations for all benchmarks are identical to those in [11].

A. Training Stability and Generalization Accuracy

We first analyze the learning dynamics of our proposed Diffusion-QL. Fig. 3a illustrates the average reward per episode, showing a rapid performance increase within the first 500 episodes, followed by stable convergence. This confirms that the agent effectively learns to jointly detect network states and separate resources without the training instabilities.

Generalization performance is quantified in Table II via evaluation on a held-out test set. Diffusion-QL achieves the best results across all metrics, notably attaining the lowest Binary Association Error Percentage (BAEP) (4.19%), which measures the error in the binary allocation decisions (UE association and PRB assignment). This low error assignment demonstrates that the generative diffusion policy captures the discrete allocation structure of the optimal policy more accurately than the DQN (9.72%) or the less expressive SS-VAE (5.45%).

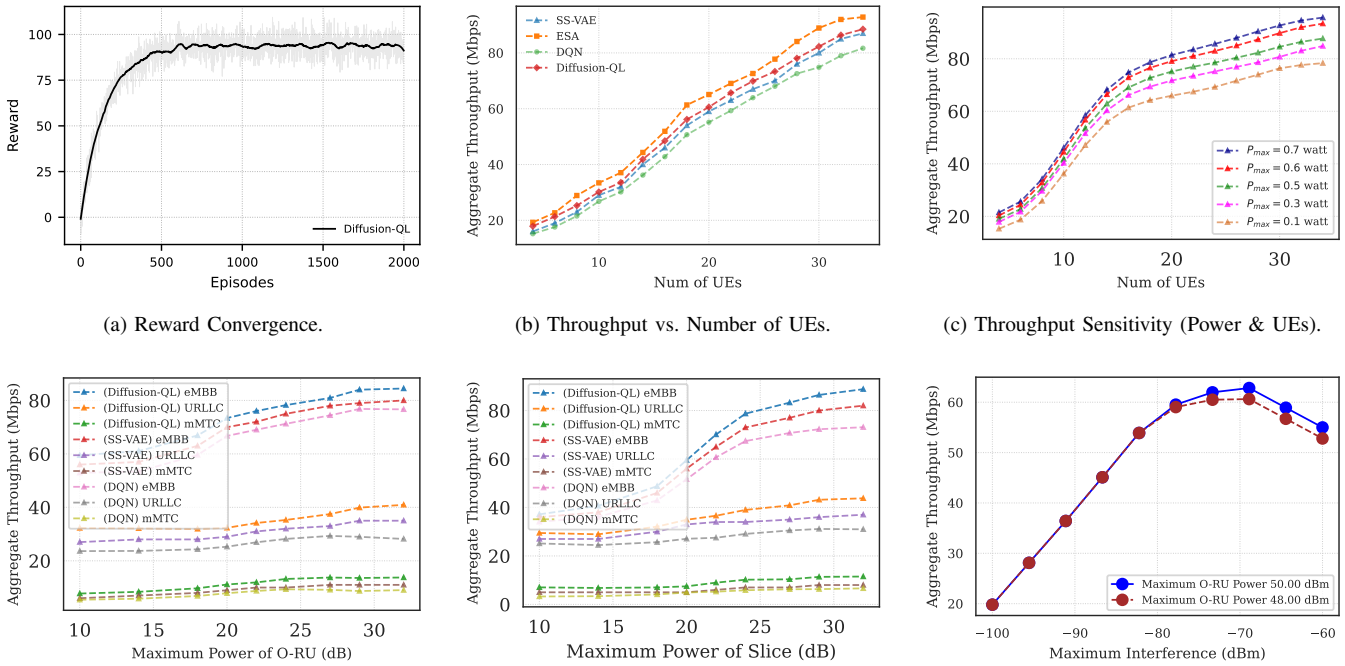
B. Scalability and Throughput Performance

A critical metric for O-RAN controllers is scalability under increasing network load. Fig. 3b evaluates this by plotting aggregate throughput as the number of UEs increases from 5 to 35, with the total O-RU power fixed at 46 dBm as defined in Table I. Diffusion-QL consistently outperforms learning-based baselines and closely tracks the computationally infeasible ESA. Throughput saturates near 35 UEs as the system reaches the physical capacity limits of the 50 available PRBs and the 46 dBm power budget.

Fig. 3c further explores sensitivity to power constraints ($P_{max} \in \{0.1, 0.3, 0.5, 0.6, 0.7\}$Watts). As transmit power of O-RU increases, aggregate throughput rises across all UE densities due to improved SINR and higher-order modulation and coding schemes. This confirms the agent's ability to adaptively utilize the available power budget to maximize spectral efficiency.

C. Per-Slice Performance and Power Sensitivity

To evaluate the management of conflicting QoS requirements of different network slices, we analyze per-slice throughput in Fig. 3d and Fig. 3e with the network load fixed at $U = 25$ UEs as a function of maximum O-RU and maximum slice-specific power, respectively. In both scenarios, Diffusion-QL achieves the highest throughput for each slice (eMBB, URLLC, and mMTC)—when compared to the SS-VAE and DQN baselines. This indicates that the Diffusion-QL finds a more globally efficient allocation that optimizes the whole system, rather than sacrificing one service for another, and correctly prioritizes resources according to service requirements: the high-data-rate eMBB slice while satisfying the stringent constraints of URLLC.



(a) Reward Convergence.

(b) Throughput vs. Number of UEs.

(c) Throughput Sensitivity (Power & UEs).

(d) Per-Slice Throughput vs. RU Power.

(e) Per-Slice Throughput vs. Slice Power.

(f) Robustness to Interference.

Fig. 3: Performance evaluation of Diffusion-QL against benchmarks. (a) Reward convergence during training. (b) Aggregate throughput scalability with increasing UEs. (c) Throughput sensitivity to RU power. (d, e) Per-slice throughput analysis under varying power constraints. (f) Robustness to inter-cell interference

TABLE III: Computational Complexity for Each Algorithm.

Method	Computational Complexity
ESA	$O(\|U\| \cdot \|B\| \cdot (M+1)! \cdot P_{\text{levels}})$
DQN	$O(E \cdot T \cdot (\|S\| \cdot \|A\| + F))$
SS-VAE	$O(E \cdot (D \cdot L + L^2)) + O(N \cdot D \cdot L)$
Diffusion-QL	$O(E \cdot T \cdot (\|S\| \cdot \|A\| + F_{\text{diff}}))$ where $F_{\text{diff}} = O(N_{\text{steps}} \cdot N_{\text{neurons}})$

D. Robustness to Inter-Cell Interference

Finally, the robustness of the Diffusion-QL is tested in an interference-limited scenario with $U = 20$ UEs (Fig. 3f). We assessed the impact of inter-cell interference by monitoring the aggregate throughput of the Diffusion-QL agent across two O-RU power levels (48 dBm and 50 dBm). As the external interference power increases (x-axis), the system performance follows an inverted-U-shaped profile. This trend captures the transition from a successful mitigation phase to an interference-limited state where the SINR degradation becomes too severe to overcome, even with the agent's optimized power allocation.

E. Computational Complexity Analysis

To provide a performance comparison, we evaluate the theoretical computational complexity of each algorithm. The costs for a single training run are summarized in Table III. Parameters are defined as follows: E is the number of episodes (or epochs), T is the steps per episode, $|S|$ and $|A|$ are the state and action space dimensions, F is the complexity of a single neural network forward pass, $|U|$ is the number of UEs, $|B|$ is the number of RUs, M is the number of PRBs, P_{levels} is the number of discrete power levels, D and L are the input and latent dimensions for the VAE, N is the number of samples, and N_{steps} is the number of diffusion denoising steps.

The complexity of each algorithm is rooted in its core operational loop. The ESA exhibits factorial complexity, $O(|U| \cdot |B| \cdot (M+1)! \cdot P_{\text{levels}})$, necessitated by the exhaustive enumeration of all possible UE-RU associations, PRB assignments, and power levels, rendering the ESA computationally infeasible for real-time O-RAN deployment. In contrast, the learning-based methods operate within polynomial complexity. The cost of DQN and Diffusion-QL is primarily driven by the number of training iterations, with each step involving interaction with the environment and a subsequent network update. The complexity of the SS-VAE is determined by its training epochs over the labeled dataset, as detailed in [11].

A key distinction lies in the complexity of the forward pass, F. For DQN and SS-VAE, this involves a single pass through a standard neural network. For our Diffusion-QL model, the effective forward pass F_{diff} is defined by N_{steps} iterations of the denoising network to generate a single unified action. This leads to a fundamental trade-off: while Diffusion-QL achieves better performance due to its expressive generative policy,

it incurs a higher computational cost during inference. We provide a detailed discussion on how this iterative cost can be reconciled with real-time O-RAN requirements in the practical considerations presented in the following section.

F. Practical Implementation Considerations

While our evaluation is simulation-based, the Diffusion-QL is designed for the operational realities of 6G O-RAN.

1) Dynamic Environments and Mobility: The MDP state $\mathfrak{s}(t)$ is structured as a time-varying vector that captures instantaneous channel power gains $|h^r_{u_s,k}|^2$. This formulation enables the UJM to adapt its "detect and separate" logic to the rapid fluctuations characteristic of high-mobility scenarios, ensuring that resource allocation remains resilient even as user distributions and channel conditions shift rapidly over time.

2) Scalability to Large-Scale Networks: The model maintains linear complexity with respect to the action vector dimensionality $\mathfrak{a}$. Consequently, scaling to higher UE densities or a wider variety of specialized slice types only requires expanding the denoising network's output layer without requiring a fundamental architectural redesign.

3) Real-Time RIC Constraints: To meet the critical sub-10ms latency requirements of the near-RT RIC, the iterative overhead of the diffusion process can be mitigated via accelerated sampling. Techniques such as Denoising Diffusion Implicit Models (DDIM) [15] can reduce the denoising steps T to a fraction of the original count—potentially reaching a single-step inference. Furthermore, knowledge distillation [16] can be employed to train a non-iterative student model that replicates the multimodal policy of the Diffusion-QL teacher.

The generative advantage is most evident in multi-peak reward landscapes—for example, distinct PRB assignment strategies that yield identical throughput. Unlike Gaussian policies that average these peaks into low-reward valleys, Diffusion-QL preserves distinct optimal action clusters, allowing the agent to sample from multiple near-optimal strategies simultaneously.

V. Conclusion

This paper introduced Diffusion-QL, a UJM for dynamic resource allocation in O-RAN that leverages a Q-guided diffusion model as a highly expressive policy. Our work addresses the NP-hard problem of jointly allocating power and PRBs to satisfy the heterogeneous QoS demands of eMBB, URLLC, and mMTC services. Simulation results demonstrated that Diffusion-QL significantly outperforms state-of-the-art DRL methods, particularly in replicating the discrete allocation structure of optimal policies and maintaining robustness against inter-cell interference. The success of our approach stems from the diffusion policy's ability to jointly detect complex network states and separate resources via a multimodal distribution of optimal strategies, thereby overcoming the fundamental limitations of unimodal Gaussian policies. By generatively constructing actions, Diffusion-QL achieves better exploration and robustness in dynamic O-RAN environments. Unlike the computationally prohibitive ESA or the label-dependent SS-VAE, our method provides a scalable and flexible solution that learns effectively without requiring pre-generated optimal datasets. Future work will focus on exploring advanced sampling techniques to accelerate iterative denoising and address the computational demands of deploying diffusion policies in the latency-critical control loops of the O-RAN architecture.

Acknowledgements

The work in this paper was supported in part by the Federal Ministry of Research, Technology, and Space (BMFTR), Germany, through the Project 6GEM+ under Grant 16KIS2409K; and in part by the European Union through the 6G-Path project under Grant 101139172.

References

[1] C.-X. Wang, X. You, X. Gao, X. Zhu, Z. Li, C. Zhang, H. Wang, Y. Huang, Y. Chen, H. Haas *et al.*, "On the road to 6g: Visions, requirements, key technologies and testbeds," *IEEE Communications Surveys & Tutorials*, 2023.

[2] E. C. Strinati, G. C. Alexandropoulos, N. Amani, M. Crozzoli, G. Madhusudan, S. Mekki, F. Rivet, V. Sciancalepore, P. Sehier, M. Stark *et al.*, "Toward distributed and intelligent integrated sensing and communications for 6g networks," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 60–67, 2025.

[3] M. K. Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. A. López, "Resource allocation in an open ran system using network slicing," *IEEE Transactions on Network and Service Management*, vol. 20, no. 1, pp. 471–485, 2022.

[4] "O-RAN: Towards an Open and Smart RAN, O-RAN Alliance White Paper, October 2018, O-RAN Alliance."

[5] M. Ji, Q. Wu, P. Fan, N. Cheng, W. Chen, J. Wang, and K. B. Letaief, "Graph neural networks and deep reinforcement learning-based resource allocation for v2x communications," *IEEE Internet of Things Journal*, vol. 12, no. 4, pp. 3613–3628, 2025.

[6] Y. Chen, Y. Sun, H. Yu, and T. Taleb, "Joint task and computing resource allocation in distributed edge computing systems via multi-agent deep reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 11, no. 4, pp. 3479–3494, 2024.

[7] D. Yan, B. K. NG, W. Ke, and C.-T. Lam, "Multi-agent deep reinforcement learning joint beamforming for slicing resource allocation," *IEEE Wireless Communications Letters*, vol. 13, no. 5, pp. 1220–1224, 2024.

[8] Z. Ming, H. Yu, and T. Taleb, "Federated deep reinforcement learning for prediction-based network slice mobility in 6g mobile networks," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11 937–11 953, 2024.

[9] Z. Zhu, H. Zhao, H. He, Y. Zhong, S. Zhang, H. Guo, T. Chen, and W. Zhang, "Diffusion models for reinforcement learning: A survey," *arXiv preprint arXiv:2311.01223*, 2023.

[10] M. K. Motalleb, C. Benzaid, T. Taleb, M. Katz, V. Shah-Mansouri, and J. Kim, "Towards secure intelligent o-ran architecture: vulnerabilities, threats and promising technical solutions using llms," *Digital Communications and Networks*, 2025.

[11] S. Nouri, M. K. Motalleb, V. Shah-Mansouri, and S. P. Shariatpanahi, "Semi-supervised learning approach for efficient resource allocation with network slicing in o-ran," *arXiv preprint arXiv:2401.08861*, 2024.

[12] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," *arXiv preprint arXiv:2208.06193*, 2022.

[13] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[14] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[15] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[16] T. Huang, Y. Zhang, M. Zheng, S. You, F. Wang, C. Qian, and C. Xu, "Knowledge diffusion for distillation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 65 299–65 316, 2023.