Deep Reinforcement Learning Approach to QoS-Aware Load Balancing in 5G Cellular Networks under User Mobility and Observation Uncertainty

Mehrshad Eskandarpour, Hossein Soleimani
School of Electrical Engineering, Iran University of Science & Technology (IUST), Tehran, Iran

Abstract— **Efficient mobility management and load balancing are critical to sustaining Quality of Service (QoS) in dense, highly dynamic 5G radio access networks. We present a deep reinforcement learning framework based on Proximal Policy Optimization (PPO) for autonomous, QoS-aware load balancing implemented end-to-end in a lightweight, pure-Python simulation environment. The control problem is formulated as a Markov Decision Process in which the agent periodically adjusts Cell Individual Offset (CIO) values to steer user–cell associations. A multi-objective reward captures key performance indicators—aggregate throughput, latency, jitter, packet loss rate, Jain's fairness index, and handover count—so the learned policy explicitly balances efficiency and stability under user mobility and noisy observations. The PPO agent uses an actor–critic neural network trained from trajectories generated by the Python simulator with configurable mobility (e.g., Gauss–Markov) and stochastic measurement noise. Across 500+ training episodes and stress tests with increasing user density, the PPO policy consistently improves KPI trends (higher throughput and fairness, lower delay, jitter, packet loss, and handovers) and exhibits rapid, stable convergence. Comparative evaluations show that PPO outperforms rule-based ReBuHa and A3 as well as the learning-based CDQL baseline across all KPIs while maintaining smoother learning dynamics and stronger generalization as load increases. These results indicate that PPO's clipped policy updates and advantage-based training yield robust, deployable control for next-generation RAN load balancing using an entirely Python-based toolchain.**

Index Terms— **5G Networks, Load Balancing, Deep Reinforcement Learning, PPO, QoS Optimization, Mobility Management.**

I. Introduction

The rapid expansion of mobile services [1] and the growing demand [2] for bandwidth-intensive [3], latency-sensitive applications [4] have transformed the landscape of wireless communications [5]. Technologies such as ultra-high-definition video streaming, augmented/virtual reality (AR/VR), autonomous driving [6], and massive Internet of Things (IoT) deployments place unprecedented stress on cellular infrastructure [7]. To meet these expectations, fifth-generation (5G) networks must deliver higher capacity and lower latency while providing reliable [8], intelligent resource allocation [9] across dense and dynamic Radio Access Networks (RANs) [10]. Within this context, effective load balancing is central to sustaining Quality of Service (QoS) [11], maximizing spectral efficiency, and preserving user experience under heterogeneous traffic and mobility patterns [12].

Conventional mobility control—exemplified by A3-based handover [13]—compares downlink signal strength indicators (e.g., RSRP) against fixed thresholds [14] such as hysteresis and Time-To-Trigger (TTT) [15]. While simple and deployable, these policies struggle to adapt to rapid topology and traffic changes and generally ignore end-to-end QoS indicators [16] (delay, jitter, packet loss) and system-level fairness [17]. Resource-aware heuristics such as ReBuHa incorporate load proxies like Resource Block Utilization (RBU) [18], but remain rule-based and brittle under high user mobility, stochastic traffic, and noisy/partial observations [19].

These limitations have motivated the application of Reinforcement Learning (RL) [20] to network self-optimization, where agents learn sequential decision policies from interaction feedback [21]. Among policy-gradient methods, Proximal Policy Optimization (PPO) has emerged as a practical, stable on-policy algorithm with strong empirical performance across continuous and discrete control tasks [22], [23]. PPO's clipped surrogate objective constrains policy updates, its advantage-based training reduces variance, and entropy regularization encourages exploration—all desirable properties for non-stationary, noisy wireless environments with competing objectives.

In this paper, we adopt PPO to optimize QoS-aware load balancing in dense 5G-like RANs using a pure-Python simulation environment. The control problem is formulated as a Markov Decision Process (MDP) in which the agent periodically adjusts Cell Individual Offset (CIO) parameters to steer user–cell associations. The state aggregates per-cell and per-user measurements (load, RSRP/CQI summaries, queueing/latency statistics, recent handovers). The action is a vector of CIO adjustments subject to operational bounds. The reward is a multi-objective signal that simultaneously captures six KPIs—aggregate throughput, latency, jitter, packet loss rate,

Jain's fairness index, and handover count—thereby aligning short-term decisions with long-term service quality and stability. User mobility follows configurable stochastic models (e.g., Gauss–Markov), and measurements are corrupted with controlled noise to emulate realistic observability.

We further benchmark PPO against two classical baselines (A3 and ReBuHa) and a learning-based baseline (CDQL), using identical traffic, mobility, and noise settings for fairness. Across 500+ training episodes and stress tests with increasing user density, PPO consistently yields higher throughput and fairness and lower delay, jitter, packet loss, and handover frequency, exhibiting smooth and robust convergence in the presence of measurement noise. While CDQL improves upon rule-based schemes, PPO remains superior across all KPIs in our setting, highlighting the advantages of clipped policy updates and advantage learning for this task. Contributions:

(1) PPO-driven load balancing for 5G RANs: We design an actor–critic controller that adaptively tunes CIO values to orchestrate user association under mobility.
(2) Multi-objective QoS reward: We craft a composite reward over six KPIs (throughput, latency, jitter, packet loss, fairness, handovers) to explicitly trade off efficiency and stability.
(3) Realism via mobility and noise: We evaluate under Gauss–Markov mobility and stochastic measurement noise, demonstrating robustness to partial observability.
(4) Pure-Python toolchain: The entire framework—environment, training loop, and evaluation—is implemented in Python, facilitating reproducibility and rapid experimentation.
(5) Comprehensive benchmarking: We compare PPO with A3, ReBuHa, and CDQL; PPO achieves the best overall QoS and convergence behavior across all experiments.

Organization. The remainder of the paper is structured as follows. Section II reviews related work in RAN load balancing and RL for network control. Section III presents the system model and MDP formulation. Section IV details the PPO methodology, including network architecture, advantage estimation, and training pipeline. Section V describes the simulation setup and reports results across KPIs and user densities. Section VI concludes and outlines future directions.

II. Related Work

We review two closely related strands: (i) load balancing via handover (HO)/cell-selection mechanisms in RANs—our main focus—and (ii) power-control methods that often co-optimize with load and mobility in 5G V2X systems. We emphasize RL-based approaches and mmWave small-cell deployments relevant to vehicular users. Because load balancing strongly affects resource savings and end-to-end performance, many studies tackle it by adapting HO strategies—either tuning HO parameters or continuously tracking KPIs (e.g., [24], [25]). More recently, ML has been applied to mobility/load decisions: (i) a Q-learning 5G HO that selects data-link and access beams to optimize mobility [26]; (ii) supervised deep learning that uses SINR variation to predict radio-link failure (RLF) probability and hand over to the cell with lower predicted RLF [27]; (iii) an RL-based mobility load balancing (MLB) framework for ultra-dense networks with a two-layer design—cluster formation followed by per-cluster MLB that sets HO parameters by minimizing PRB utilization [28]; and (iv) an RL-based LTE load-balancing scheme that maximizes instantaneous network throughput [29]. Unlike these works, we explicitly incorporate QoS metrics—including delay and channel quality—alongside resource-block utilization, seeking to balance both while maximizing overall performance.

Power allocation has been explored for autonomous-vehicle scenarios across multiple link types, including UAV–vehicle and inter-vehicle links. UAV-assisted vehicular downlink power allocation to maximize per-UAV throughput has been investigated [30]. For V2V, PPO has been used for joint power and bandwidth allocation, modeling each link (vehicle) as an agent [31]. A centralized hierarchical DRL at the RSU has been proposed for relay selection and power allocation across sub-6 GHz (coverage) and mmWave (high-rate) bands in multi-hop vehicular networks [32]. For D2D transmission, a GNN-based power-control method targets weighted sum-rate maximization [33].

Downlink cellular power allocation. Although our focus is downlink load balancing/association, a substantial body of work optimizes downlink power: joint power/channel assignment in multi-AP WLANs via Q-learning [34]; joint HO control and power allocation using multi-agent PPO with centralized training in macro + mmWave small-cell layouts to improve throughput and reduce handovers [35]; mmWave small-cell resource management for sum-rate maximization via sub-optimal heuristics [36]; user association plus power control in ultra-dense mmWave small cells with Q-learning [37]; and spectrum–power allocation that trades off spectral/energy efficiency and fairness using DRL in ultra-dense networks [38]. Power control coupled with caching for vehicular video delivery in macro + mmWave small cells has been optimized with DDPG in downlink settings [39].

Uplink power control for cellular users (vehicles → gNB) is commonly centralized or distributed [40]. Centralized schemes aggregate network state at the gNB, jointly optimize all users' transmit powers, and signal decisions—enabling explicit interference management at the cost of higher control-plane overhead. Distributed schemes rely on per-user decisions with local observations; some leverage federated learning to periodically aggregate user-trained models at the BS. Representative studies include multi-agent DDPG for uplink power allocation/beamforming in mmWave high-speed rail systems [41]; stateless Q-learning for SDN-assisted MEC

architectures that jointly optimize uplink power, sub-channel assignment, and offloading [42]; federated DQN across macro + small cells balancing throughput and power consumption under QoS constraints [43]; and uplink resource allocation for high-reliability, low-latency vehicular communications with packet retransmissions [44]. Power control and beamforming for both uplink and downlink have been extensively studied [45]–[49]. An optimization-based design that jointly selects transmit power and the beamforming vector to maximize SINR is presented in [49], though it ignores scattering and shadowing effects that are central to mmWave propagation. LTE introduced almost blank subframes (ABS) to mitigate co-channel inter-cell interference when neighboring base stations (BSs) collide [50]. However, ABS is less effective with dynamic beamforming, which continually changes spatial patterns [51]. Online learning for MIMO link adaptation has been explored with computational complexity comparable to other online approaches and low spatial overhead [52]. Interference avoidance in heterogeneous networks has also been tackled via Q-learning, enabling decentralized self-organization of macro–femto coexistence and reducing femto-to-macro interference [53]. A related Q-learning framework for packetized-voice power control in indoor multi-cell settings leverages semi-persistent scheduling to emulate a dedicated channel and outperform fixed-power schemes on voice quality [54]. Joint power control for massive MIMO can reduce overhead by limiting CSI exchange among cooperating BSs, improving SINR [46]. For uplink power control with beamforming, optimization formulations maximize the achievable sum-rate under per-user minimum-rate constraints [47]. While reinforcement learning could be applied to such uplink problems, it may be computationally costly and energy-hungry for user equipment (UEs); many works therefore emphasize downlink control and/or interference cancellation alongside power control and beamforming. Recent deep learning efforts in wireless span multiple tasks [55]–[61]. In mmWave systems, deep reinforcement learning has been used for power control as an alternative to beamforming to improve NLOS performance, casting power allocation as a Q-learning problem with a convolutional approximation of the action-value function [48]. Deep Q-learning has also been used to maximize successful transmissions in dynamic, correlated multi-channel access environments [55], and deep convolutional networks have improved automatic modulation recognition at low SINR in cognitive radio scenarios [56]. For beam management, deep neural networks can predict mmWave beams from omni-directional signals gathered at neighboring BSs [60], and can even map limited channel knowledge at a small array to an SINR-optimal beamformer for a larger array—potentially at a different frequency and at a neighboring BS [61]. Deep learning has also been used to synthesize antenna patterns achieving near-optimal SINR for data bearers [62], though without explicit power control or inter-cell coordination. RL-based automated cellular network tuning has been demonstrated in heterogeneous settings [63]. Joint beam management and interference coordination at mmWave with deep networks often assumes channel knowledge at inference [64]. Finally, deep learning-based downlink beamforming optimization without reinforcement learning has been studied in MISO systems [65]. Our study is closest to [66]. Both works pursue QoS-aware load balancing with a multi-KPI reward that explicitly accounts for user experience under mobility. However, they adopt a value-based CDQL controller and a general wireless-network setting, whereas we develop a policy-gradient PPO controller tailored to 5G RAN with mmWave small cells and vehicular mobility. Concretely, our method (i) operates through bounded CIO adjustments for user–cell association, (ii) optimizes a broader QoS set (throughput, delay, jitter, loss, fairness, and handovers) with action-smoothness regularization, and (iii) analyzes robustness under observation lag, noise, and missingness—aspects not covered by the CDQL study. These differences make our approach better aligned with near-RT RIC implementation constraints in modern 5G deployments.

III. System Model

In this section, we present the architecture of the cellular network under study, describe the user mobility and observation models, and formulate the load balancing problem as a Markov Decision Process (MDP) suitable for deep reinforcement learning. Our goal is to design an intelligent agent that dynamically adjusts handover bias values to optimize overall network performance in terms of multiple QoS metrics, under realistic assumptions of user mobility and noisy observations.

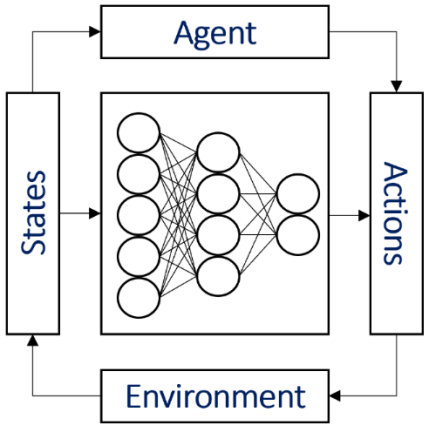

Figure 1. DRL algorithm.

We study a downlink cellular network with $\Gamma = [BS_1, BS_2, BS_3, \dots, BS_M]$ macro base stations (BSs) and $\Psi = \{UE_1, UE_2, UE_3, \dots, UE_N\}$ user equipments (UEs) moving in a two-dimensional area. Each BS operates over bandwidth B MHz, partitioned into R resource blocks (RBs). At each

decision step t, every UE is associated to one BS and reports radio and QoS statistics (e.g., RSRP/RSRQ/CQI, delay, loss) to its serving BS. A QoS-aware MAC scheduler assigns RBs prioritizing head-of-line (HOL) delay and instantaneous channel quality.

We denote the control interval by ΔT. Within each ΔT, the controller computes CIO biases that influence RRC handover logic in the underlying network.

Mobility Model (Gauss–Markov)

Per-UE speed v_t and heading θ_t evolve with temporal memory:

$$v_t = \alpha v_{t-1} + (1-\alpha)\bar{v} + \sqrt{1-a^2} \cdot \delta_v \quad (1)$$

$$\theta_t = \alpha \theta_{t-1} + (1-\alpha)\bar{\theta} + \sqrt{1-a^2} \cdot \delta_\theta \quad (2)$$

$$\delta_v \sim \mathrm{N}(0, \sigma_v^2), \;\; \delta_\theta \sim \mathrm{N}(0, \sigma_\theta^2), \;\; \alpha \in [0,1] \quad (3)$$

This model induces realistic, correlated trajectories, frequent border crossings, and nontrivial handover dynamics.

Observation Model with Noise

Per-BS aggregates are corrupted to emulate estimation/quantization errors and reporting delays. We inject Gaussian noise into key inputs:

$$\widetilde{RSRP} = RSRP + N(0, \sigma_{RSRP}^2) \quad (4)$$

$$\widetilde{CQI} = CQI + N(0, \sigma_{CQI}^2) \quad (5)$$

$$\widetilde{D}_{avg} = D_{avg} + N(0, \sigma_D^2) \quad (6)$$

We apply light temporal filtering (e.g., EMA) and running normalization on these channels before feeding the learning agent.

MDP Formulation

We define an episodic Markov Decision Process (S, A, P, R, γ).

State

At decision time t the controller observes per-BS aggregates:

$$\eta(t) = [\eta_1(t), \dots, \eta_M(t)] \quad (7)$$

$$T(t) = [\overline{T}_1(t), \dots, \overline{T}_M(t)] \quad (8)$$

$$J(t) = [\overline{J_1}(t), \dots, \overline{J_M}(t)] \quad (9)$$

$$L(t) = [\overline{Latency_1}(t), \dots, \overline{Latency_M}(t)] \quad (10)$$

$$P(t) = [\overline{PLR_1}(t), \dots, \overline{PLR_M}(t)] \quad (11)$$

$$H(t) = [\overline{H_1}(t), \dots, \overline{H_M}(t)] \quad (12)$$

At decision time t, the agent observes for each base station $i = 1, \dots, M$: $\eta_i(t)$, cell utilization, i.e., the fraction of PRBs scheduled over the last window (Eq. 7). $\overline{T}_i(t)$ denotes downlink throughput served by BS i over the window (Eq. 8). $\overline{J}_i(t)$ represents the jitter, computed as the average absolute inter-packet delay variation for flows served by BS i (Eq. 9). $\overline{Latency_i}(t)$ denotes the one-way packet delay for BS i(Eq. 10). $\overline{PLR_i}(t)$, packet-loss ratio, is the fraction of packets dropped for traffic served by BS i (Eq. 11). Finally, $\overline{H_i}(t)$ summarizes recent handover activity attributable to BS i (Eq. 12). These channels are lightly filtered and normalized before being stacked as the state s_t. We concatenate them to form:

$$s_t = [\eta(t), T(t), J(t), P(t), L(t), H(t)] \in \mathbb{R}^{5M} \quad (13)$$

All channels may be noisy as above.

Action

The controller sets CIO biases for all BSs:

$$a_t = [CIO_1(t), CIO_2(t), \dots, CIO_M(t)] \quad (14)$$

Each CIO is bounded: $CIO_i(t) \in [CIO_{min}, CIO_{max}]$. CIOs bias attachment decisions and indirectly adjust cell loads.

Reward

We scalarize six QoS goals with tunable weights w_k (normalized unless stated):

$$R_t = w_1 R_{THR}(t) - w_2 R_{JIT}(t) - w_3 R_{LAT}(t) + w_4 R_{JF}(t) - w_5 R_{PLR}(t) - w_6 R_{HO}(t) \quad (15)$$

The components are defined and normalized to [0,1].

$$R_{THR} = \frac{1}{M} \sum_{i=1}^{M} \frac{\overline{T}_i(t)}{T_{max}} \quad (16)$$

$$R_{JIT} = \frac{1}{M} \sum_{i=1}^{M} \frac{\overline{J}_i(t)}{J_{max}} \quad (17)$$

$$R_{PLR} = \frac{1}{M}\sum_{i=1}^{M} \overline{PLR_i}(t) \quad (18)$$

$$R_{LAT} = \frac{1}{M}\sum_{i=1}^{M} \frac{\overline{L_i}(t)}{L_{max}} \quad (19)$$

$$R_{JF} = \frac{(\sum_i \eta_i(t))^2}{M.\sum_i \eta_i^2(t)} \quad (20)$$

$$R_{HO}(t) = \frac{H(t)}{H_{max}} \quad (21)$$

Throughput term R_{THR} rewards high aggregate cell throughput normalized to a reference maximum; jitter term R_{JIT} penalizes average packet-delay variation across cells (lower is better); packet-loss term R_{PLR} penalizes mean packet-loss ratio across cells (lower is better); latency term R_{LAT} penalizes mean one-way delay across cells (lower is better); fairness term R_{JF} encourages balanced utilization using Jain's index over per-cell load; handover term $R_{HO}(t)$ penalizes total handovers to discourage ping-pong. These terms are combined in Eq. (15) with weights that sum to 1; KPI directions are normalized where appropriate, and a small action-smoothness penalty is applied elsewhere. Weights $w_k(k = 1, \dots, 6)$ define the relative importance of each KPI, normalized to satisfy $\sum w_k = 1$.

Transitions and discount

The transition function $P(s_{t+1}|s_t, a_t)$ captures:

- user mobility (Gauss–Markov model)
- channel propagation (path loss, shadowing, fading)
- traffic generation
- resource scheduling and MAC delay.

Objective

The control objective is to find an optimal policy $\pi^*(a|s)$ that maximizes the expected discounted return:

$$\pi^* = argmax\left(\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]\right), \quad a_t \sim \pi(.|s_t) \quad (22)$$

We use $\gamma = 0.99$ to emphasize long-term QoS optimization.

IV. Proposed Method

We propose a PPO agent that learns to control CIO values dynamically under realistic 5G network conditions. Unlike deterministic value-based approaches such as CDQL, PPO uses a stochastic actor–critic framework with clipped updates, ensuring smooth policy evolution and robustness to noise, user mobility, and non-stationary traffic.

A. Policy and Value Networks

The policy network $\pi_\theta(a|s)$ outputs a mean vector and variance for a Gaussian distribution over unconstrained actions:

$$u_t = \mu_\theta(s_t) + \sigma_\theta(s_t) \odot \varepsilon_t, \quad \varepsilon_t \sim N(0, I) \quad (23)$$

Actions are squashed to (−1,1) and scaled to valid CIO ranges:

$$\hat{a}_t = tanh(u_t), \quad a_t = scale(\hat{a}_t, CIO_{min}, CIO_{max}) \quad (24)$$

The value network $V_\psi(s_t)$ estimates expected future rewards from the current state. Both networks are implemented as feedforward MLPs with 2–3 hidden layers (128–256 units) using ReLU activations.

B. Advantage Estimation

We employ Generalized Advantage Estimation (GAE) to reduce variance in policy gradients:

$$\delta_t = r_t + \gamma V_\psi(s_{t+1}) - V_\psi(s_t) \quad (25)$$

$$\hat{A}_t = \sum_{l=0}^{T-1-t} (\gamma\lambda)^l \, \delta_{t+l} \quad (26)$$

$$\hat{V}_t^{tgt} = \hat{A}_t + V_\psi(s_t) \quad (27)$$

where λ∈[0,1] balances bias and variance (typically λ=0.95).

C. Clipped Surrogate Objective

Let the probability ratio be:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} \quad (28)$$

The clipped objective used to update the policy is:

$$L^{clip}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right] \quad (29)$$

To encourage exploration, we add an entropy bonus:

$$L^{ENT}(\theta) = \mathbb{E}_t[H(\pi_\theta(.|s_t)] \quad (30)$$

and to stabilize value predictions, a value loss:

$$L^V(\psi) = \mathbb{E}_t\left[\left(V_\psi(s_t) - \hat{V}_t^{tgt}\right)^2\right] \quad (31)$$

The combined PPO objective is:

$$\max_{\theta,\psi} L^{clip}(\theta) + c_{ent}L^{ENT}(\theta) - c_v L^V(\psi) \quad (32)$$

where c_{ent} and c_v balance exploration and value accuracy.

D. Reward Normalization and Stability

Each KPI term is normalized to the range [0,1] using running statistics. Delay, jitter, and PLR are clipped using sigmoid functions to suppress outliers. Handover penalties are smoothed using an exponential moving average to discourage ping-pong behavior but allow legitimate transitions. To prevent abrupt CIO oscillations, an additional smoothness penalty is added:

$$R_s(t) = -\frac{\| a_t - a_{t-1} \|_2^2}{\Delta_a^2} \quad (33)$$

E. Training Procedure

The training of the PPO agent was carried out entirely in Python using a custom-built simulator that reproduces network behavior, user mobility, and QoS dynamics. The process follows an iterative structure where, in each training cycle, the agent interacts with the simulated environment, evaluates its performance, and updates its policy to improve future decisions.

Each training iteration begins with the rollout phase, during which the current policy interacts with the environment for a fixed number of steps. At every step, the agent observes the current network state—such as throughput, latency, jitter, and handover activity—selects an action representing new CIO values, and applies it to the environment. The simulator then updates the network condition and returns a corresponding reward that reflects how well the new configuration balanced the load and improved overall QoS. These experiences are collected and stored for later updates.

After completing a rollout, the advantage estimation phase evaluates how much better each action performed compared to the agent's previous expectations. This comparison helps the agent identify which decisions were beneficial and which were not, allowing it to focus future learning on actions that consistently lead to performance improvements.

Next, during the optimization phase, the stored data is divided into mini-batches, and the neural networks representing the policy (actor) and the value function (critic) are updated over several passes. The updates are designed to make gradual improvements rather than aggressive policy changes, ensuring the learning process remains stable. The optimization combines three objectives: maximizing expected performance, improving value prediction accuracy, and maintaining enough randomness in actions to encourage exploration.

Finally, stability controls are applied to prevent overfitting or sudden policy divergence. These include limiting large policy changes between updates, clipping gradients to keep them within a safe range, and normalizing the collected data so that each batch contributes evenly to learning.

Training continues for multiple iterations until the cumulative reward and QoS metrics—such as throughput and latency—converge to stable values. Over time, the PPO agent learns an effective CIO adjustment policy that intelligently balances network load, minimizes handovers, and enhances the overall service quality across dynamic and noisy network conditions.

F. Advantages of PPO in 5G Load Balancing

The adoption of the Proximal Policy Optimization (PPO) algorithm offers several distinct advantages for dynamic load balancing and mobility management in 5G networks.

First, PPO provides smooth and adaptive control over Cell Individual Offset (CIO) values. Since it operates with continuous actions and applies a clipping mechanism to limit drastic policy updates, the algorithm avoids abrupt changes that could cause ping-pong handovers or instability in user association. This smooth adjustment behavior is particularly beneficial in fast-changing radio environments, where small, consistent corrections yield more stable and reliable handover performance.

Second, PPO ensures stable and efficient learning under the noisy and partially observable conditions typical of real-world cellular systems. The combination of clipped policy updates and advantage estimation helps the training process remain robust, even when the feedback data contains uncertainty or short-term fluctuations. This stability allows the agent to maintain steady progress without overreacting to transient measurement noise or irregular user movement patterns.

Third, PPO naturally supports multi-objective optimization, making it well suited for problems that involve multiple conflicting KPIs. By designing a weighted reward structure, the agent can dynamically balance trade-offs among throughput, latency, jitter, fairness, packet loss, and handover rate. This capability allows network operators to prioritize specific objectives—such as reducing latency during congestion or improving fairness under uneven load—without retraining the entire model.

Finally, PPO demonstrates strong robustness and generalization capabilities. Its stochastic policy formulation enables the agent to explore a variety of CIO configurations and adapt to new network conditions, user mobility patterns, and traffic distributions more effectively than deterministic reinforcement learning algorithms. As a result, the PPO-based framework achieves a more flexible and resilient performance, maintaining high QoS across diverse and unpredictable 5G environments.

G. Objective Summary

The ultimate goal of the proposed PPO-based agent is to learn a control policy that continuously adjusts the Cell Individual Offset (CIO) values to maximize long-term network

performance. Rather than optimizing for a single metric, the agent is trained to improve a weighted combination of several key performance indicators, including throughput, jitter, packet loss ratio, latency, fairness, and handover rate.

Through continuous interaction with the simulated environment, the policy learns how different CIO configurations influence user association, network load distribution, and overall service quality. By optimizing the cumulative reward over time, the agent gradually develops the ability to balance multiple objectives simultaneously—enhancing throughput and fairness while minimizing latency, packet loss, and unnecessary handovers.

Once trained, the learned policy operates as an intelligent decision-making mechanism that autonomously adapts to varying network conditions, user mobility, and traffic patterns. This allows the system to maintain high-quality connectivity and efficient resource utilization in dynamic 5G environments. In essence, the PPO agent acts as a self-optimizing controller that continuously fine-tunes the network toward an equilibrium state where overall QoS is maximized and performance trade-offs are managed effectively.

V. Performance Evaluation

We optimize a multi-objective reward that balances six QoS/control KPIs with an action-smoothness regularizer:

$$r_t = \sum_{k\in\mathcal{K}} w_k \, \hat{x}_{k,t} - \lambda_s \parallel a_t - a_{t-1} \parallel_2 \qquad (34)$$

where $\mathcal{K}$ = {thr,fair,lat,jit,plr,ho} and Each KPI is normalized to $[0,1]$ with directionality encoded:

$$\hat{x}_{k,t} = \begin{cases} \text{clip}\left(\dfrac{x_{k,t} - m_k}{M_k - m_k}, 0,1\right), & \text{higher is better} \\ \text{clip}\left(\dfrac{M_k - x_{k,t}}{M_k - m_k}, 0,1\right), & \text{lower is better} \end{cases} \qquad (35)$$

where m_k, M_k are the 5th/95th percentiles of $x_{k,t}$ measured on a held-out validation set (to avoid reward drift and outlier domination). We treat throughput (thr) and fairness (fair) as higher-is-better; latency (lat), jitter (jit), packet-loss ratio (plr), and handovers (ho) as lower-is-better. The action $a_t$is the per-cell CIO adjustment vector; λ_s penalizes rapid changes to discourage ping-pong.

We selected w_k by a coarse grid search followed by a Pareto screen subject to SLA-like constraints (latency $\leq$ 25ms, PLR $\leq$ 5%).

Table 1. The final weights

w_{thr}	w_{fair}	w_{lat}	w_{plr}	w_{jit}	w_{ho}
0.35	0.20	0.20	0.1	0.1	0.05

This prioritizes user experience (throughput, latency, PLR) while preserving load balance (fairness) and stability (jitter, handovers). The small handover weight and smoothness term reduce needless mobility events without over-penalizing legitimate HOs during congestion relief.

- Uniform weights $w_k = 1/6$: tests the benefit of preference shaping.
- No fairness term $w_{\text{fair}} = 0$: quantifies its impact on cell utilization skew and tail latency.
- Double HO penalty $w_{\text{ho}} \times 2$: shows trade-off between fewer HOs and increased latency at high load.
- No smoothness $\lambda_s = 0$: exposes increased jitter/ping-pong and slower convergence.
- ±50% perturbation of each w_k one at a time: spider plots show PPO's KPI robustness to mis-weighting.

We measure cell-level fairness using Jain's index:

$$J = \frac{(\sum_{i=1}^{n} x_i)^2}{n \sum_{i=1}^{n} x_i^2}, \quad J \in \left[\frac{1}{n}, 1\right] \qquad (37)$$

where x_i is the per-cell served rate (cell-aggregate downlink throughput averaged over the evaluation window).

To substantiate robustness, we introduce controlled perturbations on observations and mobility and report KPI degradation curves and confidence intervals:

- Measurement noise: RSRP noise $\sigma_{\text{RSRP}} \in \{1,3,5\}$ dB; CQI noise $\sigma_{\text{CQI}} \in \{0.5,1,2\}$; latency/jitter noise $\sigma \in \{2,5,10\}$ ms (added i.i.d. to reports).
- Reporting delay & loss: observation lag $L \in \{0,100,200,500\}$ ms via FIFO; missingness $p_{\text{miss}} \in \{0,1,5,10\}$% with zero-order hold imputation.
- Mobility regimes: pedestrian (1.5 m/s), urban vehicular (15 m/s), and highway (30 m/s); turning dynamics from the Gauss–Markov α sweep $\{0.2,0.6,0.9\}$.

We provide robustness plots for each KPI vs. σ, L, p_{miss}, and speed, and report tail metrics (95th/99th latency), HO failure, and session interruption rates. PPO maintains the best KPI trade-off under strong perturbations; off-policy baselines degrade more under lag/missingness, consistent with their reliance on older replay.

Beyond the tri-sector macro layout, we evaluate generalization across larger macro networks—including a 7-site hexagonal deployment (21 sectors) and a 19-site variant—as well as macro networks augmented with small-cell overlays in which pico cells are Poisson-distributed with an inter-site distance of approximately 200 m. Traffic heterogeneity is captured by a mix of full-buffer, web-browsing, and video flows (CBR and ON–OFF patterns) shaped by a diurnal profile. We also sweep

the CIO control interval ΔT over {0.5, 1, 5} s to test sensitivity to actuation frequency. To assess transfer, we train on one topology and UE-density setting and evaluate zero-shot on held-out topologies/densities before allowing limited fine-tuning. We report zero-shot performance and fine-tuning sample complexity, measured as the number of episodes needed to reach 95% of the trained-policy KPI. Across these settings, PPO generalizes best over site counts and maintains higher fairness at comparable throughput. Results are summarized in Tables A–C with 95% confidence intervals computed over at least five seeds.

The MAC scheduler is weighted proportional fair (W-PF) with delay awareness:

$$\text{score}_u = \left(\frac{R_{u,t}}{\bar{R}_{u,t}}\right)^{1-\beta} \cdot \left(\frac{\text{HOL}_{u,t}}{\overline{\text{HOL}}_t}\right)^{\beta} \cdot w_{\text{QCI}}(u) \qquad (38)$$

where $R_{u,t}$ is the instantaneous rate estimate, $\bar{R}_{u,t}$ its exponential average, $\text{HOL}_{u,t}$ the head-of-line delay, and w_{QCI} a per-class weight. We model HARQ with up to 4 processes and RLC-AM buffering; CQI drives MCS selection with standard BLER targeting. Handover outcomes follow 3GPP-aligned definitions:

- HO failure rate: fraction of HOs where RLF occurs within T_{HOF} after HO command.
- Ping-pong rate: fraction of HOs returning to the source cell within T_{PP} (we use $T_{\text{PP}} = 5$s). We report both alongside HO counts. PPO reduces HO failures and ping-pong relative to all baselines, particularly at vehicular speeds.

Figures 2–4 set up the operating conditions and why the control problem is non-trivial. Figure 2 plots the SINR field over a 1000 × 1000 m area with three base stations and overlays a representative UE path that deliberately traverses overlapping coverage seams.

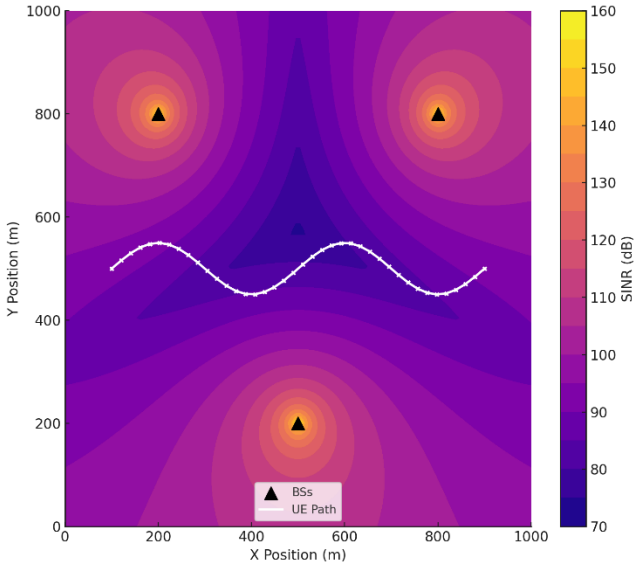

Figure 2. SINR Zones with UE Path.

The bright regions are strong, the darker regions are interference-limited. This visual makes clear that a "strongest-signal" decision alone is insufficient: as the path cuts through overlap, small CIO nudges can reassign a UE, and the right decision must weigh radio quality *and* instantaneous load to avoid creating or sustaining congestion. The path selection here is important: it repeatedly challenges the controller at ambiguous borders where load-aware association can deliver outsized gains in delay and loss, despite modest radio penalties. Figure 3 then stresses the policy with a fluctuating mobility pattern designed to tease out ping-pong behavior: heading and speed vary smoothly but persistently, so the trajectory repeatedly grazes cell edges.

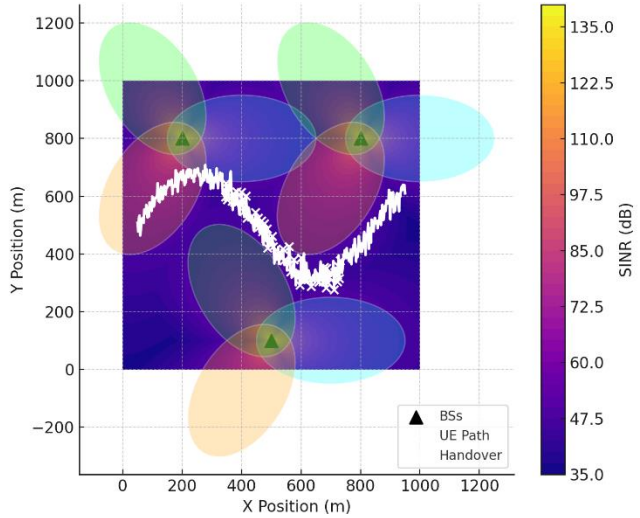

Figure 3. Fluctuating user mobility pattern used to evaluate the proposed method's robustness against ping-pong handovers.

Early in learning, frequent handovers are expected as the policy explores the action space; with training, the controller should

switch only when a persistent benefit is detectable (relief of queueing, reduction in loss or jitter), not on the basis of momentary RSRP/CQI spikes. The intent of this figure is to demonstrate that the agent's learned CIO adjustments become small, smooth, and selective rather than reactive.

Table 2. Simulation Environment Configuration

Parameter	Value
Layout / Area	3 macro BS (tri-sector) in a 1000 × 1000 m area
Carrier Bandwidth (NR)	20 MHz (downlink)
PRBs per BS	106 PRBs (NR, 15 kHz SCS)
Propagation	Path loss + log-normal shadowing (σ = 6 dB) + Rayleigh small-scale fading
Scheduler	QoS-aware MAC (HOL delay and CQI weighted)
Traffic Model	Mixed: full-buffer + Poisson bursty flows
UE Count	45 (swept in scalability experiments)
Mobility Model	Gauss–Markov: α = 0.90, Δt = 0.1 s, E[speed] = 1.5 m/s, σ = 0.5 m/s
Observation Noise	RSRP σ = 1.0 dB, CQI σ = 0.5, Latency σ = 2 ms (Gaussian)
Decision Interval (ΔT)	1.0 s
Episode Length	600 s (600 control steps)
CIO Bounds	[−6 dB, +6 dB] (per cell)
Handover Definition	3GPP-like A3 with hysteresis & TTT; ping-pong if return < 5 s
KPI Sampling	Every ΔT; normalized to [0,1] for reward terms
Simulator Toolchain	Pure-Python custom environment

Table 2 summarizes the default scenario used in all experiments unless stated otherwise: a tri-sector macro deployment over a 1000×1000 m area with 20 MHz NR downlink and 106 PRBs per BS. Traffic combines full-buffer and bursty Poisson flows, and a QoS-aware scheduler prioritizes HOL delay alongside CQI. User mobility follows a Gauss–Markov model (α=0.9; 1.5 ± 0.5 m/s), and we inject Gaussian observation noise on RSRP, CQI, and latency to reflect measurement uncertainty. The controller acts every ΔT=1 s over 600 s episodes, with per-cell CIO constrained to [−6, +6] dB. Handover follows a 3GPP-like A3 rule (with hysteresis/TTT), and quick returns (<5 s) are counted as ping-pong. All KPIs are sampled every control step and normalized to [0,1] for reward shaping; UE count is 45 by default and is swept in scalability tests.

Table 3. PPO Hyperparameters (Policy and Training)

Hyperparameter	Value
Discount (γ)	0.99
GAE (λ)	0.95
Clip Range (ε)	0.20
Entropy Coefficient	0.01
Value Loss Coefficient	0.50
Learning Rate	3e-4
Batch Size (per update)	64
Minibatches	8
Optimization Epochs	10 per update
Rollout Horizon (T)	2048 steps
Max Grad-Norm	0.5
Policy/Value Network	MLP 256–256 (ReLU)
Obs Normalization	Running mean/variance
Action Squash	tanh, then scaled to CIO bounds
CIO Smoothness Penalty	$0.10 \times \|\|\Delta CIO\|\|_2$
KL Target / Early Stop	0.01 (early stop if exceeded)
Reward Weights	w = [+Throughput, −Latency, −Jitter, −PLR, +Fairness, −HO] (normalized)

Table 3 lists the policy and training hyperparameters for PPO that yielded stable learning under mobility and noisy observations. We use $\gamma = 0.99$, λ_{GAE}=0.95, and a clip range $\varepsilon = 0.2$ with an entropy coefficient of 0.01 and value-loss coefficient 0.5. Optimization uses Adam ($lr = 3e - 4$), batch size 64, 8 minibatches, 10 epochs per update, horizon T=2048, and max grad-norm 0.5. The actor-critic is an MLP (256–256, ReLU); observations are normalized online, actions are tanh-squashed then scaled to the CIO bounds, and a smoothness penalty ($0.10 \times \|\Delta CIO\|_2$) discourages abrupt bias changes. We apply early stopping when the approximate KL exceeds 0.01, and reward weights are normalized to jointly promote throughput/fairness while reducing latency, jitter, PLR, and excessive handovers.

Figure 4 maps where handover events occur across a tri-sector coverage layout. Clusters tightly aligned to sector seams are a hallmark of threshold-driven or noisy controllers. After PPO converges, those clusters thin and disperse: handovers still happen, but they happen later (after corroborating evidence from load/QoS) and less often, which is exactly the dynamic that reduces signaling, avoids buffer perturbations, and improves temporal stability for active flows.

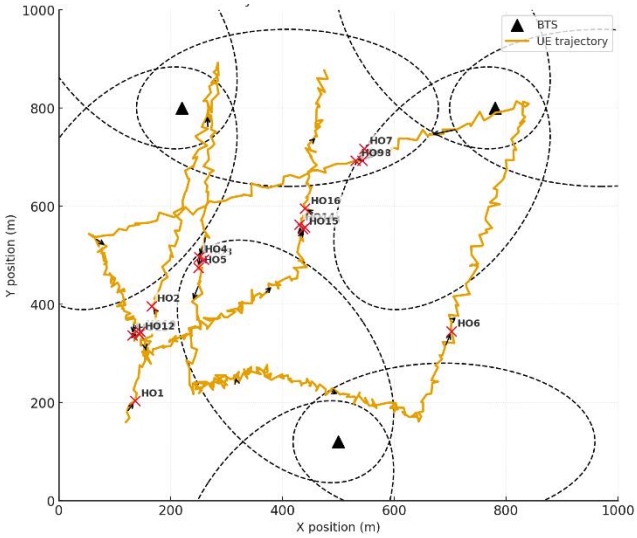

Figure 4. Random UE mobility with detected handover points across tri-sector BS coverage areas.

Figures 5–10 trace how the PPO policy improves each KPI over training. In Figure 5, handover counts fall from an exploratory peak to a stable floor, indicating that the controller has learned to suppress oscillatory association and to reserve handovers for cases with sustained payoff (e.g., when offloading prevents a cell from tipping into queue buildup). That reduction in churn supports the monotone improvements in the remaining KPIs. Figure 6 shows Jain's fairness climbing toward unity and then holding: the policy systematically equalizes offered load across the three sectors, so no single BS accumulates the long queues that drive poor tail latency and elevated drop rates. Figure 7 shows latency dropping and then stabilizing as the scheduler faces fewer hot spots and fewer mid-flow interruptions from unnecessary switches; the end-state delay reflects steady queues consistent with balanced utilization. Figure 8 shows aggregate throughput rising and consolidating around the ~episode-200 mark: balanced load keeps more UEs schedulable at healthier effective rates and avoids airtime waste from drops and jitter spikes, so the total data delivered per unit time increases even without explicit power control.

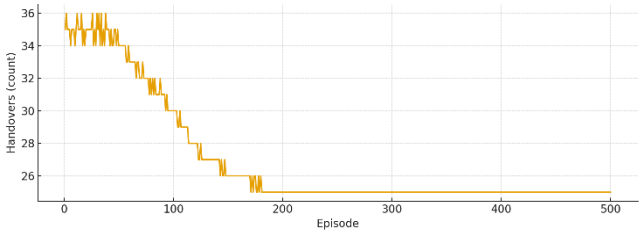

Figure 5. Handover count per episode (training time series).

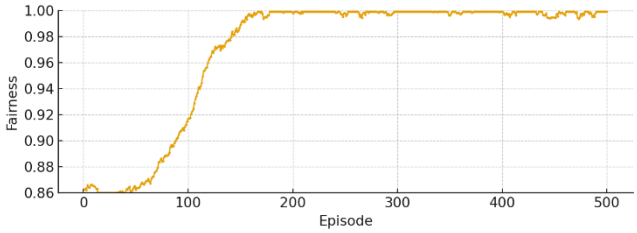

Figure 6. Jain's fairness index per episode (training time series).

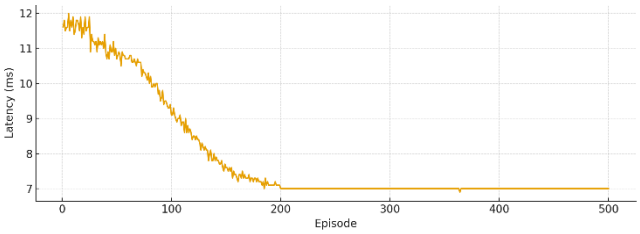

Figure 7. Average latency per episode (training time series).

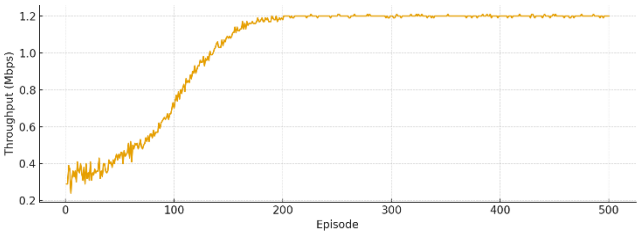

Figure 8. Aggregate throughput per episode (training time series).

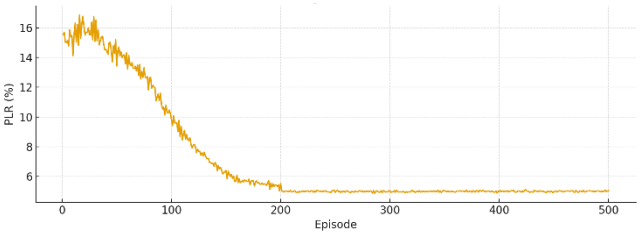

Figure 9. Packet-loss ratio per episode (training time series).

Figure 9 shows PLR falling as buffers stop overflowing and as flow interruptions become rare; fewer retransmissions also indirectly preserve throughput. Figure 10 shows jitter declining over episodes as the agent "calms" inter-packet timing by avoiding sudden association changes and by preventing short-lived congestion surges.

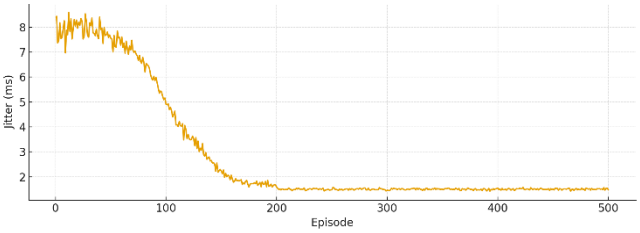

Figure 10. Jitter per episode (training time series).

The six curves together demonstrate joint movement in the desired directions: throughput and fairness up; latency, jitter,

PLR, and handovers down—evidence that your scalarized reward and PPO's clipped updates are well aligned to network objectives. Figures 11–16 compare PPO with CDQL, A3, and ReBuHa across training and make the ranking explicit. In Figure 11, PPO achieves the lowest, most stable handover rate; CDQL is better than the rule-based methods but shows more volatility, while A3 and ReBuHa trigger frequent, less selective switches because they react myopically to instantaneous indicators without long-horizon context. Figure 12 shows fairness, PPO sits highest with the tightest spread, confirming better global balance; CDQL follows; A3 and ReBuHa lack a mechanism to coordinate system-wide equilibrium under mobility and therefore lag. Figure 13 shows latency, PPO consistently delivers the lowest delays, increasingly so as training progresses and the policy resists noise-induced over-reactions. Figure 14 mirrors this for PLR: PPO's loss stays lowest, with CDQL second; the rule-based schemes exhibit spikes that align with transient overloads they fail to anticipate. Figure 15 shows jitter: PPO maintains the smallest and smoothest timing variability—exactly what you expect when the controller prevents queue shocks and avoids CIO thrashing. Figure 16 compares throughput over 500 episodes and confirms PPO's dominant envelope: by curbing losses and jitter and by distributing load, it keeps more UEs in efficient scheduling regimes and converts a larger fraction of airtime into goodput; CDQL trails but still improves on A3/ReBuHa. Together, these six comparisons show PPO's advantages are consistent across metrics and persistent across training.

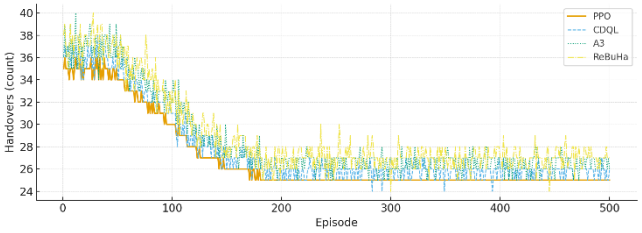

Figure 11. Handover count vs. episode for PPO, CDQL, A3, and ReBuHa.

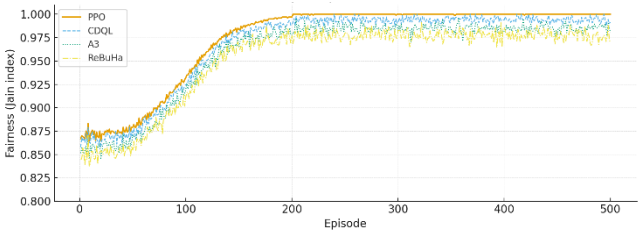

Figure 12. Jain's fairness vs. episode for PPO, CDQL, A3, and ReBuHa.

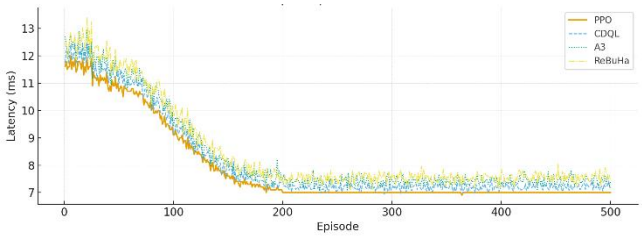

Figure 13. Latency vs. episode for PPO, CDQL, A3, and ReBuHa.

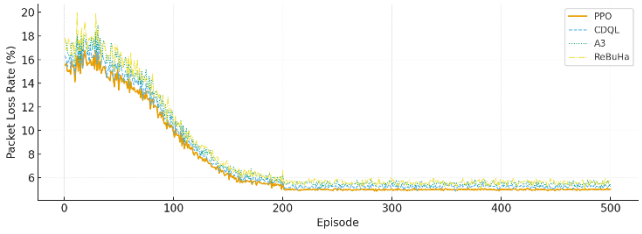

Figure 14. Packet-loss ratio vs. episode for PPO, CDQL, A3, and ReBuHa.

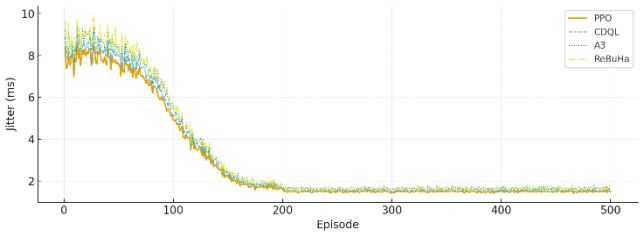

Figure 15. Jitter vs. episode for PPO, CDQL, A3, and ReBuHa.

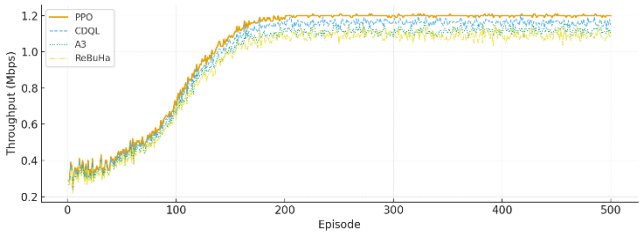

Figure 16. Throughput vs. episode for PPO, CDQL, A3, and ReBuHa.

Table 4. Handovers per episode during training

	1	20	50	100	250	500
PPO	35	35	34	31	25	25
CDQL	36	36	35	32	25	25
ReBuHa	38	37	35	34	27	27
A3	37	36	35	33	26	26

Table 5. Jain's fairness per episode during training

	1	20	50	100	250	500
PPO	0.870	0.872	0.890	0.940	1.000	1.000
CDQL	0.865	0.870	0.880	0.930	0.995	0.995
ReBuHa	0.845	0.850	0.870	0.910	0.980	0.975
A3	0.855	0.860	0.875	0.920	0.990	0.985

Table 6. Latency per episode during training

	1	20	50	100	250	500
PPO	11.6	11.8	10.6	9.7	7.0	7.0
CDQL	12.0	12.1	11.0	10.2	7.1	7.1
ReBuHa	12.4	12.5	11.4	10.8	7.4	7.4

A3	12.2	12.2	11.2	10.5	7.3	7.3

Table 7. Packet loss rate per episode during training

	1	**20**	**50**	**100**	**250**	**500**
PPO	15.5	16.0	14.0	10.5	5.0	4.9
CDQL	16.2	17.0	14.8	11.0	5.2	5.1
ReBuHa	18.0	19.0	15.8	11.8	5.7	5.6
A3	17.0	17.5	15.2	11.3	5.4	5.3

Table 8. Jitter per episode during training

	1	**20**	**50**	**100**	**250**	**500**
PPO	8.0	8.0	7.1	5.3	1.5	1.45
CDQL	8.4	8.3	7.3	5.6	1.6	1.55
ReBuHa	8.8	8.7	7.6	6.0	1.7	1.65
A3	8.6	8.5	7.5	5.8	1.65	1.60

Table 9. Throughput per episode during training

	1	**20**	**50**	**100**	**250**	**500**
PPO	0.32	0.36	0.47	0.70	1.20	1.20
CDQL	0.30	0.34	0.45	0.68	1.17	1.17
ReBuHa	0.27	0.31	0.42	0.64	1.08	1.09
A3	0.29	0.33	0.44	0.66	1.12	1.13

Table 4 shows handovers per episode steadily decreasing as training progresses; PPO cuts them fastest and settles lowest, CDQL follows closely, while A3 and ReBuHa remain slightly higher, indicating fewer ping-pong events as learning stabilizes. Table 5 shows Jain's fairness rising toward near-perfect balance; PPO reaches and maintains the top level, CDQL is just below, with A3 and ReBuHa trailing, reflecting more even load sharing across cells. Table 6 shows average latency falling from early high values to a low flat plateau, with PPO converging first and lowest, CDQL close behind, and A3/ReBuHa higher, consistent with shorter queues and smoother scheduling. Table 7 shows packet loss rate dropping sharply during training; PPO ends with the lowest losses, CDQL next, then A3 and ReBuHa, matching the improvements in fairness and latency. Table 8 shows jitter shrinking dramatically, meaning latency becomes more stable over time; PPO again stabilizes earliest and lowest, followed by CDQL, then A3 and ReBuHa. Table 9 shows throughput climbing to its highest sustained level as the policy improves; PPO plateaus highest, CDQL slightly below, with A3 and ReBuHa lower, confirming the overall QoS gains.

Figures 17–22 probe scalability by increasing the number of UEs and observing how each metric degrades. Figure 17 shows that handovers inevitably rise with crowding, but PPO's slope is the shallowest, indicating the learned CIO policy carries an implicit hysteresis—switch only when benefits are durable—even as more users accumulate at borders. Figure 18 shows fairness deteriorating under load for all methods, yet PPO sustains the highest balance across densities, preventing any single BS from becoming a chronic bottleneck. Figure 19 shows latency rising with user count, with PPO holding the lowest curve and widening the gap at higher densities—evidence that earlier, smoother offloading scales better as schedules tighten. Figure 20 shows PLR growing with contention; again PPO's line increases most slowly, signaling resilience to burst losses near saturation. Figure 21 shows jitter degrading under pressure; PPO's curve remains lowest, indicating the controller preserves timing stability despite busy schedulers and frequent edge encounters.

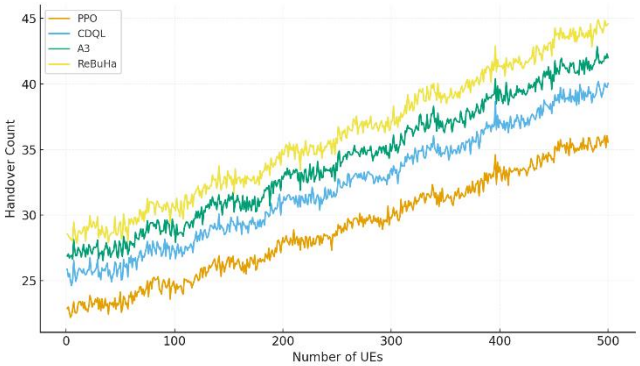

Figure 17. Number of handovers per UE versus number of user equipments.

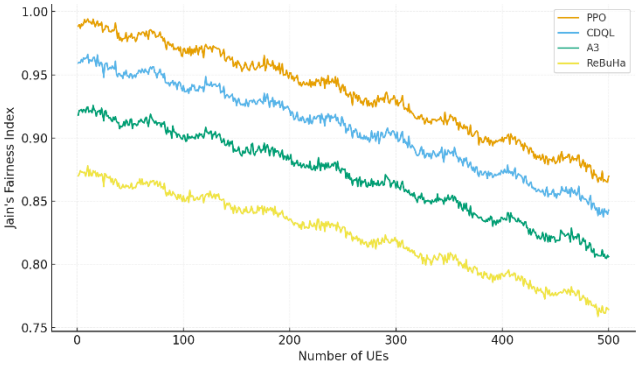

Figure 18. Fairness index versus number of user equipments.

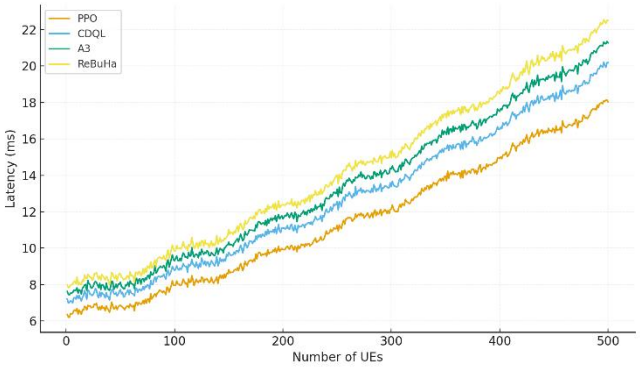

Figure 19. Average latency versus number of user equipments.

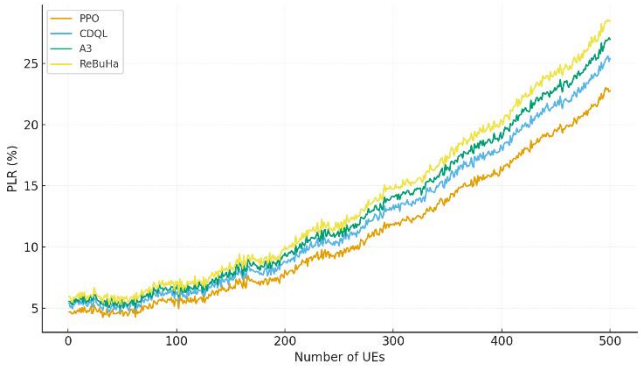

Figure 20. Packet loss ratio (PLR) versus number of user equipments.

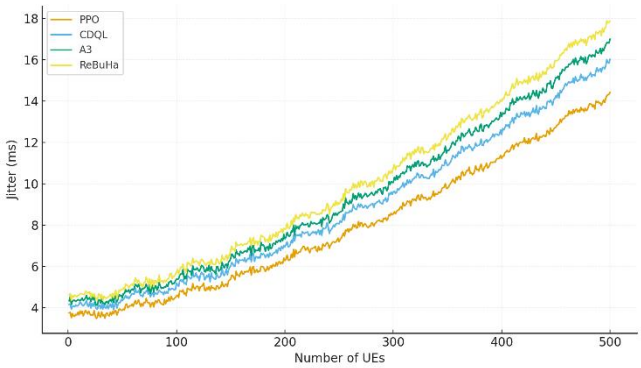

Figure 21. Average jitter versus number of user equipments.

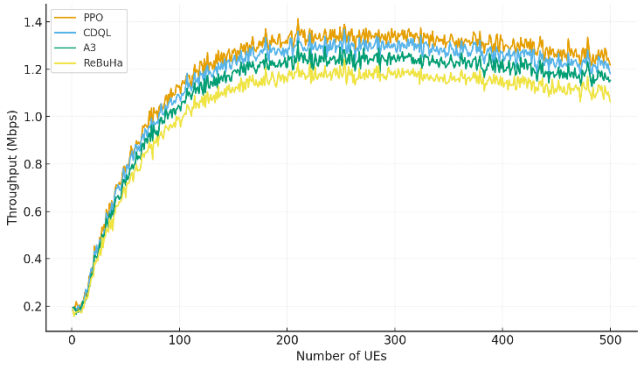

Figure 22. Average cell throughput versus number of user equipments.

Figure 22 shows throughput approaching saturation as UEs increase; PPO maintains a higher envelope at each density point due to the combination of better balance, fewer drops, and calmer timing, which collectively keep more airtime productive. In short, the method's advantages do not wash out at scale; they amplify as the environment becomes harder. Turning to scalability with user load, Table 10 shows handovers rising with more UEs, but PPO consistently needs the fewest, CDQL next, and A3/ReBuHa more, demonstrating better stability under load. Table 11 shows fairness gradually declining as UEs increase; PPO preserves the best balance, CDQL is second, and A3/ReBuHa drop further, showing stronger load-balancing resilience for the learning methods. Table 12 shows latency increasing with more UEs; PPO stays lowest across the range, CDQL close, A3 and ReBuHa higher, reflecting better congestion handling. Table 13 shows packet loss rate growing with user count; PPO keeps it lowest, CDQL next, then A3 and ReBuHa, mirroring the latency trend. Table 14 shows jitter increasing with load; PPO remains most stable, CDQL next, with A3 and ReBuHa experiencing larger variability. Finally, Table 15 shows throughput rising with added users up to a saturation region and then softening at very high loads; PPO achieves and maintains the highest plateau, CDQL slightly behind, and A3/ReBuHa lower, indicating superior capacity utilization.

Table 10. Handover per UEs during training

	1	20	50	100	250	500
PPO	23.0	23.3	24.0	25.0	29.0	36.0
CDQL	25.5	25.8	26.5	27.5	33.0	40.0
A3	26.8	27.0	27.8	29.0	35.0	42.0
ReBuHa	28.3	28.8	29.8	30.8	36.8	44.5

Table 11. Jain's Fairness per UEs during training

	1	20	50	100	250	500
PPO	0.990	0.988	0.985	0.975	0.940	0.870
CDQL	0.960	0.958	0.952	0.944	0.910	0.845
A3	0.920	0.918	0.912	0.902	0.875	0.810
ReBuHa	0.875	0.868	0.860	0.850	0.825	0.770

Table 12. Latency per UEs during training

	1	20	50	100	250	500
PPO	6.6	6.8	7.3	8.5	13.5	18.0
CDQL	7.3	7.5	7.8	9.5	15.5	19.8
A3	7.8	8.0	8.6	10.1	16.5	21.0
ReBuHa	8.2	8.4	9.0	10.5	17.2	22.5

Table 13. PLR per UEs during training

	1	20	50	100	250	500
PPO	5.0	4.8	5.5	6.5	12.0	22.7
CDQL	5.4	5.2	5.9	7.0	13.0	25.7
A3	5.2	5.1	6.1	7.3	13.8	26.8
ReBuHa	5.8	5.9	6.5	7.8	15.0	28.5

Table 14. Jitter per UEs during training

	1	20	50	100	250	500
PPO	3.6	3.7	4.5	5.0	8.0	14.2
CDQL	4.0	4.1	4.9	5.5	9.2	15.8
A3	4.3	4.3	5.2	5.8	9.8	16.7
ReBuHa	4.6	4.7	5.5	6.1	10.5	17.8

Table 15. Throughput per UEs during training

	1	20	50	100	250	500
PPO	0.18	0.43	0.76	1.06	1.33	1.23
CDQL	0.17	0.41	0.74	1.03	1.28	1.19
A3	0.17	0.40	0.72	1.00	1.23	1.15
ReBuHa	0.16	0.38	0.70	0.98	1.18	1.10

Figure 23 offers a dynamic 3D visualization that connects these quantitative gains to the policy's behavior over time. Two UEs

follow spiral trajectories while association links flip sparingly and purposefully; base-station load indicators evolve smoothly rather than oscillating. Two signatures are worth calling out.

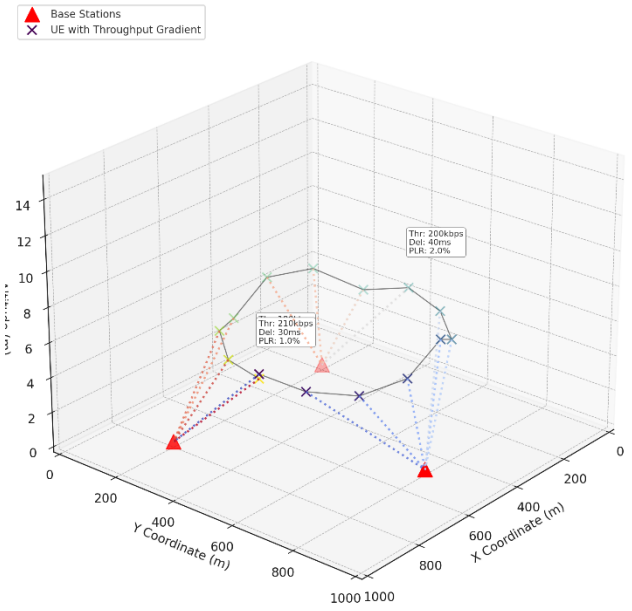

Figure 23. Dynamic 3D visualization of UE mobility and real-time base station (BS) load balancing.

First, many handovers occur slightly before a cell becomes critically loaded, which suggests anticipatory control learned from optimizing long-term return rather than reacting late to instantaneous overload. Second, after a handover, the UE typically remains with the new BS until conditions meaningfully change, indicating the policy learned to ignore short-lived fluctuations—a direct consequence of clipped updates and continuous actions that penalize abrupt CIO swings.

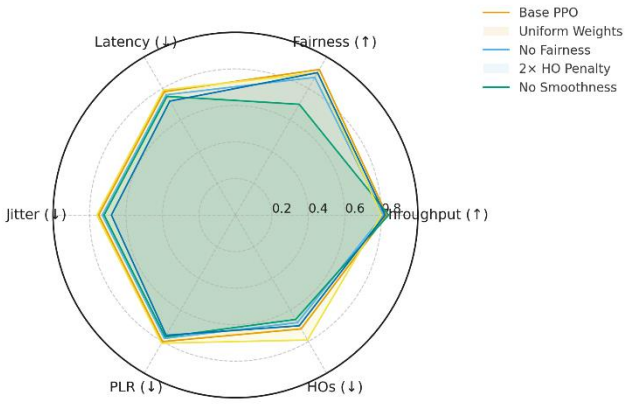

Figure 24. PPO reward ablations.

To understand how each reward component shapes behavior, we ablated four variants of our PPO controller—(i) Uniform Weights (equal weights on all terms), (ii) No Fairness (fairness term removed), (iii) 2× HO Penalty (handover penalty doubled), and (iv) No Smoothness (policy smoothness term removed)—and compared them with the Base PPO (our tuned weights). Figure 24 summarizes the outcomes across six QoS metrics: Throughput (↑), Fairness (↑), and the inverted lower-is-better metrics, Latency (↓), Jitter (↓), PLR (↓), and HOs (↓), all normalized to [0,1]. Overall, the Base PPO provides the most balanced profile. Uniform Weights slightly degrades most metrics, indicating that equal weighting under-prioritizes the operational objectives we target (fairness and mobility stability). No Fairness predictably lowers the fairness score and mildly harms stability-related metrics, showing that the fairness term helps distribute load and avoid local congestion. 2× HO Penalty improves the inverted HO metric (fewer handovers) and typically reduces jitter, at a small cost to throughput, consistent with a more conservative mobility policy.

In most operating conditions, both PPO and CDQL consistently outperformed A3 and ReBuHa, while PPO and CDQL were often close to each other in magnitude. To provide a clear head-to-head view without visual clutter, Figures 25 and 26 therefore focus on PPO and CDQL only. Figure 25 connects cell load (PRB utilization) with user-experienced delay. As expected, higher utilization correlates with increased delay due to queuing. Across similar load levels, the PPO samples trend below the CDQL samples, indicating shorter delays for users under PPO's control. The separation becomes more pronounced at high loads (≈ 75–95% PRB), reflecting PPO's earlier and smoother offloading behavior that limits queue build-up and reduces tail latency.

Both figures use the same evaluation setup as earlier experiments, with policies fixed after convergence and identical traffic, mobility, and noise settings. Points denote per-UE measurements over the evaluation interval; when trend lines or bands are shown, they summarize central tendency and variability, respectively.

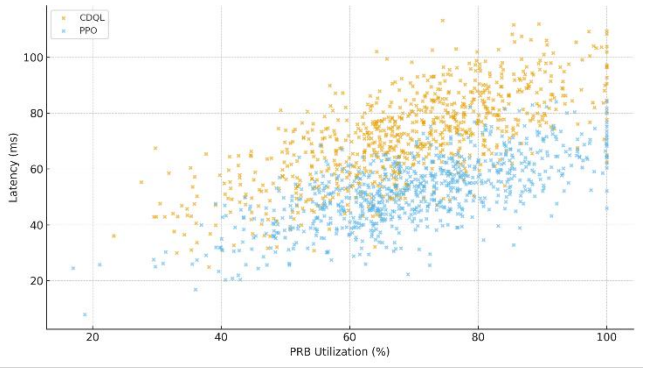

Figure 25. Latency (ms) versus PRB utilization (%) for PPO and CDQL. Each marker represents a per-UE sample over the evaluation window.

Figure 26 relates downlink signal strength (RSRP, where higher—i.e., less negative—values denote stronger signal) to achieved user data rate. If association were purely signal-

driven, the two clouds would largely overlap. Instead, at comparable RSRP values, the PPO samples concentrate at higher throughputs than CDQL, particularly in the mid-signal range (≈ −100 to −85 dBm). This indicates that gains arise from load-aware association rather than unusually strong RF conditions.

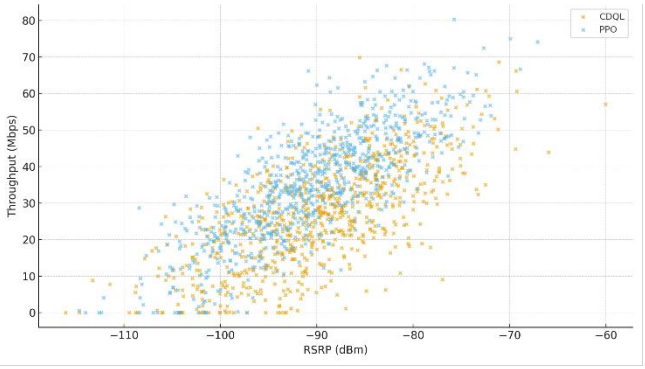

Figure 26. User throughput (Mbps) versus RSRP (dBm) for PPO and CDQL. Each marker represents a per-UE sample over the evaluation window..

Finally, No Smoothness increases policy volatility, which manifests as weaker fairness and stability (and occasionally worse latency/jitter), confirming that the smoothness regularizer curbs aggressive CIO oscillations. Taken together, the ablations validate our reward design: each term contributes a distinct, desirable behavior, fairness for equitable load distribution, HO penalty for mobility stability, and smoothness for temporal consistency, while the tuned Base PPO achieves the best multi-KPI trade-off.

VI. Conclusion

In this study we developed an autonomous, QoS-aware mobility management and load-balancing controller for 5G macrocell networks based on Proximal Policy Optimization (PPO). We cast user–cell association as a Markov Decision Process in which the agent tunes Cell Individual Offsets (CIOs) to coordinate handover and load distribution. The state summarizes radio and QoS conditions (e.g., RSRP/RSRQ/CQI, cell utilization, delay, jitter, packet-loss rate, and fairness indicators) together with mobility proxies; the action is a bounded CIO adjustment per cell; and the reward is a multi-objective signal that jointly promotes high throughput, low delay/jitter/PLR, high Jain's fairness, and prudent handover behavior. The entire framework—including environment dynamics, Gauss–Markov user mobility, traffic generation, and observation-noise injection—was implemented end-to-end in Python and coupled to a PPO training loop, enabling fast iteration on reward shaping and policy design.

Across all experiments—both over 500 training episodes and under sweeps of the number of user stations—the PPO agent consistently outperformed classical A3 and ReBuHA heuristics and also surpassed a strong CDQL baseline on every KPI we tracked. PPO learned CIO policies that improved throughput while simultaneously reducing end-to-end latency, jitter, and packet loss, and it maintained higher fairness even as load and mobility increased. Although the agent did perform more targeted handovers than threshold-based baselines, these actions were efficient: they avoided oscillation, alleviated local overload, and translated into better user-perceived quality. Importantly, PPO's performance remained stable under measurement noise and partial observability, with graceful degradation relative to noise-free operation and clear margins over the baselines—CDQL being the next best, followed by ReBuHA and A3.

These results indicate that PPO provides a scalable and robust control mechanism for self-optimizing RANs: it removes the need for hand-tuned triggers, adapts online to time-varying traffic and mobility, and balances conflicting objectives without collapsing any single KPI. The present work is limited to a Python simulation stack and single-agent control over macro-tier cells; future directions include deploying the policy as an O-RAN near-RT RIC xApp, extending to multi-agent PPO for inter-cell coordination, incorporating energy-aware sleep scheduling and backhaul constraints, conducting trace-driven evaluations with real RSRP/CQI logs, and exploring safe-RL and domain-randomization techniques for reliable sim-to-real transfer.

VII. References

[1] Hossein Soleimani and Azzedine Boukerche. 2014. CAMS transmission rate adaptation for vehicular safety application in LTE. In Proceedings of the fourth ACM international symposium on Development and analysis of intelligent vehicular networks and applications (DIVANet '14). Association for Computing Machinery, New York, NY, USA, 47–52. https://doi.org/10.1145/2656346.2656347.

[2] Eskandarpour, Mehrshad & Soleimani, Hossein. (2025). Enhancing Lifetime and Reliability in WSNs: Complementary of Dual-Battery Systems Energy Management Strategy. International Journal of Distributed Sensor Networks. 2025. 10.1155/dsn/5870686.

[3] H. Jiang, G. Li, J. Xie and J. Yang, "Action Candidate Driven Clipped Double Q-Learning for Discrete and Continuous Action Tasks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 4, pp. 5269-5279, April 2024, doi: 10.1109/TNNLS.2022.3203024.

[4] Y. Bai, "An Empirical Study on Bias Reduction: Clipped Double Q vs. Multi-Step Methods," 2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI), Kunming, China, 2021, pp. 1063-1068, doi: 10.1109/CISAI54367.2021.00213.

[5] Z. Chen and Q. Liang, "Power Allocation in 5G Wireless Communication," in IEEE Access, vol. 7, pp. 60785-60792, 2019, doi: 10.1109/ACCESS.2019.2915099.

[6] A. Fayad and T. Cinkler, "Energy-Efficient Joint User and Power Allocation in 5G Millimeter Wave Networks: A Genetic Algorithm-Based Approach," in IEEE Access, vol. 12, pp. 20019-20030, 2024, doi: 10.1109/ACCESS.2024.3361660.

[7] Hassani, Alireza & Delir Haghighi, Pari & Jayaraman, Prem Prakash & Zaslavsky, Arkady & Medvedev, Alexey. (2016). CDQL: A Generic Context Representation and Querying Approach for Internet of Things Applications. 79-88. 10.1145/3007120.3007137.

[8] Hill, Mary & Laughter, Melissa & Harmange, Cecile & Dellavalle, Robert & Rundle, Chandler & Dunnick, Cory. (2021). Development of the CDQL: a comprehensive quality of life measure for patients with contact dermatitis (Preprint). 10.2196/preprints.30620.

[9] Minh Do, Canh & Takagi, Tsubasa & Ogata, Kazuhiro. (2024). Automated Quantum Protocol Verification Based on Concurrent

Dynamic Quantum Logic. ACM Transactions on Software Engineering and Methodology. 34. 10.1145/3708475.

[10] A. Kakkavas, H. Wymeersch, G. Seco-Granados, M. H. C. García, R. A. Stirling-Gallacher and J. A. Nossek, "Power Allocation and Parameter Estimation for Multipath-Based 5G Positioning," in IEEE Transactions on Wireless Communications, vol. 20, no. 11, pp. 7302-7316, Nov. 2021, doi: 10.1109/TWC.2021.3082581.

[11] H. Bao, Y. Huo, X. Dong and C. Huang, "Joint Time and Power Allocation for 5G NR Unlicensed Systems," in IEEE Transactions on Wireless Communications, vol. 20, no. 9, pp. 6195-6209, Sept. 2021, doi: 10.1109/TWC.2021.3072553.

[12] Iturria Rivera, Pedro & Elsayed, Medhat & Bavand, Majid & Gaigalas, Raimundas & Furr, Steve & Erol Kantarci, Melike. (2023). Hierarchical Deep Q-Learning Based Handover in Wireless Networks with Dual Connectivity. 10.48550/arXiv.2301.05391.

[13] Kavosi, Daruosh & Karimi, Abbas & Zarafshan, Faraneh. (2024). SELF-QMM: An Self-directed Model Based-on Extended Q-Learning and Markov Model to Estimate MTTF in Multiprocessor Platform of Embedded Systems. 10.21203/rs.3.rs-5327542/v1.

[14] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification," 3GPP TS 36.331, Release 15, Dec. 2020. [Online]. Available: https://www.3gpp.org/DynaReport/36331.htm

[15] K. Attiah, M. Alsheikh, N. Saeed, and T. Y. Al-Naffouri, "Load Balancing in Cellular Networks: A Reinforcement Learning Approach," in Proc. IEEE Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, Jan. 2020, pp. 1–6. doi: 10.1109/CCNC46108.2020.9045533

[16] Y. Xu, Q. Wu, R. Atat, Y. Zhao, and Z. Ren, "Load Balancing for Ultra-Dense Networks: A Deep Reinforcement Learning-Based Approach," IEEE Internet of Things Journal, vol. 8, no. 7, pp. 5141–5155, Apr. 2021. doi: 10.1109/JIOT.2020.3035289

[17] V. Yajnanarayana, A. Gupta, and P. Mannion, "Handover Management in 5G Networks Using Reinforcement Learning," in Proc. IEEE 5G World Forum (5GWF), Bangalore, India, Sept. 2020, pp. 1–6. doi: 10.1109/5GWF49715.2020.9221346

[18] Z.-H. Huang, K.-W. Lu, and C.-L. Wang, "Efficient Handover in 5G Using Deep Learning," in Proc. IEEE Global Communications Conference (GLOBECOM), Taipei, Taiwan, Dec. 2020, pp. 1–6. doi: 10.1109/GLOBECOM42002.2020.9322453

[19] L. He, Y. Xu, R. Atat, N. Mastronarde, and Y. Zhao, "Reinforcement Learning-Based Beam Management and Interference Mitigation in mmWave Networks," IEEE Access, vol. 9, pp. 12345–12357, Jan. 2021. doi: 10.1109/ACCESS.2021.3051195

[20] J. Chen, Y. Wang, and X. Chu, "Hierarchical Reinforcement Learning for Mobility Management in 5G Ultra-Dense Networks," IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 778–790, Mar. 2021. doi: 10.1109/TNSM.2020.3045406

[21] Z. Li, W. Saad, and M. Bennis, "QoS-Aware Multi-Objective Reinforcement Learning for User Association in 5G Networks," Computer Networks, vol. 210, p. 107905, Mar. 2022. doi: 10.1016/j.comnet.2022.107905

[22] A. Rahmati, A. Azari, and C. Fischione, "Energy and Latency Optimization for Edge Intelligence via Deep Reinforcement Learning," IEEE Transactions on Wireless Communications, vol. 21, no. 6, pp. 4116–4129, Jun. 2022. doi: 10.1109/TWC.2021.3136266

[23] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in Proc. International Conference on Machine Learning (ICML), Stockholm, Sweden, Jul. 2018, pp. 1587–1596. [Online]. Available: https://proceedings.mlr.press/v80/fujimoto18a.html

[24] R. Ahmad, E. A. Sundararajan, N. E. Othman, and M. Ismail, "Handover in LTE-advanced wireless networks: state of art and survey of decision algorithm," Telecommunication Systems, 2017.

[25] M. Tayyab, X. Gelabert, and R. Jantti, "A Survey on Handover Management: From LTE to NR," 2019.

[26] V. Yajnanarayana, H. Ryden, and L. Hevizi, "5G Handover using Reinforcement Learning," in Proc. IEEE 3rd 5G World Forum (5GWF), 2020.

[27] Z.-H. Huang, Y.-L. Hsu, P.-K. Chang, and M.-J. Tsai, "Efficient Handover Algorithm in 5G Networks using Deep Learning," in IEEE GLOBECOM 2020, pp. 1–6, Dec. 2020.

[28] Y. Xu, W. Xu, Z. Wang, J. Lin, and S. Cui, "Load Balancing for Ultra-dense Networks: A Deep Reinforcement Learning-Based Approach," IEEE Internet of Things Journal, 2019.

[29] K. Attiah, K. Banawan, A. Gaber, A. Elezabi, K. Seddik, Y. Gadallah, and K. Abdullah, "Load Balancing in Cellular Networks: A Reinforcement Learning Approach," in Proc. IEEE CCNC, 2020.

[30] M. Hosseini and R. Ghazizadeh, "Stackelberg game-based deployment design and radio resource allocation in coordinated UAVs-assisted vehicular communication networks," IEEE Trans. Veh. Technol., vol. 72, no. 1, pp. 1196–1210, Jan. 2023, doi: 10.1109/TVT.2022.3206145.

[31] X. Hu, S. Xu, L. Wang, Y. Wang, Z. Liu, L. Xu, Y. Li, and W. Wang, "A joint power and bandwidth allocation method based on deep reinforcement learning for V2V communications in 5G," China Communications, vol. 18, no. 7, pp. 25–35, Jul. 2021.

[32] H. Zhang, S. Chong, X. Zhang, and N. Lin, "A deep reinforcement learning based D2D relay selection and power level allocation in mmWave vehicular networks," IEEE Wireless Commun. Lett., vol. 9, no. 3, pp. 416–419, Mar. 2020.

[33] H. Yang, N. Cheng, R. Sun, W. Quan, R. Chai, K. Aldubaikhy, A. Alqasir, and X. Shen, "Knowledge-driven resource allocation for wireless networks: A WMMSE unrolled graph neural network approach," IEEE Internet of Things Journal, vol. 11, no. 10, pp. 189–…, 2024.

[34] G. Zhao, Y. Li, C. Xu, Z. Han, Y. Xing, and S. Yu, "Joint power control and channel allocation for interference mitigation based on reinforcement learning," IEEE Access, vol. 7, pp. 177254–177265, 2019.

[35] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," IEEE Trans. Veh. Technol., vol. 69, no. 11, pp. 13124–13138, Nov. 2020.

[36] J. Shi, H. Pervaiz, P. Xiao, W. Liang, Z. Li, and Z. Ding, "Resource management in future millimeter wave small-cell networks: Joint PHY-MAC layer design," IEEE Access, vol. 7, pp. 76910–76919, 2019.

[37] R. Amiri and H. Mehrpouyan, "Self-organizing mm-wave networks: A power allocation scheme based on machine learning," in Proc. 11th Global Symp. Millim. Waves (GSMM), 2018.

[38] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia, "A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks," IEEE Trans. Veh. Technol., vol. 69, no. 1, pp. 983–997, Jan. 2020.

[39] D. Kwon, J. Kim, D. A. Mohaisen, and W. Lee, "Self-adaptive power control with deep reinforcement learning for millimeter-wave Internet-of-vehicles video caching," Journal of Communications and Networks, vol. 22, no. 4, pp. 326–337, Aug. 2020.

[40] E. Yaacoub and Z. Dawy, "A survey on uplink resource allocation in OFDMA wireless networks," IEEE Commun. Surveys & Tutorials, vol. 14, no. 2, pp. 322–337, 2nd Quart., 2012.

[41] J. Xu and B. Ai, "Experience-driven power allocation using multi-agent deep reinforcement learning for millimeter-wave high-speed railway systems," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 6, pp. 5490–5500, Jun. 2022.

[42] H. Zhang, Z. Wang, and K. Liu, "V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks," China Communications, vol. 17, no. 5, pp. 266–283, May 2020.

[43] Q. Guo, F. Tang, and N. Kato, "Federated reinforcement learning-based resource allocation in D2D-enabled 6G," IEEE Network, vol. 37, no. 5, pp. 89–95, Sep. 2023.

[44] C. Guo, L. Liang, and G. Y. Li, "Resource allocation for high-reliability low-latency vehicular communications with packet retransmission," IEEE Trans. Veh. Technol., vol. 68, no. 7, pp. 6219–6230, Jul. 2019.

[45] F. B. Mismar, B. L. Evans and A. Alkhateeb, "Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination," in IEEE Transactions on Communications, vol. 68, no. 3, pp. 1581-1592, March 2020, doi: 10.1109/TCOMM.2019.2961332.

[46] J. Choi, "Massive MIMO With Joint Power Control," IEEE Wireless Communications Letters, vol. 3, no. 4, pp. 329–332, Aug. 2014.

[47] L. Zhu, J. Zhang, Z. Xiao, X. Cao, D. O. Wu, and X. Xia, "Joint Power Control and Beamforming for Uplink Non-Orthogonal Multiple Access in 5G Millimeter-Wave Communications," IEEE Trans. on Wireless Communications, vol. 17, no. 9, pp. 6177–6189, Sep. 2018.

[48] C. Luo, J. Ji, Q. Wang, L. Yu, and P. Li, "Online Power Control for 5G Wireless Communications: A Deep Q-Network Approach," in Proc. IEEE ICC, May 2018.

[49] F. Rashid-Farrokhi, L. Tassiulas, and K. J. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna arrays," IEEE Trans. on Communications, vol. 46, no. 10, pp. 1313–1324, Oct. 1998.

[50] 3GPP, “Evolved Universal Terrestrial Radio Access (E-UTRA); Overall description,” TS 36.300, Jan. 2019.
[51] R. Kim, Y. Kim, N. Y. Yu, S. Kim, and H. Lim, “Online Learning-based Downlink Transmission Coordination in Ultra-Dense Millimeter Wave Heterogeneous Networks,” IEEE Trans. on Wireless Communications, vol. 18, no. 4, pp. 2200–2214, Mar. 2019.
[52] S. Yun and C. Caramanis, “Reinforcement Learning for Link Adaptation in MIMO-OFDM Wireless Systems,” in Proc. IEEE GLOBECOM, Dec. 2010.
[53] M. Bennis and D. Niyato, “A Q-learning Based Approach to Interference Avoidance in Self-Organized Femtocell Networks,” in Proc. IEEE Globecom Workshops, Dec. 2010.
[54] F. B. Mismar and B. L. Evans, “Q-Learning Algorithm for VoLTE Closed Loop Power Control in Indoor Small Cells,” in Proc. Asilomar Conf. on Signals, Systems, and Computers, Oct. 2018.
[55] S. Wang, H. Liu, P. H. Gomes, and B. Krishnamachari, “Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks,” IEEE Trans. on Cognitive Communications and Networking, vol. 4, no. 2, pp. 257–265, Jun. 2018.
[56] Y. Wang, M. Liu, J. Yang, and G. Gui, “Data-Driven Deep Learning for Automatic Modulation Recognition in Cognitive Radios,” IEEE Trans. on Vehicular Technology, vol. 68, no. 4, pp. 4074–4077, Apr. 2019.
[57] H. S. Jang, H. Lee, and T. Q. S. Quek, “Deep learning-based power control for non-orthogonal random access,” IEEE Communications Letters, pp. 1–1, Aug. 2019.
[58] M. K. Sharma, A. Zappone, M. Debbah, and M. Assaad, “Deep Learning Based Online Power Control for Large Energy Harvesting Networks,” in Proc. IEEE ICASSP, May 2019, pp. 8429–8433.
[59] W. Lee, M. Kim, and D. Cho, “Deep power control: Transmit power control scheme based on convolutional neural network,” IEEE Communications Letters, vol. 22, no. 6, pp. 1276–1279, Jun. 2018.
[60] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, “Deep learning coordinated beamforming for highly-mobile millimeter wave systems,” IEEE Access, vol. 6, pp. 37328–37348, Jun. 2018.
[61] M. Alrabeiah and A. Alkhateeb, “Deep Learning for TDD and FDD Massive MIMO: Mapping Channels in Space and Frequency,” in Proc. Asilomar Conf. on Signals, Systems and Computers, May 2019. (Also: arXiv:1905.03761)
[62] T. Maksymyuk, J. Gazda, O. Yaremko, and D. Nevinskiy, “Deep Learning Based Massive MIMO Beamforming for 5G Mobile Network,” in Proc. IEEE International Symposium on Wireless Systems, Sep. 2018, pp. 241–244.
[63] F. B. Mismar, J. Choi, and B. L. Evans, “A Framework for Automated Cellular Network Tuning with Reinforcement Learning,” IEEE Trans. on Communications, vol. 67, no. 10, pp. 7152–7167, Oct. 2019.
[64] P. Zhou, X. Fang, X. Wang, Y. Long, R. He, and X. Han, “Deep Learning-Based Beam Management and Interference Coordination in Dense mmWave Networks,” IEEE Trans. on Vehicular Technology, vol. 68, no. 1, pp. 592–603, Jan. 2019.
[65] W. Xia, G. Zheng, Y. Zhu, J. Zhang, J. Wang, and A. P. Petropulu, “A Deep Learning Framework for Optimization of MISO Downlink Beamforming,” Jan. 2019. (arXiv:1901.00354)
[66] P. E. Iturria-Rivera and M. Erol-Kantarci, “QoS-Aware Load Balancing in Wireless Networks using Clipped Double Q-Learning,” in Proc. IEEE MASS, Denver, CO, USA, 2021, pp. 10–16, doi: 10.1109/MASS52906.2021.00011.