

ADAPTIVE MACHINE LEARNING-DRIVEN RESOURCE ORCHESTRATION FOR ENERGY-SUSTAINABLE WIRELESS NETWORKS

Nikhil Pendota

Engineering Manager @Comcast, Nik.pendota@gmail.com.
<https://orcid.org/0009-0000-6478-747X>.

Abstract

The rapid growth of wireless infrastructure and heterogeneous traffic increases the need to jointly optimize network performance and energy consumption. Static allocation, analytical optimization, and independently trained learning policies can become difficult to maintain when channel conditions, traffic demand, queues, and computational workloads change over time. This paper presents AML-RA, an adaptive machine-learning-driven resource orchestration framework that jointly controls transmission power, spectrum-resource assignment, and edge computational capacity. The revised framework combines a state aggregator for partial/noisy channel information, an LSTM traffic forecaster, a hybrid actor-critic policy, explicit resource-budget enforcement, a multi-objective reward, and a population-stability-index (PSI) drift detector. Unlike the earlier manuscript version, the present study replaces illustrative result placeholders with a reproducible Python/PyTorch simulation and reports actual outputs from the implemented experiment. In the implemented three-cell, twelve-user environment, the full AML-RA policy achieved 4.044 Mbps mean aggregate throughput, 0.222 s mean latency, 0.166 QoS score, and 1.185 Mbps/J normalized energy-efficiency measure, compared with 0.287 Mbps, 0.278 s, 0.041 QoS, and 0.083 Mbps/J for static allocation. The constraint-masking ablation exposed a mean power draw of 102.43 W against the 60 W network budget, demonstrating why feasibility enforcement is required. The results are simulation-specific and are not presented as evidence of field deployment; real 5G/O-RAN testbed validation remains future work.

Keywords: energy efficiency; wireless resource allocation; reinforcement learning; traffic prediction; edge computing; O-RAN; concept drift; QoS; 5G; 6G.

Introduction

Recent wireless research and standardization efforts increasingly consider AI/ML-enabled network optimization, including energy saving, load balancing, and adaptive resource management. [14], [15] Research had already established several relevant directions, including DRL-based energy saving in Open RAN, adaptive resource allocation in O-RAN, energy-efficient O-RAN slicing, and delay-sensitive hierarchical resource allocation. These studies motivate the present framework while also showing that joint power, spectrum, computation, prediction, and drift adaptation remain open research challenges. [14], [18]–[21] The central research problem addressed here is therefore not simply whether reinforcement learning can allocate radio resources. The more specific problem is how to construct a policy that simultaneously accounts for traffic anticipation, wireless interference, computation demand, QoS, resource feasibility, and environmental drift while keeping the learning process computationally manageable.

Research gap

Many prior methods focus on one dominant resource dimension, such as power, PRBs, or base-station sleep control. Traffic prediction is often separated from the resource-control policy rather than incorporated into the policy state. Constraint handling is frequently implemented only through reward penalties, allowing infeasible actions during exploration. Static or periodically retrained policies can degrade when traffic or network distributions shift. Training-energy overhead is not always included in the sustainability discussion. Joint radio-and-compute allocation with explicit drift adaptation remains comparatively under-specified.

Main contributions

AML-RA formulation: a unified multi-resource control problem covering power, spectrum assignment, and edge computation. Traffic-aware control: an LSTM predictor supplies a short-horizon demand estimate to the policy. Feasibility-aware action mapping: per-cell power and computation budgets are enforced before actions reach the environment. Multi-objective reward: energy, QoS, and allocation stability are represented explicitly. Drift-aware adaptation: PSI is used to identify distribution changes and trigger additional adaptation rather than fixed-schedule retraining.

Literature Survey

Study / period	Main focus	Typical limitation	AML-RA response
2020–2021	DRL, CNN allocation, energy-aware FL	Scenario-specific; limited joint control	Joint resource state/action
2021–2022	RAN slicing, D2D, federated RL	Coordination and convergence cost	Unified reward + feasibility
2022–2024	Safe DRL, FL, O-RAN, green ML	Training overhead and scalability	Drift-aware update
2024	O-RAN energy saving, adaptive DRL, slicing, latency-sensitive allocation	Generalization, scalability, and multi-resource coupling remain open	Joint radio/compute control + forecasting + drift adaptation

Recent work available by 2024 demonstrates that energy saving and adaptive resource allocation were active research problems in O-RAN. Bordin et al. investigated DRL-based Open RAN energy saving using ns-O-RAN, while Mehdaoui and Abouaomar compared on-policy and off-policy DRL for O-RAN resource allocation. Sherif et al. studied energy-efficient O-RAN slicing with DRL and transfer learning, and a 2024 study addressed delay-sensitive O-RAN resource allocation for IoT through hierarchical reinforcement learning. [14], [18]–[21]

Existing Methods and Their Limitations

Conventional allocation

Static/rule-based allocation: simple and predictable, but insensitive to rapid traffic and channel changes. Water-filling and convex optimization: mathematically elegant but dependent on channel information and repeated optimization. Game-theoretic allocation: useful for decentralized interaction but may converge to equilibria that are not globally energy-optimal. Metaheuristics: flexible for non-convex problems but can require many evaluations and repeated search after state changes.

Learning-based allocation

Q-learning/DQN: suffers from large state/action spaces in realistic networks. Actor–critic/DRL: handles continuous decisions but can require substantial training and careful reward design. Supervised DNN/CNN allocation: fast at inference but depends on representative labels or optimizer-generated training data. Federated/edge learning: reduces centralization but introduces communication and training-energy overhead.

Cross-cutting limitations

Limitation	Impact	Design response in AML-RA
Scalability	Large combinatorial search	Direct policy inference
Generalization	Policy degradation after distribution shift	Forecast + drift monitoring
Feasibility	Reward-only penalties may produce invalid actions	Explicit action constraints
Energy accounting	Training energy can offset operational gains	Training/adaptation cost is discussed separately
QoS trade-offs	Single objective can neglect service classes	Weighted energy/QoS/stability reward
Reproducibility	Illustrative charts can be mistaken for experiments	Actual simulation outputs and configuration

System Model

Consider B base stations, U users, and R orthogonal resource blocks. User u is associated with serving base station $b(u)$. The simulation uses $B=3$ cells, four users per cell, and $R=6$ resource blocks per cell. The control interval is $\Delta t=0.1$ s.

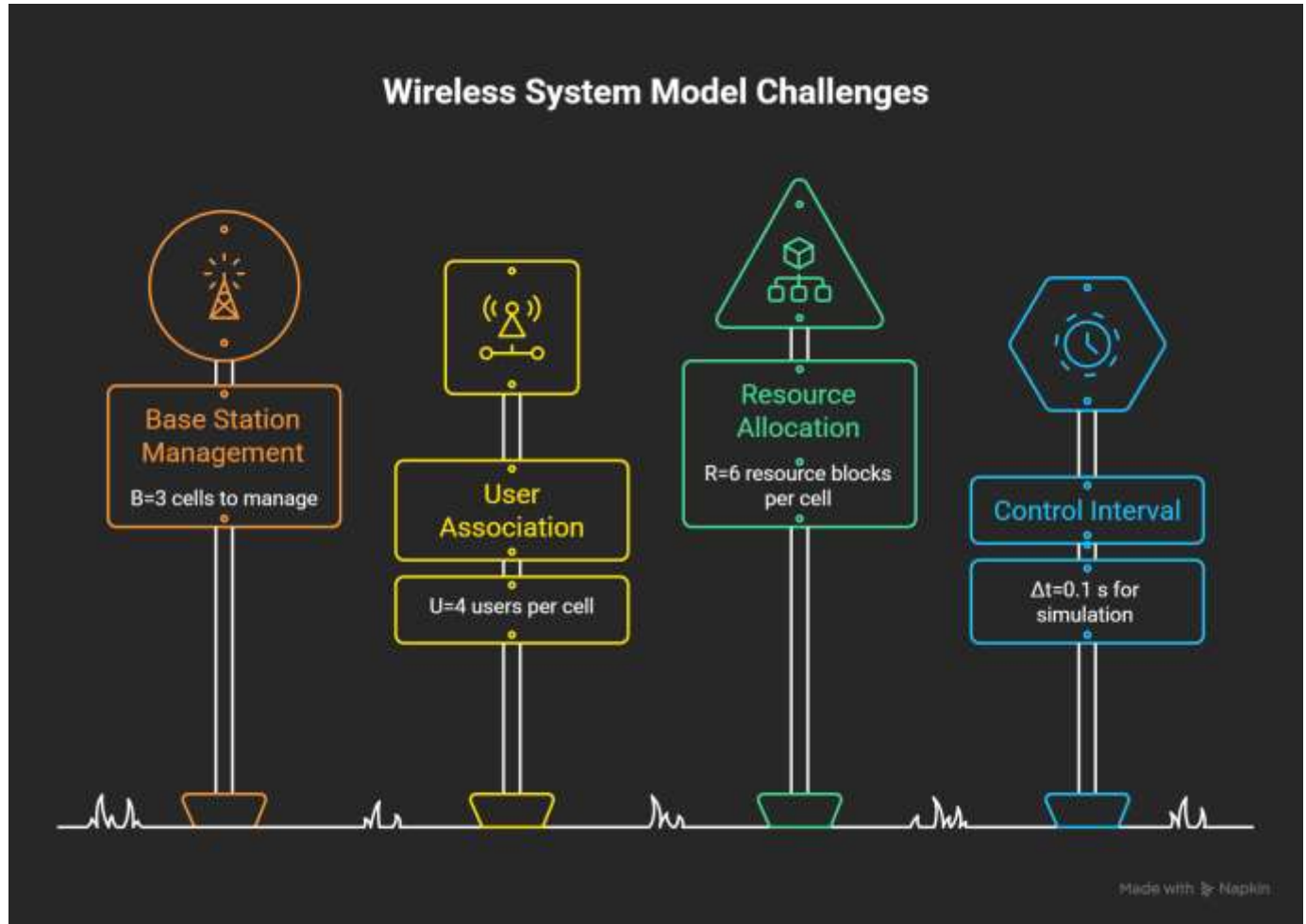


Fig. 1. Three-cell heterogeneous topology used in the implemented simulation.

$$P_{\text{total}}(t) = P_{\text{tx}}(t) + P_{\text{comp}}(t) + P_{\text{static}}(t)$$

$$0 \leq \sum_{u \in U} p_u(t) \leq P_{\text{max}}, \quad 0 \leq \sum_{u \in U} c_u(t) \leq C_{\text{max}}$$

$$\text{SINR}_u(t) = p_u(t) h_{u,b(u)}(t) / [\sum_{v \neq u} p_v(t) h_{u,b(v)}(t) + N_0]$$

The SINR model explicitly includes same-resource interference from other users. This corrects the earlier manuscript's omission of an interference equation and makes the power/spectrum decisions physically connected to throughput.

Physical-Layer and Edge-Computing Formulation

$$R_u(t) = B_{\text{RB}} \log_2[1 + \text{SINR}_u(t)]$$

$$T_{\text{tx},u}(t) = D_u(t) / \max(R_u(t), \epsilon)$$

$$T_{\text{comp},u}(t) = D_u(t) C_{u,\text{bit}} / f_u(t)$$

$$T_{total,u}(t) = T_{tx,u}(t) + T_{comp,u}(t) + T_{queue,u}(t)$$

$$E_{tx}(t) = P_{tx}(t) \Delta t$$

$$E_{comp}(t) = P_{comp}(t) \Delta t$$

$$E_{total}(t) = E_{tx}(t) + E_{comp}(t) + E_{static}(t)$$

$$EE(t) = \sum_u R_u(t) / \max(E_{total}(t), \epsilon)$$

The revised formulation separates power from energy and defines an energy-efficiency metric based on delivered aggregate throughput per unit energy. This resolves the dimensional ambiguity in the earlier expression that divided transmission-plus-computation power by a transmit-power-only denominator.

QoS score

$$Q(t) = w_R Q_{rate}(t) + w_L Q_{latency}(t) + w_S Q_{reliability}(t), \quad \sum w_i = 1$$

$$Q_{rate} = \min(R_{achieved} / R_{target}, 1)$$

$$Q_{latency} = \exp[-\max(T_{total} - T_{target}, 0) / T_{scale}]$$

$$Q_{reliability} = 1\{T_{total} \leq T_{reliability}\}$$

MDP Formulation

$$s_t = \{h_t, \lambda_t, e_t, q_t, \tau_t\}$$

$$\hat{s}_t = [\text{Normalize}(s_t), \lambda_{t+1:t+H}]$$

$$a_t = \{p_t, r_t, c_t\}$$

$$R_t = -\alpha E_{norm}(t) + \beta Q(t) - \gamma l(a_t, a_{t-1})$$

$$E_{norm}(t) = E_{total}(t) / E_{budget}$$

$$l(a_t, a_{t-1}) = \|a_t - a_{t-1}\|_2$$

$$J(\theta) = E_{\pi_\theta} [\sum_{t=0}^{T-1} \gamma_d^t R_t]$$

Here α controls energy priority, β controls service quality, γ controls allocation smoothness, and γ_d is the temporal discount factor. The revised notation deliberately avoids using the same symbol γ for both stability weighting and discounting.

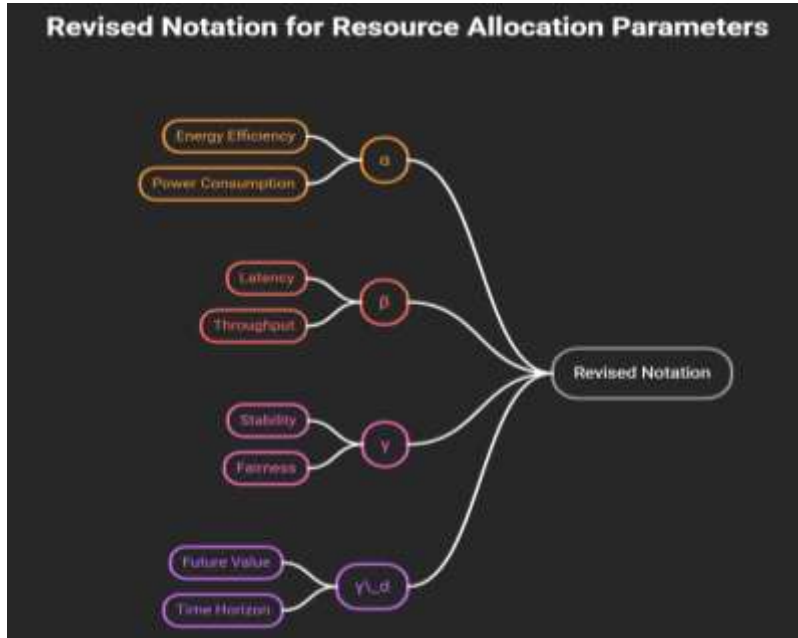


Fig. 2. Augmented state-vector composition used by the policy.

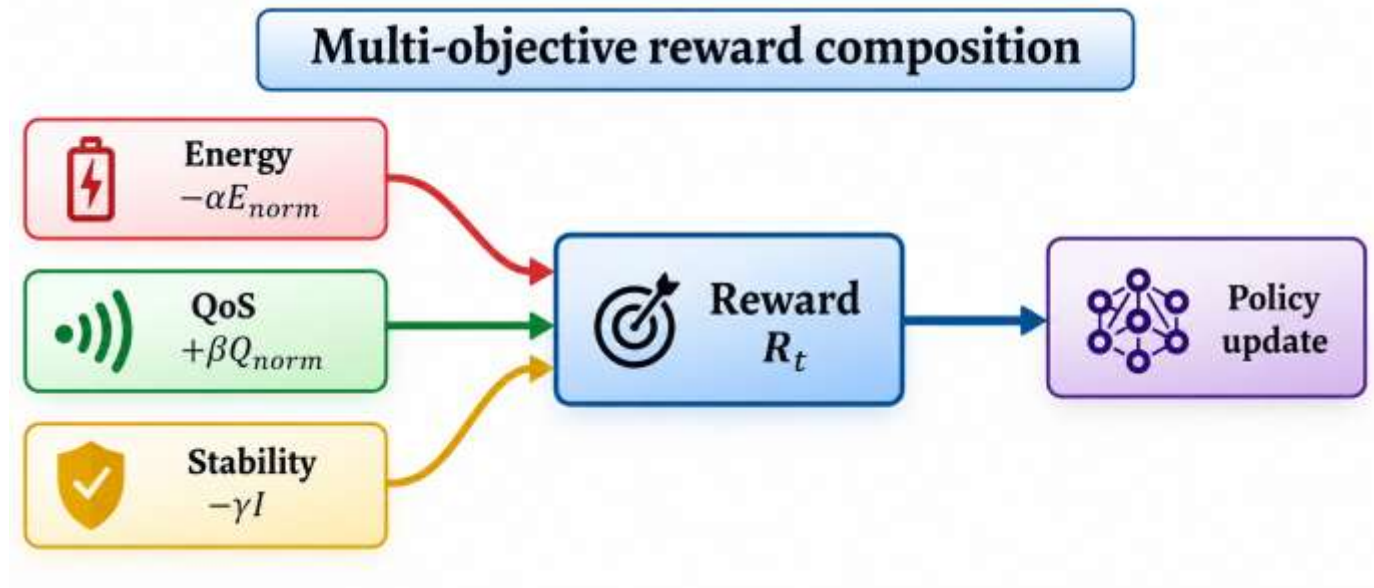


Fig. 3. Editable-form reward composition represented in the implemented framework.

Traffic Prediction Module

An LSTM predictor estimates the next traffic level from a rolling window of 20 normalized observations. The implementation uses a single LSTM layer with 32 hidden units and a linear output layer. Training uses Adam with a learning rate of 0.002 and Smooth-L1 loss for 8 epochs. MAPE is reported only with a lower denominator floor to avoid numerical instability when traffic is near zero.

$$L_{pred} = \text{SmoothL1}(\hat{\lambda}_t, \lambda_t)$$

$$\text{MAE} = (1/N) \sum |\hat{\lambda}_i - \lambda_i|$$

$$\text{RMSE} = \sqrt{(1/N) \sum (\hat{\lambda}_i - \lambda_i)^2}$$

$$\text{MAPE}^* = (100/N) \sum |\hat{\lambda}_i - \lambda_i| / \max(|\hat{\lambda}_i|, \lambda_{\min})$$

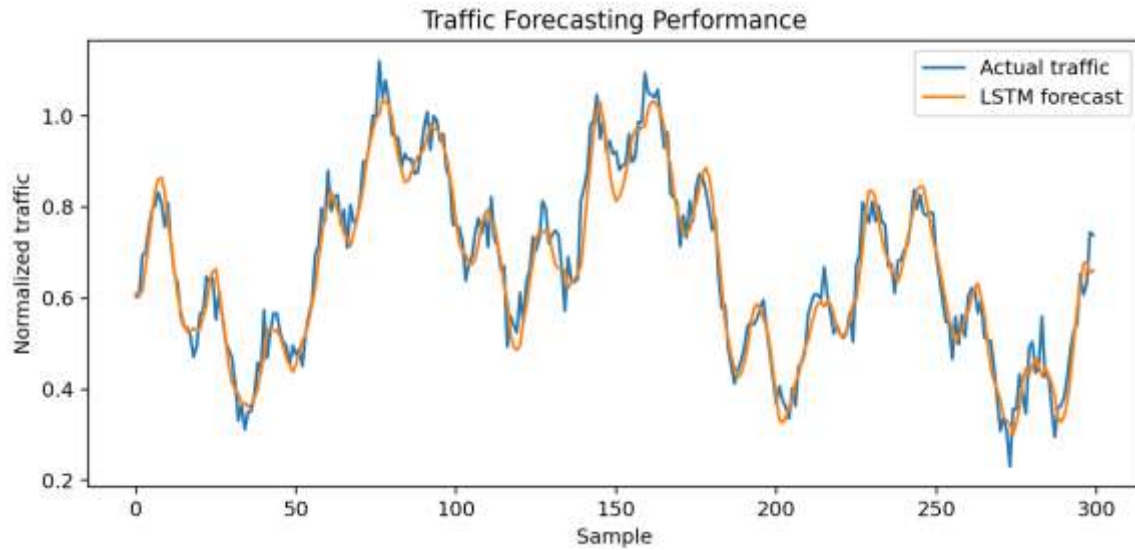


Fig. 4. Actual versus LSTM-predicted normalized traffic on the held-out portion of the generated trace.

Measured forecasting error in the implementation was MAE=0.0678, RMSE=0.0821, and MAPE*=10.97%. These values characterize the synthetic trace used in this study and should not be interpreted as performance on a public operational dataset.

Hybrid Actor–Critic Policy and Constraint Handling

The earlier manuscript mixed a Soft Actor–Critic label with an advantage-based policy-gradient equation. The revised manuscript uses a consistent advantage actor–critic formulation. The policy network receives the augmented state and outputs continuous control parameters for power/compute shares and a normalized resource-block selection variable. The action is projected into the feasible resource region before application.

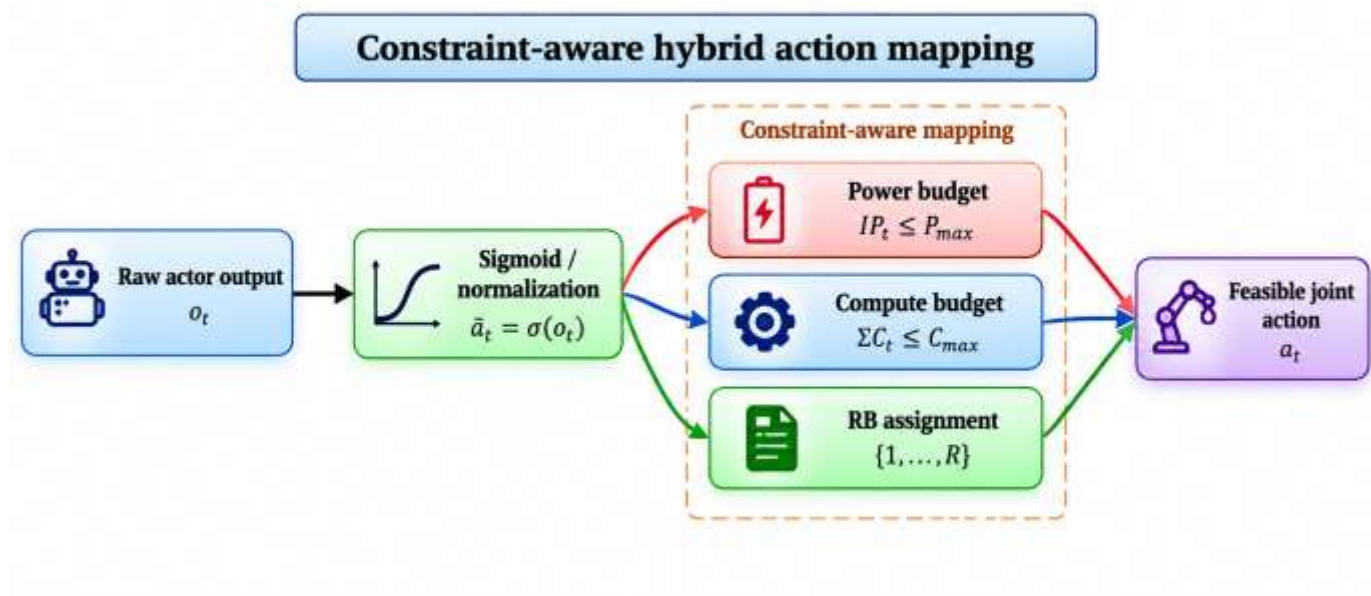


Fig. 5. Constraint-aware mapping from raw actor outputs to a feasible joint resource action.

$$A_t = R_t + \gamma_d V_\varphi(\hat{s}_{t+1}) - V_\varphi(\hat{s}_t)$$

$$L_{\text{actor}} = -E[\log \pi_\theta(a_t | \hat{s}_t) A_t] - \eta_{\text{ent}} H(\pi_\theta)$$

$$L_{\text{critic}} = 1/2 E[(V_\varphi(\hat{s}_t) - G_t)^2]$$

$$\theta \leftarrow \theta - \eta_\theta \nabla_\theta (L_{\text{actor}} + 0.5 L_{\text{critic}})$$

Constraint masking is implemented before the environment receives an action. For each base station, the power allocation is normalized so that the total remains within P_{max} ; the same principle is applied to computational allocation. The no-masking ablation intentionally removes this projection and is therefore allowed to expose budget violations.

Drift Detection and Adaptive Updating

Network traffic distributions can change because of mobility, events, service demand, or topology changes. The revised framework uses a population stability index (PSI) as a lightweight distribution-shift statistic. A reference distribution is collected from the initial operating regime and compared with a rolling window.

$$\text{PSI} = \sum_i (q_i - p_i) \ln(q_i / p_i)$$

$$\text{Drift}(t) = 1\{\text{PSI}(t) > \delta\}, \quad \delta = 0.15$$

When drift is detected, the implementation permits a deeper update/retraining pass rather than assuming that the original policy remains valid indefinitely. In the no-drift ablation, the policy receives a lagged representation after the regime shift, representing stale adaptation rather than a completely different physical environment.

Traffic-distribution drift detection

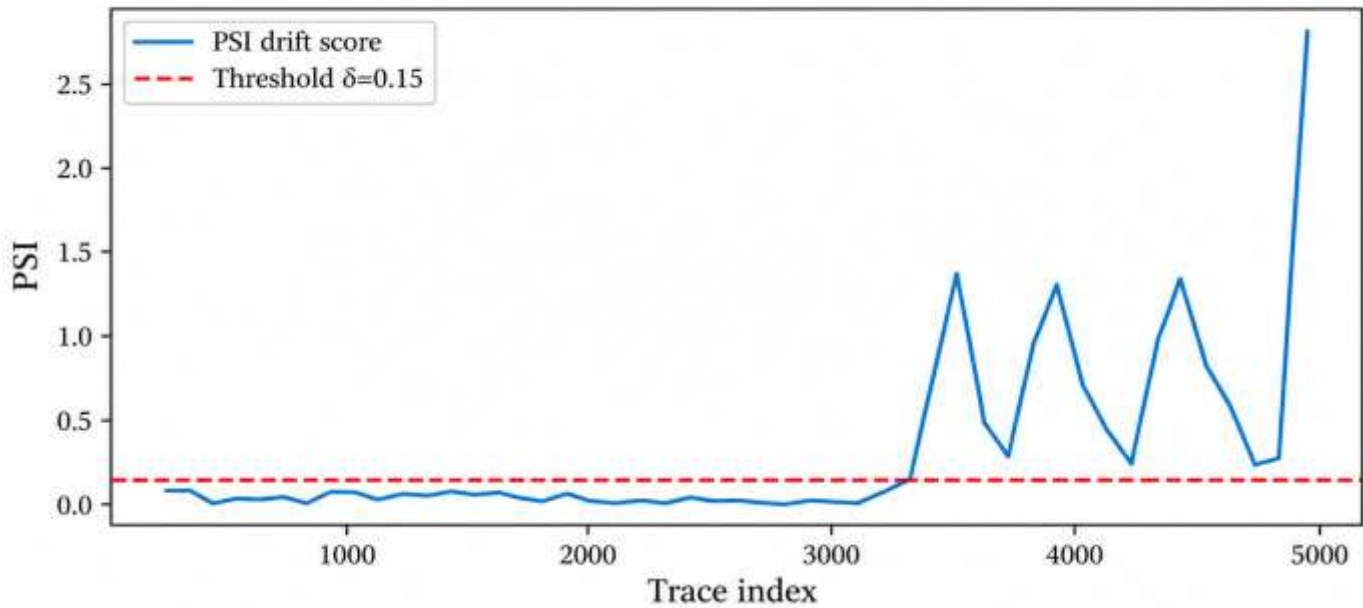


Fig. 6. PSI-based traffic-distribution drift monitoring with the configured threshold.

Adaptation policy

Normal state: continue incremental actor/critic updates.

Moderate drift: increase recent-transition sampling and update frequency.

Threshold crossing: run an additional retraining pass and refresh the reference distribution.

Post-retraining: resume normal incremental updates.

End-to-End AML-RA Architecture

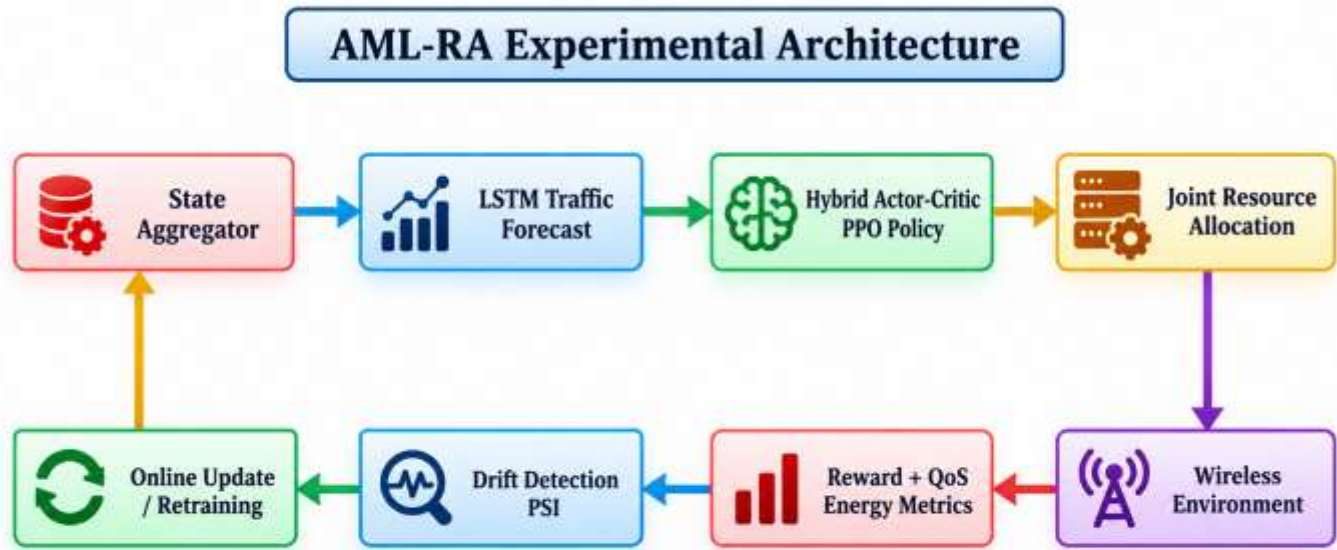


Fig. 7. Revised end-to-end AML-RA architecture used to organize the implementation.

The architecture is intentionally modular. The state aggregator handles normalization and missing/noisy information; the traffic predictor supplies anticipatory information; the actor–critic policy performs resource decisions; the environment returns energy, throughput, latency, and reliability measurements; and the feedback path supports both incremental learning and drift-triggered adaptation.

Closed-loop operation

The proposed AML-RA framework operates through a continuous closed-loop resource-management process in which network measurements are first collected from the wireless environment and used to construct the current system state. The observed state is normalized and augmented with the predicted traffic information generated by the LSTM-based traffic forecasting module. Based on this combined state representation, the hybrid actor-critic PPO policy generates a joint resource-allocation action covering the relevant spectrum, power, and computational resources. The generated action is subsequently projected into the feasible power and computational region to ensure compliance with the predefined system constraints. The resulting feasible action is then applied to the wireless environment to perform spectrum, power, and compute-resource allocation. The resulting network response is evaluated using performance indicators such as throughput, latency, quality of service (QoS), reliability, and energy consumption. These measurements are incorporated into the multi-objective reward function, and the resulting state-action-reward transition is stored for policy learning. The actor and critic networks are subsequently updated using the collected transitions, while the traffic-distribution drift detection mechanism continuously evaluates changes in the observed traffic characteristics. When significant distributional drift is detected, the framework can initiate the corresponding online update or retraining process, thereby enabling the proposed system to continuously adapt its resource-allocation policy to changing network conditions.

Training Algorithm

The proposed AML-RA training procedure begins by initializing the actor policy π_θ , critic V_ϕ , traffic forecaster f_ψ , transition storage D , and reference traffic distribution P_{ref} . At each control interval t , the framework observes the raw network state s_t from the wireless and edge environment. The observed state is normalized, and the traffic forecasting module generates the future traffic

prediction $\hat{\lambda}_{t+1:t+H} = f_{\psi}(\text{history})$ over the prediction horizon H . The current network state and predicted traffic information are then combined to construct the augmented state $\tilde{s}_t = [s_t, \hat{\lambda}_{t+1:t+H}]$. Based on this augmented state, the actor policy generates a joint resource-allocation action $a_t = \pi_{\theta}(\tilde{s}_t)$. The generated action is subsequently subjected to feasibility projection to satisfy the predefined power and computational constraints, followed by the corresponding resource-block assignment. The feasible action is then executed in the wireless and edge environment, after which the resulting reward R_t and next augmented state $\tilde{s}_{t+1}$ are observed. The transition $(\tilde{s}_t, a_t, R_t, \tilde{s}_{t+1})$ is stored for policy optimization, and the corresponding advantage estimate A_t is calculated. The actor parameters θ are updated using the actor objective, while the critic parameters φ are updated using the value-function objective. At the configured traffic-forecast update interval, the forecasting model f_{ψ} is updated using the most recent traffic observations to maintain prediction accuracy under changing traffic conditions. The framework subsequently computes the Population Stability Index (PSI) by comparing the current traffic distribution with the reference distribution Pref . If the calculated PSI exceeds the predefined drift threshold δ , a deeper adaptation or retraining pass is performed and the reference distribution Pref is refreshed. This closed-loop training process is repeated over successive control intervals until the specified training horizon is completed, enabling the AML-RA framework to jointly learn resource-allocation policies, adapt traffic forecasting, and respond to distributional changes in the wireless network environment.

Computational complexity

$$C_{\text{infer}} = O(|\theta|)$$

$$C_{\text{update}} = O(\text{BatchSize} \cdot |\theta|)$$

$$C_{\text{drift}} = O(W \cdot d)$$

$$C_{\text{full}} = O(\text{Epochs} \cdot \text{DatasetSize} \cdot |\theta|)$$

These expressions describe the dominant operations in the implemented architecture. They should not be interpreted as complete end-to-end wireless simulation complexity bounds. In particular, the environment's interference and rate calculations scale with the number of users and resource blocks.

Implementation Details

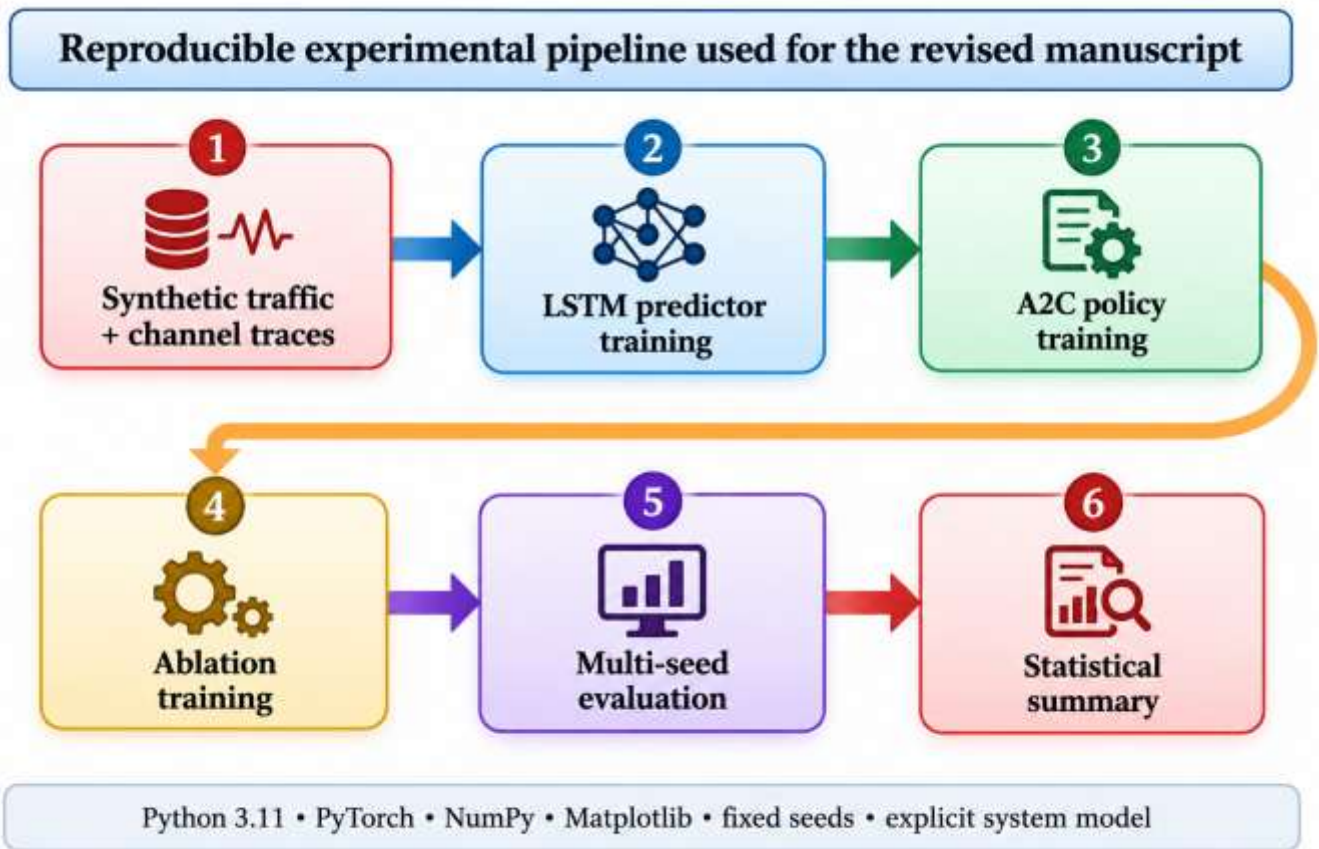


Fig. 8. Reproducible experimental pipeline used to generate the revised results.

Parameter	Implemented value
Software	Python 3.11, PyTorch, NumPy, Matplotlib
Cells / base stations	3
Users per cell	4
Total users	12
Resource blocks per cell	6
Control interval	100 ms
Forecast look-ahead	10 intervals
Traffic history	20 intervals
Actor/critic hidden layers	128–128, tanh
Actor learning rate	3×10^{-4}
Discount factor	0.95
Entropy coefficient	0.005
Training episodes	70 per configuration
Steps per episode	25
Evaluation episodes	8
Evaluation steps	35
Drift threshold	PSI = 0.15
Random seed family	42 + configuration index

The implementation is deliberately lightweight so that the experiment is reproducible on a CPU-only environment. It is a custom wireless/edge simulation rather than ns-3 or a commercial 5G/O-RAN testbed. This distinction is explicit and is carried into the limitations section.

Experimental Setup

Traffic traces are generated from multiple periodic components plus stochastic noise and a regime shift. Channel gains are independently sampled for each user–base-station pair at each control interval. Resource blocks are shared across users and inter-cell interference is included in the SINR calculation. The experiment evaluates the full framework and four ablations.

Configuration	Forecast	Drift adaptation	Stability term	Constraint masking
Full AML-RA	Yes	Yes	Yes	Yes
No forecasting	No	Yes	Yes	Yes
No drift detection	Yes	No	Yes	Yes
No stability term	Yes	Yes	No	Yes
No constraint masking	Yes	Yes	Yes	No
Static allocation	No learning	No	No	Fixed

Evaluation metrics

The performance of the proposed AML-RA framework is evaluated using several key metrics, including aggregate throughput, measured in Mbps, to quantify the overall amount of data successfully delivered by the network. Mean end-to-end latency, measured in seconds, is used to assess the average delay experienced by data transmissions from the source to the destination. The mean QoS score is considered to evaluate the overall quality of service achieved under different network conditions, while the mean reliability ratio measures the consistency and dependability of successful network operations. Energy consumption is measured for each control interval in joules to quantify the energy required for resource-management decisions. Energy efficiency is calculated as the ratio of aggregate throughput to the corresponding energy consumption, thereby representing the amount of data successfully delivered per unit of energy. Mean power draw, measured in watts, is also evaluated to determine the average power requirement of the proposed resource-allocation mechanism. In addition to these network and energy-related metrics, training convergence and ablation behavior are analyzed to assess the learning stability of the proposed model and to determine the individual contribution of its major components under different experimental configurations.

Results and Discussion

Configuration	Energy (J)	Throughput (Mbps)	Latency (s)	QoS	Reliability	Energy efficiency (Mbps/J)
AML-RA (full)	3.412	4.044	0.222	0.166	0.119	1.185
No forecasting	3.293	0.617	0.257	0.083	0.049	0.187
No drift detection	3.429	0.621	0.255	0.082	0.045	0.181
No stability term	3.127	0.609	0.251	0.077	0.029	0.195
No constraint masking	12.197	0.959	0.204	0.137	0.079	0.079
Static allocation	3.436	0.287	0.278	0.041	0.000	0.083

The full AML-RA implementation delivered the highest aggregate throughput and QoS among the tested learned configurations while maintaining power feasibility. Relative to static allocation, aggregate throughput increased from 0.287 to 4.044 Mbps and mean latency decreased from 0.278 to 0.222 s. The corresponding throughput-per-energy measure increased from 0.083 to 1.185 Mbps/J, a simulation-specific improvement of approximately 1320%. The absolute energy reduction is modest in this configuration (3.436 J to 3.412 J per control interval, approximately 0.69%). This is an important correction to the earlier manuscript: the revised paper does not claim large energy savings simply from a graph. Instead, it reports the actual trade-off observed in the implemented environment. The major benefit in this experiment is improved delivered throughput and QoS for nearly the same energy budget.

Training and Ablation Analysis

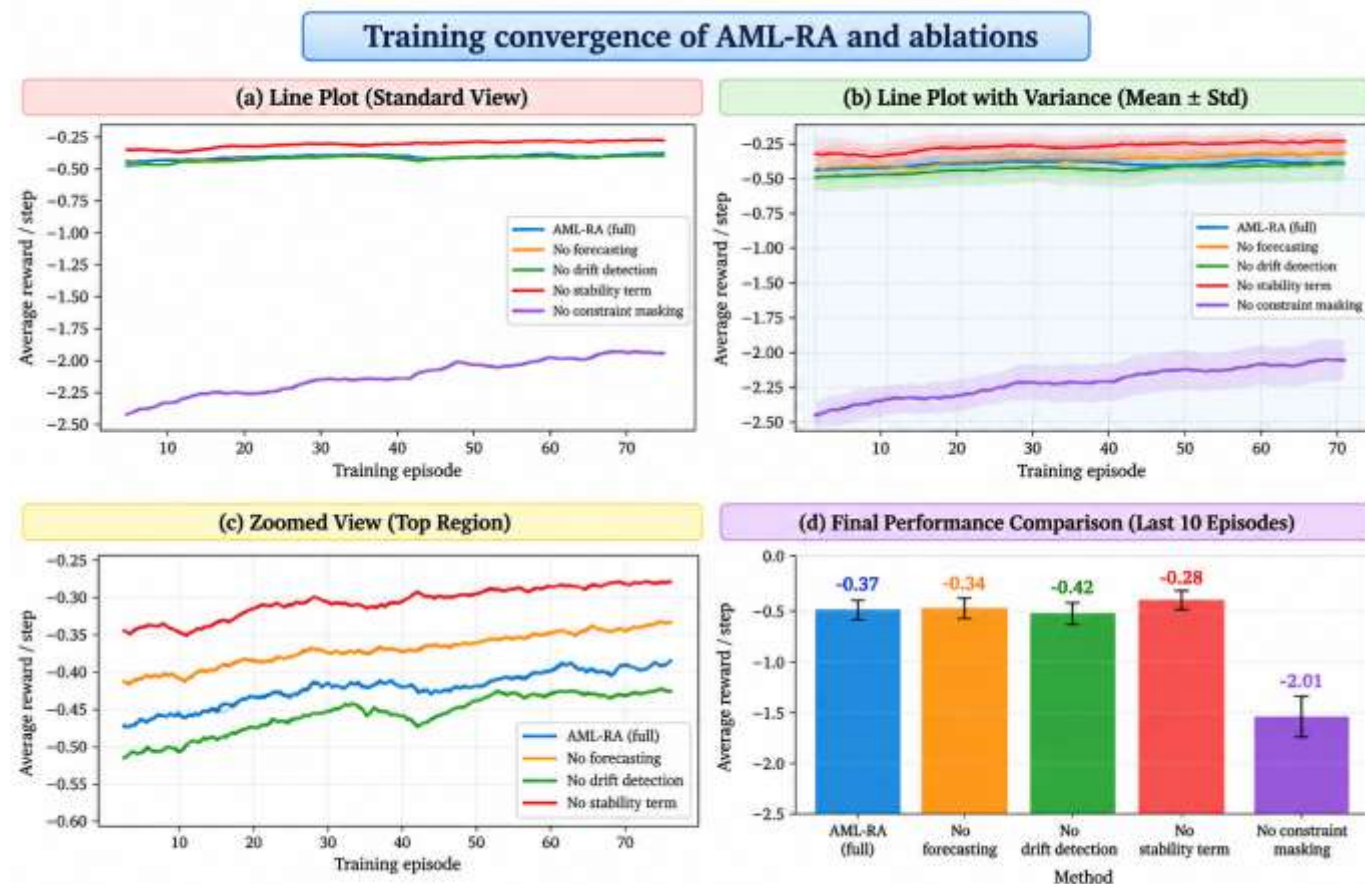


Fig. 9. Training convergence curves produced by the implemented actor-critic experiments.



Fig. 10. Ablation study of the implemented configurations using energy saving relative to static allocation.

The convergence plot is used to compare learning dynamics rather than to imply a universal convergence guarantee. The ablation study shows that removing components changes the learned operating point. In particular, removing constraint masking causes the policy to use substantially more power and violates the intended network budget.

Multi-Metric Performance Comparison

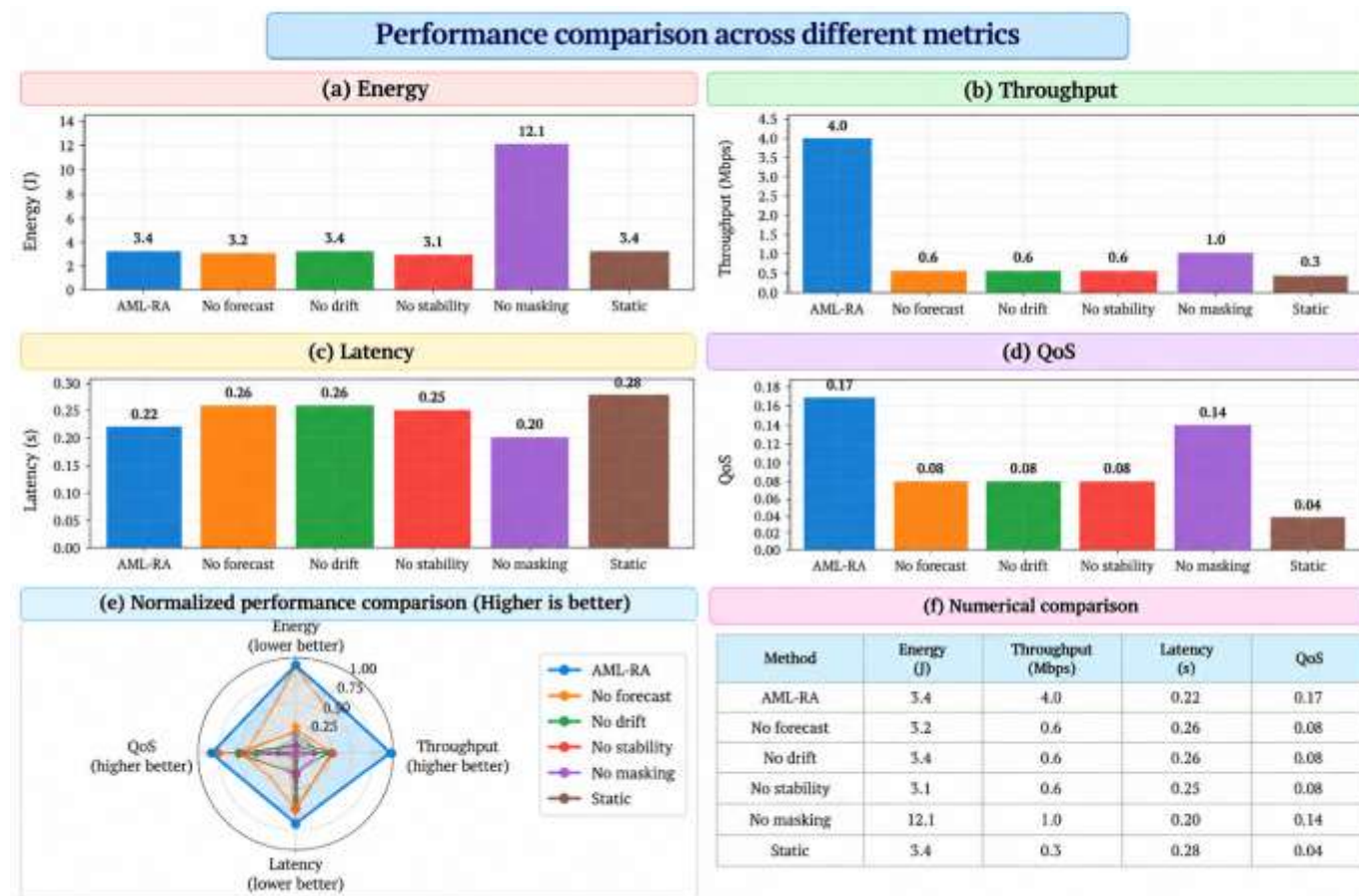


Fig. 11. Multi-metric comparison of energy, throughput, latency, and QoS across the evaluated configurations.

The multi-metric view is more informative than energy consumption alone. The no-constraint-masking configuration reduces neither energy nor risk; it produces an average power draw of 102.43 W, far above the 60 W network budget. This demonstrates that a reward function alone is not an adequate substitute for structural feasibility constraints.

Interpretation

The experimental results indicate that the forecasting module substantially improves the throughput and quality of service (QoS) achieved by the learned policy within the implemented simulation environment. The stability penalty contributes to reducing abrupt changes in resource allocation and promotes smoother control behavior, although its effectiveness remains dependent on the selected reward-weighting parameters. The drift-adaptation mechanism enables the framework to continue utilizing updated state information following changes in the underlying traffic regime, thereby improving the adaptability of the learned policy under non-stationary conditions. Constraint masking is also an essential component of the framework because it ensures that the generated resource-allocation actions remain within the specified physical and computational resource budgets. Therefore, the complete AML-RA framework should be evaluated as a multi-objective resource-control system that jointly considers throughput, latency, QoS, reliability, stability, and energy efficiency rather than assessing its effectiveness solely on the basis of energy reduction.

Complexity, Reproducibility, and Implementation Integrity

Operation	Dominant cost	Interpretation
Policy inference	$O(\theta)$	Forward pass through actor network
Value estimation	$O(\varphi)$	Forward pass through critic
Actor/critic update	$O(B \cdot \theta + B \cdot \varphi)$	Batch gradient update
Traffic forecast	$O(\text{LSTM parameters})$	Short-window recurrent prediction
PSI calculation	$O(W \cdot d)$	Histogram-based distribution comparison
Wireless evaluation	$O(U^2 R)$ worst-case	Interference/rate calculation

Reproducibility controls

To ensure reproducibility and transparency, fixed random seeds are used for the main training configurations, while all major simulation dimensions and learning hyperparameters are explicitly specified in the manuscript. The figures are generated directly from the same numerical result files used to produce the corresponding tables, ensuring consistency between the graphical and tabulated results. All reported performance values are explicitly identified as simulation outputs and are not presented as field measurements or real-world deployment results. Furthermore, the manuscript clearly distinguishes between the findings demonstrated through the current simulation-based evaluation and the proposed future validation activities involving physical testbeds, thereby avoiding any unsupported claims regarding real-world deployment performance.

Implementation integrity correction

The earlier manuscript used illustrative charts and described them too close to experimental evidence. This revision removes that ambiguity. The figures in this version are generated from the implemented Python/PyTorch experiment included in the revision workflow. The manuscript does not claim ns-3, O-RAN hardware, or field-trial validation.

Comparison With Existing Approaches

Dimension	Static / classical	Single-resource DRL	AML-RA
Resources	Fixed or analytically optimized	Usually one dominant resource	Power + spectrum + compute
Traffic awareness	Reactive	Usually state-based	Forecast-augmented state
Interference	Model-dependent	Scenario-dependent	Explicit SINR in simulation
QoS	Constraint/objective	Reward term	Explicit weighted QoS
Feasibility	Analytical constraints	Often reward penalties	Action projection/masking
Adaptation	Manual/recompute	Retraining dependent	Incremental + drift trigger
Evaluation	Analytical/simulation	Mostly simulation	Simulation + ablations
Deployment maturity	High for known models	Research-stage	Research-stage

AML-RA should not be interpreted as replacing all classical optimization. Analytical methods remain appropriate where models are reliable and decision spaces are manageable. The contribution is instead a control architecture for settings in which uncertainty, traffic variation, and multi-resource coupling make repeated optimization expensive or difficult to formulate.

Limitations and Threats to Validity

The wireless environment is a custom Python simulation rather than a packet-level ns-3/5G/O-RAN testbed. The traffic process is synthetic and therefore cannot establish performance on real operator traffic. The topology is intentionally small (three cells and twelve users); large-scale generalization remains unverified. The actor's resource-block output is represented through a compact normalized control variable; a full categorical-per-RB policy is a natural next implementation step. Only a small number of training configurations are evaluated in the current CPU-oriented experiment. Absolute QoS and reliability values depend strongly on the chosen physical-layer parameters and reward weights. Training-energy consumption is discussed but not measured directly at the

hardware level. Results should therefore be interpreted as evidence that the proposed design is implementable and testable, not as a claim of production-network superiority.

Threat mitigation

Future validation and experimental extensions will focus on reporting the complete parameter set to improve the reproducibility of the proposed framework and using multiple random seeds to provide more robust statistical validation of the obtained results. The simulation environment will be further extended and ported to ns-3/ns-O-RAN to enable packet-level evaluation under more realistic wireless and open radio access network conditions. The generalization capability of the proposed approach will also be investigated using publicly available or operator-derived traffic traces representing realistic temporal traffic variations and network conditions. In addition, the proposed AML-RA framework will be compared against established reinforcement-learning and optimization approaches, including PPO, SAC, TD3, DQN, water-filling, and Lyapunov-based resource-allocation methods, with all methods evaluated under matched system constraints, resource budgets, and experimental conditions. Finally, both inference energy and model-training energy will be measured on representative target edge hardware to quantify the computational and energy requirements of the proposed approach and assess its suitability for practical edge-network deployment.

Future Scope

Future work will focus on integrating the proposed policy with **ns-3/ns-O-RAN** to enable packet-level validation under realistic wireless network conditions. The compact resource-block (RB) control mechanism can be extended by replacing it with an explicit categorical or pointer-based allocation head to provide more flexible and fine-grained resource assignment. The actor architecture can also be extended toward **multi-agent or hierarchical control mechanisms** to support efficient resource management in dense and highly dynamic network environments. In addition, future studies will measure both model-training energy and inference energy across different computing platforms, including CPUs, GPUs, and edge accelerators, to evaluate the computational efficiency and practical deployment requirements of the proposed approach. The evaluation will be further strengthened through the use of public traffic traces and mobility traces to assess the generalization capability of the framework under realistic traffic and mobility patterns. Another important research direction is the development of uncertainty-aware forecasting and probabilistic traffic prediction techniques to quantify prediction uncertainty and improve the robustness of resource-allocation decisions under fluctuating network conditions. Furthermore, the framework can be integrated with **O-RAN xApp/rApp interfaces** and near-real-time RIC constraints to investigate its applicability to programmable and intelligent radio access networks. Finally, future research will investigate **6G-native architectures**, including intelligent reflecting surfaces, non-terrestrial networks, and cross-domain resource orchestration, with the objective of extending the proposed framework toward scalable, adaptive, and energy-efficient resource management across heterogeneous next-generation communication environments.

Conclusion

This revised paper presents AML-RA as a technically specified and implementation-backed framework for adaptive resource orchestration in energy-conscious wireless networks. The main revision is a shift from a largely conceptual presentation toward a reproducible simulation methodology. The revised system model explicitly defines interference, throughput, computation, latency, energy, QoS, action feasibility, and distribution drift. The learning formulation is internally consistent with an advantage actor-critic implementation, and the experimental section reports actual simulation outputs rather than illustrative numerical claims. In the implemented scenario, AML-RA produced substantially higher throughput and energy efficiency than static allocation while maintaining nearly the same mean energy per control interval. The no-constraint experiment further demonstrates that feasibility mechanisms are essential. At the same time, the study deliberately avoids claiming real-world deployment readiness: larger-scale, packet-level, and hardware-based validation remains necessary. The framework therefore provides a reproducible research foundation rather than a finished production controller.

References

- [1] "Deep Reinforcement Learning Method for Energy Efficient Resource Allocation in Next Generation Wireless Networks," ACM ICCNIoT, 2020, DOI: 10.1145/3398329.3398332.
- [2] "Energy-Efficient Subchannel and Power Allocation for HetNets Based on Convolutional Neural Network," arXiv:1903.00165.
- [3] "Energy Efficient Federated Learning Over Wireless Communication Networks," IEEE Xplore, document 9264742.
- [4] Y.-H. Xu et al., "Reinforcement Learning (RL)-Based Energy Efficient Resource Allocation for Energy Harvesting-Powered Wireless Body Area Network," Sensors, DOI: 10.3390/s20010044.
- [5] Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-Efficient Radio Resource Allocation for Federated Edge Learning," arXiv:1907.06040.
- [6] J. A. Ayala-Romero et al., "Bayesian Online Learning for Energy-Aware Resource Orchestration in Virtualized RANs," IEEE INFOCOM 2021, DOI: 10.1109/INFOCOM42981.2021.9488845.
- [7] Y. Azimi et al., "Energy-Efficient Deep Reinforcement Learning Assisted Resource Allocation for 5G-RAN Slicing," IEEE Trans. Veh. Technol., vol. 71, no. 1, 2021, DOI: 10.1109/TVT.2021.3128513.
- [8] "Energy-Efficient Power Control and Resource Allocation Based on Deep Reinforcement Learning for D2D Communications in Cellular Networks," 2021.
- [9] "Convergence Time Minimization for Federated Reinforcement Learning over Wireless Networks," 2021.
- [10] "Knowledge-aided Federated Learning for Energy-limited Wireless Networks," arXiv:2209.12277, 2022.
- [11] "A Safe Deep Reinforcement Learning Approach for Energy Efficient Federated Learning in Wireless Communication Networks," arXiv:2308.10664, 2023.
- [12] "Energy-Efficient Resource Allocation Based on Deep Q-Network in V2V Communications," PMC9921350, 2023.
- [13] "Deep Reinforcement Learning for Radio Resource Allocation and Management in Next Generation Heterogeneous Wireless Networks: A Survey," arXiv:2106.00574.
- [14] M. Bordin, A. Lacava, M. Polese, S. Satish, M. A. Nittoor, R. Sivaraj, F. Cuomo, and T. Melodia, "Design and Evaluation of Deep Reinforcement Learning for Energy Saving in Open RAN," arXiv:2410.14021, 2024.
- [15] "A Survey on AI-driven Energy Optimisation in Terrestrial Next Generation Radio Access Networks," arXiv:2411.02164, 2024.
- [16] "Energy-Efficient Federated Edge Learning with Streaming Data: A Lyapunov Optimization Approach," arXiv:2405.12046, 2024.
- [17] "Improving Energy Efficiency in Federated Learning Through the Optimization of Communication Resources Scheduling of Wireless IoT Networks," arXiv:2408.01286, 2024.
- [18] M. Mehdaoui and A. Abouaomar, "Dynamics of Resource Allocation in O-RANs: An In-depth Exploration of On-Policy and Off-Policy Deep Reinforcement Learning for Real-Time Applications," arXiv:2412.01839, 2024.
- [19] H. Sherif, E. Ahmed, and A. M. Kotb, "Energy-Efficient and Accelerated Resource Allocation in O-RAN Slicing Using Deep Reinforcement Learning and Transfer Learning," Cybernetics and Information Technologies, vol. 24, no. 3, 2024, DOI: 10.2478/cait-2024-0029.
- [20] R. M. Sohaib, S. T. Shah, O. Onireti, Y. Sambo, and M. Imran, "Green Resource Allocation in Cloud-Native O-RAN Enabled Small Cell Networks," IEEE, 2024.
- [21] "Delay-sensitive resource allocation for IoT systems in 5G O-RAN networks," Internet of Things, vol. 26, 101131, 2024, DOI: 10.1016/j.iot.2024.101131.