EExApp: GNN-Based Reinforcement Learning for Radio Unit Energy Optimization in 5G O-RAN

Jie Lu, Peihao Yan, and Huacheng Zeng
Department of Computer Science and Engineering, Michigan State University, USA

***Abstract*—With over 3.5 million 5G base stations deployed globally, their collective energy consumption (projected to exceed 131 TWh annually) raises significant concerns over both operational costs and environmental impacts. In this paper, we present EExAPP, a deep reinforcement learning (DRL)-based xApp for 5G Open Radio Access Network (O-RAN) that jointly optimizes radio unit (RU) sleep scheduling and distributed unit (DU) resource slicing. EExAPP uses a dual-actor-dual-critic Proximal Policy Optimization (PPO) architecture, with dedicated actor-critic pairs targeting energy efficiency and quality-of-service (QoS) compliance. A transformer-based encoder enables scalable handling of variable user equipment (UE) populations by encoding all-UE observations into fixed-dimensional representations. To coordinate the two optimization objectives, a bipartite Graph Attention Network (GAT) is used to modulate actor updates based on both critic outputs, enabling adaptive trade-offs between power savings and QoS. We have implemented EExAPP and deployed it on a real-world 5G O-RAN testbed with live traffic, commercial RU and smartphones. Extensive over-the-air experiments and ablation studies confirm that EExAPP significantly outperforms existing methods in reducing the energy consumption of RU while maintaining QoS.**

***Index Terms*—Cellular networks, 5G, O-RAN, xAPP, energy efficiency, deep reinforcement learning, intelligent control**

I. Introduction

Over 3.5 million 5G base stations (BSs) have been deployed globally, and the number continues to grow [1]. The total electricity consumption of the 5G radio access network (RAN) is projected to exceed 131 terawatt-hours per year [2], raising serious concerns not only about operational expenses but also environmental impact due to carbon emissions. Among various components of the 5G RAN, radio units (RUs) are the most energy-intensive, accounting for up to 80% of the total energy consumption [3]. This is primarily due to their essential role in managing the physical (PHY) layer of the radio interface, including signal transmission and reception, low-PHY signal processing, and other critical operations that directly influence energy consumption [4]. As 5G networks continue to expand and adopt wider bandwidths, the energy demand of individual RUs continues to grow, making them a key focus for energy efficiency optimization within the RAN architecture.

Measurements of real-world 5G BSs show that their transmissions are highly bursty, depending on upper-layer data traffic patterns [5]. Fig. 1 shows our measurements of an O-RAN system serving smartphones engaged in typical activities such as web browsing, video streaming, messaging, and voice calls. The observed transmission pattern reveals a comb-like structure, where bursts of radio activity are interleaved with idle time slots. These idle slots, which dominate the millisecond-level timeline, align with findings in [5] showing that inactive periods account for more than half of the operating time in median cells. This presents a significant opportunity for energy savings by transitioning RUs into sleep mode during short idle intervals, without compromising service quality. In practice, RU sleep scheduling is closely tied to resource slicing at the distributed unit (DU) of the RAN. A joint optimization of these two functions can maximize RU sleep opportunities while meeting the quality-of-service (QoS) requirements of user equipment (UE), thereby balancing energy efficiency with network performance. Furthermore, the recent adoption of the O-RAN architecture provides a unique opportunity to implement this joint optimization using xApps deployed within the Near-Real-Time RAN Intelligent Controller (Near-RT RIC), enabling near-real-time adaptation to dynamic network conditions [6]–[8].

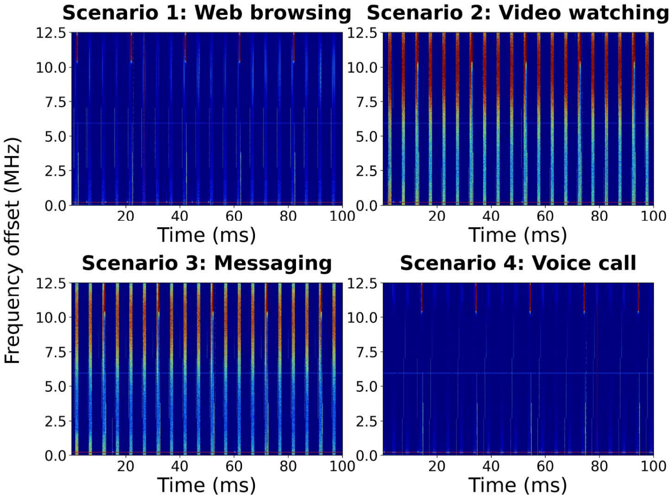

Fig. 1: The radio transmission pattern instances of a 5G base station.

In this paper, we propose a joint optimization framework for RU's sleep scheduling and DU's resource slicing in a 5G O-RAN system, with the objective of maximizing the energy efficiency of RU while maximally meeting the QoS demands of UEs. To attain this objective, we introduce EExAPP, a deep reinforcement learning (DRL)-based solution implemented as an xApp within the Near-RT RIC. EExAPP employs a *dual-actor-dual-critic* architecture built upon the model-free

Proximal Policy Optimization (PPO) algorithm. By taking the representation of the network conditions, EExAPP includes two actor networks for policy generation: (i) EE Actor generates discrete sleep scheduling decisions, determining the sleep pattern of the RU per signal frame; (ii) RS Actor produces continuous resource slicing decisions for DU to ensure the QoS compliance. Each actor is paired with a dedicated critic network, which independently evaluate energy efficiency and QoS performance. The two actors are trained using separate update strategies, allowing the framework to optimize distinct objectives effectively. Compared to its single-actor-single-critic (SASC) counterpart, this dual-actor-dual-critic architecture accelerates the learning convergence and enhances the adaptability of the DRL model.

One challenge in designing EExAPP is the time-varying number of UEs in cellular networks. In real-world networks, the UE population fluctuates as devices enter, exit, or hand over between cells [8]. Since the DRL framework relies on all-UE observations (e.g., traffic demand and performance metrics), the input state space of EExAPP varies over time, violating the fixed input dimensionality required by neural networks. To address this, EExAPP incorporates a Transformer-based encoder [9], designed in accordance with the O-RAN standard. This module processes a variable-length set of key performance indicators (KPIs) collected from the RAN's central unit (CU) and DU, encoding them into a fixed-dimensional latent representation. The encoder leverages self-attention to capture contextual relationships among UEs and supports scalability to varying input sizes. This approach significantly reduces the parameter count and inference complexity of the DRL model while preserving capacity to learn effective policies.

Another challenge lies in modeling the interaction between the two actor-critic pairs. In O-RAN, sleep scheduling and resource slicing pursue distinct objectives: one for energy savings and the other for QoS satisfactions. Training these modules in isolation can lead to suboptimal coordination. For example, extending the sleep duration of RU may degrade QoS, whereas strict QoS enforcement may limit energy-saving opportunities. To capture these inter-dependencies, EExAPP employs a bipartite Graph Attention Network (GAT) to connect the two actor-critic pairs. The bipartite GAT dynamically learns how much influence each critic should exert on each actor during training. This allows each actor to receive a weighted combination of both critic values, effectively balancing the two objectives while maintaining modular learning for each actor-critic pair.

We implemented EExAPP as an xApp and deployed it on a 5G O-RAN testbed consisting of one commercial BS and eight smartphones. Experimental results show that the dual-actor–dual-critic architecture consistently outperforms its single-actor–single-critic counterpart. Ablation studies further confirm that both the encoder and GAT modules significantly improve EExAPP's policy learning. End-to-end system evaluations demonstrate that EExAPP achieves superior performance compared to state-of-the-art (SOTA) baselines.

This work advances the SOTA as follows:

- EExAPP presents a novel joint optimization framework for RU sleep scheduling and DU resource slicing in 5G O-RAN, using a dual-actor-dual-critic DRL architecture that efficiently handles dynamic UE populations.
- EExAPP is deployable in commercial O-RAN systems, with decision-making offloaded to the Near-RT RIC, which operates under more relaxed latency constraints compared to the DU.
- Extensive over-the-air experiments show that EExAPP significantly outperforms the existing approaches.

II. System Modeling

A. A Primer on 5G NR and O-RAN

5G and Beyond. 5G New Radio (NR) introduces a flexible and highly scalable frame structure designed to accommodate the diverse service requirements of 5G networks, ranging from enhanced mobile broadband (eMBB) and ultra-reliable low-latency communications (URLLC) to massive machine-type communications (mMTC) [10]. The 5G NR frame structure is built around a 10-ms frame, which is divided into subframes, each lasting 1 ms. These subframes are further divided into slots, and each slot can consist of 14 symbols. The flexibility of the frame structure is largely driven by numerology, which determines the time-frequency grid's granularity based on different subcarrier spacings. While cellular networks evolve from 5G to 6G, the frame structure and numerology are likely to remain largely the same to ensure backward compatibility.

O-RAN. The architecture of O-RAN is designed to be open, flexible and modular, enabling the integration of diverse vendors' equipment and improving network management, orchestration, and performance [11]. RU, DU, and CU are the key components in this architecture. RU is responsible for handling the lower layers of the radio interface, including signal processing for PHY transmission and reception. It manages the antenna and RF components, which are physically located at the cell sites. DU is responsible for the lower layers of the protocol stack (L2 and L3), including the MAC, RLC, and PDCP layers [12]. It performs functions like scheduling, beamforming, and other radio resource management tasks. CU is in charge of the upper layers of the protocol stack, such as SDAP, RRC, and NAS. The O-RAN architecture also includes the Near-RT RIC, which plays a central role in enhancing RAN's performance through near-real-time decision-making and optimization. It can interact with the DU and CU to monitor the network performance and adjust parameters. The Near-RT RIC operates in coordination with xApps, which are specific applications deployed within the Near-RT RIC to provide customized functionalities such as resource management, traffic steering, and energy efficiency optimization.

Energy Saving in 3GPP. 3GPP has standardized a range of energy-saving mechanisms and sleep modes aimed at reducing the power consumption of RANs. These mechanisms target both network-side components (e.g., RU and DU) and UEs. At the network level, Carrier-Level Sleep allows dynamic activation and deactivation of carriers based on traffic demand, effectively enabling small cells or secondary carriers to power down

TABLE I: Notation.

Symbol	Explanation
$\mathcal{I}$	The set of slices
$\mathcal{K}_i$	The set of UEs in slice $i \in \mathcal{I}$
$q_{t,k}$	Average data throughput of UE k in timestep t
$d_{t,k}$	Average data queueing delay of UE k in timestep t
$\mathbf{s}_t$	Observation of RAN at time frame t
$\hat{\mathbf{s}}_t$	Encoded RAN observation, i.e., state of RL, in timestep t
$V(\hat{\mathbf{s}}_t)$	Individual value function of critic in RL
$\hat{V}(\hat{\mathbf{s}}_t)$	Aggregated value functions of critic in RL
$\boldsymbol{\alpha}_t$	$\boldsymbol{\alpha}_t = [a_t, b_t, c_t]$ is for RU sleep control (RL's sleep action) in time frame t, where b_t is the # of time slots for sleep
$\boldsymbol{\beta}_t$	$\boldsymbol{\beta}_t = [\beta_{t,i}]$ is for RU's slicing control (RL's slicing action)

during low-utilization periods [13], [14]. Transmitter/Receiver (TRx) On/Off, also referred to as RU Sleep, selectively powers down RF chains or antenna elements within a radio unit when transmission or reception is not needed [13], [15], [16]. Similarly, MIMO Layer/Chain Deactivation allows high-order MIMO systems to reduce energy consumption by disabling unused antenna paths [15]. In addition, a mechanism referred to as DU Sleep Mode Control has been defined in 3GPP [14], allowing CU to initiate the sleep mode of DU. On the UE side, 3GPP defines Discontinuous Reception (DRX) in both LTE and NR to conserve battery power during idle periods, which also indirectly contributes to overall RAN energy savings.

B. Problem Formulation

We formulate the joint energy saving and resource slicing problem as an optimization problem. Table I lists our key symbol representations.

Resource Slicing: Consider a 5G O-RAN system that comprises RU, DU, CU and Near-RT RIC. Denote $\mathcal{I}$ as the set of slices in the RAN, with $I = |\mathcal{I}|$. Denote $\mathcal{K}_i$ as the set of UEs assigned within slice $i \in \mathcal{I}$, with $\mathcal{K} = \cup_{i \in \mathcal{I}} \mathcal{K}_i$. Referring to Fig. 2, denote $\beta_{t,i}$ as the percentage of the physical resource blocks (PRBs) that are allocated for slice $i \in \mathcal{I}$ in timestep t. Denote $\boldsymbol{\beta}_t = [\beta_{t,1}, \beta_{t,2}, \ldots, \beta_{t,i}]$, with $\sum_{i \in \mathcal{I}} \beta_{t,i} = 1$. $\boldsymbol{\beta}_t$ is the online optimization variables for resource slicing in t.

Sleep Scheduling: Again, referring to Fig. 2, each frame is 10 ms and divided into $N_{\text{ts}} = 2^{\mu} \times 10$ slots, depending on the 3GPP numerology adopted by the O-RAN, with $\mu \in \{0, 1, 2, 3, 4\}$ [17]. Among the N_{ts} slots in a frame, denote b_t as the number of slots scheduled for the sleeping of the RU in timestep t. To denote the position of sleep slots, we let a_t and c_t be the number of active slots before and after the sleep period. Then, we have $a_t + b_t + c_t = N_{\text{ts}}$. For simplicity, we let $\boldsymbol{\alpha}_t = [a_t, b_t, c_t]$. $\boldsymbol{\alpha}_t$ is the online optimization variables for the sleep scheduling of RU. We note that, as shown in Fig. 2, an timestep may span over multiple frames. In this case, we apply the same sleep scheduling decision to individual frames.

QoS Demands: Different slices are to meet different QoS demands. Denote Q_i and D_i as the target throughput and delay demands of UEs in slice $i \in \mathcal{I}$. Denote $q_{t,k}$ as the achievable throughput and $d_{t,k}$ as the achievable delay of UE $k \in \mathcal{K}$ in timestep t. Obviously, many factors may affect $q_{t,k}$ and $d_{t,k}$, including the variables of resource slicing and sleep scheduling. We denote them as: $q_{t,k} = f(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{s}_t, k)$ and $d_{t,k} = g(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{s}_t, k)$, where $\boldsymbol{s}_t$ is the global RAN state/condition in timestep t. Then, the QoS constraints can be expressed as: $q_{t,k} \geq Q_i$ and $d_{t,k} \leq D_i$ for $k \in \mathcal{K}_i$ and $i \in \mathcal{I}$. It is worth noting that in practice, $f(\cdot)$ and $g(\cdot)$ are unknown and hard to estimate due to the complex network conditions.

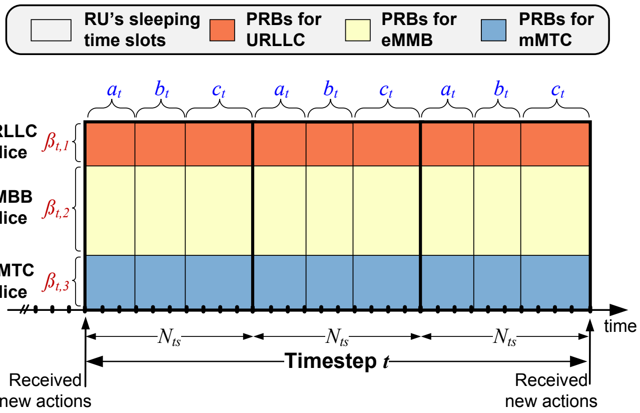

Fig. 2: Illustrating the online policy for joint resource slicing and power-saving optimization at an O-RAN RU.

Formulation: The objective of this work is to develop a policy that can maximize the sleep time of the RU while meeting the QoS demands of individual UEs in the network. Specifically, we model the optimization objective by $\max(\frac{b_t}{N_{\text{ts}}})$. Based on this objective, we formulate the optimization as follows:

$$\max \quad \mathbb{E}\left[\lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \frac{b_t}{N_{\text{ts}}}\right] \tag{1a}$$

$$\text{s.t. } q_{t,k} = f(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{s}_t, k) \geq Q_i, \quad k \in \mathcal{K}_i, i \in \mathcal{I}, \tag{1b}$$

$$d_{t,k} = g(\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t, \boldsymbol{s}_t, k) \leq D_i, \quad k \in \mathcal{K}_i, i \in \mathcal{I}, \tag{1c}$$

$$a_t + b_t + c_t = N_{\text{ts}}, \tag{1d}$$

$$\sum_{i \in \mathcal{I}} \beta_{t,i} = 1, \tag{1e}$$

where $t = 1, 2, \ldots$ is the timestep of decision making process. Q_i and D_i are the throughput and delay demands of slice i, respectively. Constraints (1b) and (1c) ensure that the QoS demands are met for every UE in every slice. (1d) and (1e) characterizes the underlying relations of optimization variables, as explained before.

C. MDP Modeling and Relaxation

The problem in (1) is a constrained stochastic optimization problem with mixed-integer variables. Due to the time-varying network condition $\boldsymbol{s}_t$, the behavior of the system $q_{t,k}$ and $d_{t,k}$ evolve dynamically over time with the complex yet unknown functions $f(\cdot)$ and $g(\cdot)$. This non-stationary and black-box nature makes the problem impractical to solve using static and deterministic methods. DRL has emerged as an efficient approach for stochastic optimization, as it can learn from the dynamic environment and adapt its policy based on observed

TABLE II: The per-UE KPI observations from the RAN.

Data Type	Metric Name	UL/DL	Description
KPM data	PDCP SDU	UL&DL	Data payload unit
	Delay	DL	SDU delay of RLC
	Throughput	UL&DL	Actual data achieved by UE
	PRB	UL&DL	Physical resource blocks
MAC data	TBS	DL	Current transport block size
	RB	DL	Scheduled resource blocks
	PUSCH SNR	UL	SNR of PUSCH
	PUCCH SNR	UL	SNR of PUCCH
	CQI	DL	Channel quality indicator
	MCS	UL&DL	Modulation and coding scheme
	PHR	UL	Power headroom report
	BLER	UL&DL	Block error rate

states. In particular, model-free DRL is well-suited for online decision-making in this scenario, as it operates effectively without requiring explicit knowledge of the underlying functions $f(\cdot)$ and $g(\cdot)$ [6].

In what follows, we formulate the optimization problem as a Markov Decision Process (MDP) in the O-RAN environment, aimed at developing a DRL-based xApp for online optimization of joint sleep scheduling and resource slicing.

- **State $\boldsymbol{s}_t$:** The state representation in our DRL model is constructed from key performance indicator (KPI) generated by the operation of the RAN, including both CU and DU. This data captures the performance of individual UEs and overall network conditions. Table II summarizes the KPI data that an xApp can obtain from the RAN via standard E2 interface. It includes UE-level metrics such as throughput, delay, and allocated physical resource blocks (PRBs), as well as MAC-layer indicators like the signal-to-noise ratio (SNR) of the Physical Uplink Shared Channel (PUSCH) and Physical Uplink Control Channel (PUCCH), power headroom report (PHR), modulation and coding scheme (MCS). *We stress that, while there is a considerable body of work on DRL for 5G network optimization, most existing studies rely on simulated data. In contrast, this work uniquely models the DRL framework using a realistic dataset.*
- **Action $\boldsymbol{a}_t = [\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t]$:** As illustrated in Fig. 2, the action in our DRL model comprises two components: $\boldsymbol{\alpha}_t = [a_t, b_t, c_t]$, representing discrete decisions for sleep scheduling, and $\boldsymbol{\beta}_t = [\beta_{t,1}, \beta_{t,2}, \ldots, \beta_{t,I}]$, representing continuous decisions for resource slicing. This defines a hybrid action space. The discrete action $\boldsymbol{\alpha}_t$ determines the time scheduling of active and sleep slots for the RU, while the continuous action $\boldsymbol{\beta}_t$ controls the resource allocation among multiple network slices to support key 5G service types such as URLLC, eMBB, and mMTC.
- **Reward r_t:** *Problem* (1) aims to maximize the energy efficiency of RU while ensuring that each service slice satisfies QoS constraints. However, in practice, due to the unpredictable traffic patterns and limited resources in the wireless channel, the QoS constraints may not always be strictly satisfied. To address this, we reformulate *Problem* (1) using the Lagrangian relaxation technique, which allows for soft constraint violation by introducing penalty terms into the objective function. By incorporating the QoS constraints into the objective function, we have

$$L(\pi, \lambda_p, \lambda_d) = \mathbb{E}\Bigg[\frac{1}{T}\sum_{t=1}^{T}\Bigg(\frac{b_t}{N_{\text{ts}}} - \lambda_q \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}(1-\frac{q_{t,k}}{Q_i})^{+} - \lambda_d \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}(\frac{d_{t,k}}{D_i}-1)^{+}\Bigg)\Bigg], \quad (2)$$

where $(\cdot)^{+} = \max(0, \cdot)$, λ_q and λ_d are Lagrangian multipliers for the throughput and delay constraints, respectively. In our experiments, we observed that some packets may have large delay and thus dominate the objective function. To address this issue, we define $\hat{d}_{t,k} = \min(d_{t,k}, 2D_i)$. Then, based on the Lagrangian reformulation in (2), the reward at time step t is computed by:

$$r_t = \frac{b_t}{N_{\text{ts}}} - \lambda_q \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}(1-\frac{q_{t,k}}{Q_i})^{+} - \lambda_d \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}(\frac{\hat{d}_{t,k}}{D_i}-1)^{+}, \quad (3)$$

where λ_q and λ_d are empirically set.
- **Policy π:** The agent's goal is to learn a policy $\pi(\boldsymbol{a}_t|\boldsymbol{s}_t)$ that maximizes the reward function in Eq (3) over time.
- **Transition:** $P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$ defines how the system state evolves from the current state $\boldsymbol{s}_t$ to the next state $\boldsymbol{s}_{t+1}$ after taking action $\boldsymbol{a}_t$.

III. EExAPP: Design

A. Overview

To solve the above MDP problem, we propose EExAPP, a dual-actor-dual-critic DRL framework based on *Proximal Policy Optimization (PPO)*. Fig. 3 shows the architecture of EExAPP. It deploys two actor networks: EE Actor and RS Actor. EE Actor is responsible for generating discrete *sleep scheduling* decisions, and RS Actor outputs continuous *resource slicing* decisions. The two actors use separate updating strategies, enabling EExAPP to prioritize distinct optimization objectives: EE Actor focuses on energy efficiency, while RS Actor learns to maximize QoS satisfaction.

To support these objectives, EExAPP incorporates two dedicated critics (EE Critic and RS Critic in Fig. 3) that independently evaluate energy efficiency and QoS satisfaction. EExAPP is a modular DRL for joint energy efficiency and QoS optimization in O-RAN control. Built on the model-free PPO algorithm, it separates decision-making into two coordinated policies in EE Actor and RS Actor. A lightweight Transformer Encoder processes variable-length UE observations into fixed-size representations, enabling adaptation to dynamic network conditions. A bipartite GAT mechanism fuses critic evaluations to capture the coupling between energy-saving and QoS objectives, supporting stable and effective joint optimization in highly variable wireless environments.

The design of EExAPP faces two challenges. One key challenge is that the input state to EExAPP is dynamic in size and composition, due to fluctuations in the number of

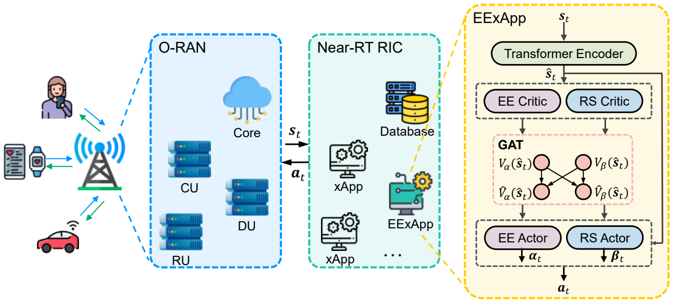

Fig. 3: The architecture of EExAPP in O-RAN system. The EExAPP is deployed on the Near-RT RIC, interacting with the RAN (CU and DU) to perform closed-loop control. Internally, the framework utilizes a Transformer encoder to extract latent features from real-time network states $\boldsymbol{s}_t$. A dual-actor-dual-critic structure is employed to decouple the optimization of energy efficiency (EE) and resource slicing (RS), coordinated by a bipartite GAT that aggregates critic values to balance conflicting objectives in the joint action $\boldsymbol{a}_t$.

UEs connected to the gNB and the set of active slices over time. Specifically, the number of UEs changes rapidly in real-world deployments as devices arrive, leave, or hand over between cells. This presents a difficulty because the PPO models, which underlie EExAPP, require a fixed-size input data dimensionality for both the actor and critic networks. To address this challenge, we employ a *Transformer Encoder* to process the UE state information before policy generation. The Transformer Encoder allows EExAPP to represent a set of UE observations of varying size in a unified, fixed-dimensional latent space. Details are provided in §III-B.

The second challenge lies in the modeling of interaction between the two actor-critic pairs for joint optimization. In O-RAN control, resource slicing and sleep scheduling target distinct objectives: one for QoS satisfaction and the other for energy efficiency. While using separate actor-critic pairs enables specialization, it overlooks their potential interaction. For instance, increasing sleep duration may degrade QoS, while strict QoS enforcement reduces energy-saving opportunities. To model this inter-dependence while maintaining modularity between actor-critic pairs, we introduce a bipartite GAT that connects two critic nodes to two actor nodes in a bipartite graph. The attention mechanism learns how much each critic should influence each actor, allowing each actor to receive a weighted combination of both critic values based on their relevance to its policy update. Details are provided in §III-D.

B. Transformer Encoder for Dynamic Input Representation

At each timestep t, EExAPP acquires a set of real-time observations $\boldsymbol{s}_t = [\boldsymbol{s}_{t,1}, \boldsymbol{s}_{t,2}, \dots, \boldsymbol{s}_{t,K}] \in \mathbb{R}^{K \times F}$ from the DU through the E2 interface, where K denotes the number of active UEs. Each $\boldsymbol{s}_{t,k} \in \mathbb{R}^F$ consists of F KPI features of the network. The raw features $[\boldsymbol{s}_{t,k}]$ are first projected into a latent embedding space:

$$\boldsymbol{e}_{t,k} = \boldsymbol{W}_e \boldsymbol{s}_{t,k} + \boldsymbol{b}_e, \quad \boldsymbol{e}_{t,k} \in \mathbb{R}^d, \quad k \in \mathcal{K}, \tag{4}$$

where d is the embedding dimension, and $\boldsymbol{W}_e \in \mathbb{R}^{d \times F}$ and $\boldsymbol{b}_e \in \mathbb{R}^d$ are learnable parameters. The sequence of projected embeddings $\boldsymbol{e}_t = [\boldsymbol{e}_{t,1}, \boldsymbol{e}_{t,2}, \dots, \boldsymbol{e}_{t,K}] \in \mathbb{R}^{K \times d}$ is then processed by a lightweight *Transformer Encoder* tailored for the low-latency and near-RT inference in wireless networks. Specifically, our Encoder comprises 2 self-attention layers, with a reduced embedding dimension $d = 64$ and 4 attention heads. The feedforward network hidden size is reduced to $2d = 128$, significantly reducing parameters and inference complexity while retaining Transformer's key properties: contextual embedding of UE states and scalability to dynamic input sizes. This yields a sequence of contextually enriched UE embeddings:

$$\boldsymbol{h}_t = [\boldsymbol{h}_{t,1}, \boldsymbol{h}_{t,2}, \dots, \boldsymbol{h}_{t,K}], \quad \boldsymbol{h}_{t,k} \in \mathbb{R}^d, \tag{5}$$

The output at this stage still reflects the dynamic size of the input (i.e., it depends on the size of K). To satisfy the fixed-size input requirement of the policy optimization, we apply a mean pooling operation over all UE embeddings to construct a fixed-size global context vector:

$$\hat{\boldsymbol{s}}_t = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{h}_{t,k}, \tag{6}$$

Consequently, the resulting global state $\hat{\boldsymbol{s}}_t$ is a fixed-size vector of length d.

C. Dual-Actor-Critic Architecture

From a systems-level perspective, maintaining two specialized policies instead of a single unified one enhances modularity and adaptability. While a unified policy risks over-generalization and suboptimal performance across diverse conditions, distinct policies can be tailored to capture regime-specific dynamics, enabling targeted optimization and greater robustness.

The encoded state $\hat{\mathrm{s}}_t$ is then processed in parallel by actor and critic networks. The discrete EE Actor is responsible for sampling a sleep time decision $\boldsymbol{\alpha}_t \sim \pi_\alpha(\boldsymbol{\alpha}_t | \hat{\mathrm{s}}_t)$ from a categorical policy over $\boldsymbol{\alpha}_t = (a_t, b_t, c_t)$. Simultaneously, the continuous RS Actor samples a slicing decision $\boldsymbol{\beta}_t \sim \pi_\beta(\boldsymbol{\beta}_t | \hat{\mathrm{s}}_t)$ from a Gaussian policy. After applying the joint action $\boldsymbol{a}_t = [\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t]$ to the O-RAN system, the environment transitions to the next observation s_{t+1} and returns the reward r_t. The global reward is decomposed into two components, $r_{t,\alpha}$ and $r_{t,\beta}$, targeting energy efficiency and QoS satisfaction, respectively:

$$\begin{cases} r_{t,\alpha} = \dfrac{b_t}{N_{\mathrm{ts}}}, \\ r_{t,\beta} = -\lambda_q \sum\limits_{i \in \mathcal{I}} \sum\limits_{k \in \mathcal{K}_i} (1 - \dfrac{q_{t,k}}{Q_i})^+ - \lambda_d \sum\limits_{i \in \mathcal{I}} \sum\limits_{k \in \mathcal{K}_i} (\dfrac{\hat{d}_{t,k}}{D_i} - 1)^+. \end{cases} \tag{7}$$

Parallel to the actors, the critics estimate the expected returns. The discrete critic $V_\alpha(\hat{\mathrm{s}}_t)$ estimates the value for the sleep action, while the continuous critic $V_\beta(\hat{\mathrm{s}}_t)$ estimates the value for the slicing action. These raw value estimates are further processed to form *aggregated* critic estimates, $\hat{V}_\alpha(\hat{\mathrm{s}}_t)$

and $\hat{V}_\beta(\hat{\mathrm{s}}_t)$, which incorporate a weighted combination of both critics via a bipartite GAT (detailed in §III-D).

To compute the training targets, we adopt the Generalized Advantage Estimation (GAE) method to balance bias and variance over finite-horizon trajectories [18]. At each timestep t, the GAE is calculated for each actor using its corresponding reward and the *aggregated* value estimate. For `EE Actor` (and similarly for the `RS Actor`), the Temporal Difference (TD) error $\delta_{t,\alpha}$ is defined as:

$$\delta_{t,\alpha} = r_{t,\alpha} + \gamma \hat{V}_\alpha(\hat{\mathrm{s}}_{t+1}) - \hat{V}_\alpha(\hat{\mathrm{s}}_t), \tag{8}$$

where $\gamma \in [0, 1]$ is the discount factor. The advantage estimate $A_{t,\alpha}$ is then obtained by exponentially weighting future TD errors with the GAE parameter $\lambda \in [0, 1]$:

$$A_{t,\alpha} = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l,\alpha}, \tag{9}$$

where T denotes the trajectory length.

The actor network, parameterized by θ_α, is updated by maximizing the Clip objective function:

$$\mathcal{L}(\theta_\alpha) = \mathbb{E}_t \left[\min\left(\rho_t(\theta_\alpha) A_{t,\alpha},\ \mathrm{clip}(\rho_t(\theta_\alpha), 1-\epsilon, 1+\epsilon) A_{t,\alpha} \right) \right], \tag{10}$$

where $\rho_t(\theta_\alpha) = \frac{\pi_{\theta_\alpha}(\boldsymbol{\alpha}_t|\hat{\mathrm{s}}_t)}{\pi_{\theta_{\mathrm{old}}}(\boldsymbol{\alpha}_t|\hat{\mathrm{s}}_t)}$ is the probability ratio, and $\epsilon \in [0.1, 0.2]$ is the clipping hyperparameter.

The critic network, parameterized by ϕ_α, minimizes the error between the predicted value $V_\alpha(\hat{\mathrm{s}}_t)$ and the target return $R_{t,\alpha}$. Consistent with GAE, the target return is defined as the sum of the current value estimate and the computed advantage:

$$R_{t,\alpha} = \hat{V}_\alpha(\hat{\mathrm{s}}_t) + A_{t,\alpha}. \tag{11}$$

To enhance the robustness, we employ the Huber loss:

$$\mathcal{L}(\phi_\alpha) = \mathbb{E}_t \left[f_\zeta(V_\alpha(\hat{\mathrm{s}}_t) - R_{t,\alpha}) \right], \tag{12}$$

where $f_\zeta(\cdot)$ is the Huber loss function with threshold ζ:

$$f_\zeta(x) = \begin{cases} \frac{1}{2}x^2 & \text{if } |x| \le \zeta, \\ \zeta\left(|x| - \frac{1}{2}\zeta\right) & \text{otherwise.} \end{cases} \tag{13}$$

The resource slicing actor-critic pair $(\theta_\beta, \phi_\beta)$ follows the identical update procedure using $r_{t,\beta}$ and $\hat{V}_\beta$.

D. GAT for Critic Aggregation

In O-RAN control, resource slicing and sleep scheduling target distinct objectives: one for QoS satisfaction and the other for energy efficiency. Employing separate actor-critic pairs allows for specialized learning, but it overlooks the interplay between these objectives. For example, extending sleep duration may compromise QoS, whereas rigid QoS requirements can limit opportunities for energy savings. To model this interplay while maintaining modularity between actor-critic pairs, we introduce a GAT that connects two critic nodes to two actor nodes within a bipartite graph structure. The attention mechanism dynamically learns the degree to which each critic should influence each actor, allowing each actor to incorporate a weighted combination of critic values relevant to its policy update.

Algorithm 1 EExApp training process.

```
Input: Learning rates $\eta_\theta$, $\eta_\phi$, discount factor $\gamma$, GAE parameter $\lambda$
Initialize: $\theta_\alpha$, $\theta_\beta$ for actors, $\phi_\alpha$, $\phi_\beta$ for critics, bipartite graph $G = (\mathcal{S}, \mathcal{T}, \mathcal{E})$
for $t = 1, \ldots, T$ do
    Observe the network state $\boldsymbol{s}_t$
    Encode the state $\hat{\mathrm{s}}_t$ using Transformer
    Execute actions $\boldsymbol{a}_t = [\boldsymbol{\alpha}_t, \boldsymbol{\beta}_t]$ sampled from $\pi_\theta(\boldsymbol{a}_t|\hat{\mathrm{s}}_t)$ and store the transition $(\hat{\mathrm{s}}_t, \boldsymbol{a}_t, r_{t,\alpha}, r_{t,\beta}, \mathrm{s}_{t+1})$
    Calculate raw critic values $V_\alpha(\hat{\mathrm{s}}_t), V_\beta(\hat{\mathrm{s}}_t)$ and aggregate them into $\hat{V}_\alpha(\hat{\mathrm{s}}_t), \hat{V}_\beta(\hat{\mathrm{s}}_t)$ via GAT
    Compute the advantage estimate $A_{t,\alpha}, A_{t,\beta}$ using GAE based on collected trajectory
    Compute the probability ratio $\rho_t(\theta_\alpha), \rho_t(\theta_\beta)$
    Optimize the clipped objective function $\mathcal{L}(\theta)$ for each actor: $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}(\theta)$
    Update the value function (critic) using Huber loss $\mathcal{L}(\phi)$: $\phi \leftarrow \phi - \eta_\phi \nabla_\phi \mathcal{L}(\phi)$
```

To model the relationship between the separate critic values (i.e., $V_\alpha(\hat{\mathrm{s}}_t)$ and $V_\beta(\hat{\mathrm{s}}_t)$) and the aggregated critic values (i.e., $\hat{V}_\alpha(\hat{\mathrm{s}}_t)$ and $\hat{V}_\beta(\hat{\mathrm{s}}_t)$), we create a lightweight bipartite graph $G = (\mathcal{S}, \mathcal{T}, \mathcal{E})$, where $\mathcal{S} = \{S_\alpha, S_\beta\}$ represents the set of source nodes, $\mathcal{T} = \{T_\alpha, T_\beta\}$ represents the set of target nodes, and $\mathcal{E} = \mathcal{S} \times \mathcal{T}$ denotes the edges. For the two source nodes, the node features are defined as the raw critic estimates: $V_\alpha(\hat{\mathrm{s}}_t)$ for S_α and $V_\beta(\hat{\mathrm{s}}_t)$ for S_β. For the two target nodes, we utilize their corresponding raw estimates as the initial query features: $V_\alpha(\hat{\mathrm{s}}_t)$ for T_α and $V_\beta(\hat{\mathrm{s}}_t)$ for T_β. Leveraging the bipartite GAT mechanism, we update the feature of target nodes by:

$$\hat{V}_j(\hat{\mathrm{s}}_t) = \sum_{i \in \{\alpha,\beta\}} \gamma_{ij} w_s V_i(\hat{\mathrm{s}}_t), \quad j \in \{\alpha, \beta\}, \tag{14}$$

where γ_{ij} is attention score given by $\gamma_{ij} = \frac{\exp(e_{ij})}{\sum_{i' \in \{\alpha,\beta\}} \exp(e_{i'j})}$ with $e_{ij} = \mathrm{LeakyReLU}\left([w_s V_i(\hat{\mathrm{s}}_t), w_t \hat{V}_j(\hat{\mathrm{s}}_t)] \mathbf{p}^\top \right)$ for $i, j \in \{\alpha, \beta\}$. Here, $w_s \in \mathbb{R}$ and $w_t \in \mathbb{R}$ are learnable parameters, $\mathbf{p}$ is learnable parameters as well.

The features of the target nodes, i.e., $\hat{V}_\alpha(\hat{\mathrm{s}}_t)$ and $\hat{V}_\beta(\hat{\mathrm{s}}_t)$, represent the aggregated critic value functions, which are used to update the policy in each timestep as described above.

E. Training Logic

The training process of `EExAPP` is summarized in Alg. 1. Each training iteration begins with the collection of interaction trajectories between `EExAPP` and the O-RAN environment. At each timestep t, the `EExAPP` receives a sequence of KPI observations $\boldsymbol{s}_t$, which are encoded into a fixed-dimensional state representation $\hat{\boldsymbol{s}}_t$ via a Transformer-based encoder. This latent state then serves as the input to all actor and critic networks. For each state $\hat{\boldsymbol{s}}_t$, two actor-critic pairs operate in parallel: the discrete `EE Actor` samples a sleep decision

$\boldsymbol{\alpha}_t \sim \pi_\alpha(\boldsymbol{\alpha}_t|\hat{\boldsymbol{s}}_t)$ from a categorical policy, while the continuous RS Actor samples a slicing decision $\boldsymbol{\beta}_t \sim \pi_\beta(\boldsymbol{\beta}_t|\hat{\boldsymbol{s}}_t)$ from a Gaussian policy, At the same time, a critic $V_\alpha(\hat{\boldsymbol{s}}_t)$ estimates the return from the sleep action $\boldsymbol{\alpha}_t$, while another critic $V_\beta(\hat{\boldsymbol{s}}_t)$ evaluates the long-term reward from the slicing action $\boldsymbol{\beta}_t$. These value estimates are passed to a bipartite GAT, which models the interdependence between the two control tasks. The GAT produces aggregated critic values $\hat{V}_\alpha(\hat{\boldsymbol{s}}_t)$ and $\hat{V}_\beta(\hat{\boldsymbol{s}}_t)$, which are used in subsequent advantage estimation and policy updating.

IV. Experimental Evaluation

In this section, we present a series of experiments to evaluate the effectiveness of the proposed EExAPP. Specifically, we aim to address the following research questions:

- **Q1 (§IV-B)**: How does EExAPP perform in the realistic O-RAN deployment?
- **Q2 (§IV-B)**: Does the dual-actor-dual-critic architecture outperform its single-actor-single-critic counterpart?
- **Q3 (§IV-C)**: What is the individual impact of the Transformer encoder and the GAT-based aggregator on the overall system?
- **Q4 (§IV-D)**: How does EExAPP compare against the SOTA solutions?

A. Implementation and Experimental Setup

We established an end-to-end experimental testbed to evaluate EExAPP in a realistic indoor environment. As illustrated in Fig. 4, our system integrates a 5G Core, RAN components (CU and DU), a commercial RU, a Near-RT RIC, and eight commercial smartphones serving as UEs. As shown in Fig. 4 (Left), the experiment is conducted in a laboratory floor plan where the RU is fixed at a specific location, and the UEs are distributed across different positions to capture diverse channel conditions and spatial dynamics. Fig. 4 (Right) further details the network topology and hardware connections among these components. Specific hardware specifications and software versions are listed in Table III.

On the software side, we implemented the O-RAN system using open-source repositories from OAI for both the 5G core and RAN components [19], [20]. To enable fine-grained sleep scheduling and resource slicing, we modified the underlying OAI 5G codebase: specifically, we updated openair2/E2AP/RAN_FUNCTION for slicing service modeling and openair2/LAYER2/NR_MAC_gNB for real-time gNB transmission control. On the hardware side, the Pegatron PR1450 serves as the commercial RU, configured with 4×4 MIMO antennas and operating in the n78 band (TDD mode) with a 30 kHz subcarrier spacing. EExAPP is deployed on the Near-RT RIC using the FlexRIC platform [21], interacting with the RAN via the E2AP (v2.03) and KPM (v2.03) service models. The UEs consist of heterogeneous smartphones running various Android versions.

For evaluation, we use the reward function defined in §II-C, along with the QoS violation ratio, which quantifies service degradation due to unmet QoS requirements. The single-actor-single-critic (SASC) approach serves as the primary baseline to validate the benefits of our decoupled architecture.

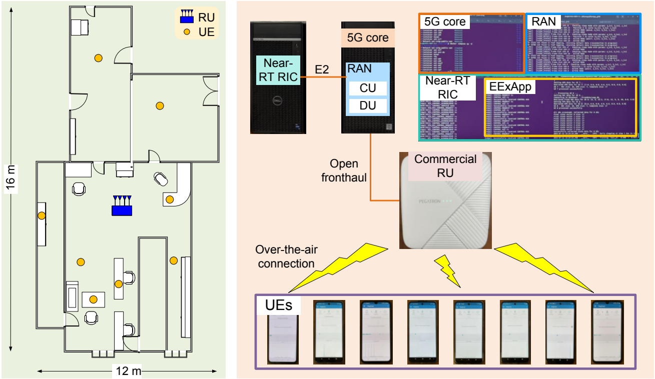

Fig. 4: Our O-RAN system for experimental evaluation. The left subfigure shows the floor plan of our system deployment, and the right subfigure shows the network elements and architecture.

TABLE III: Key components of our testbed.

Unit	Hardware	Software
5G Core	Intel Core i7-12700	OAI CN5G [19]
RAN	Intel Core i7-12700	OAI 5G [20]
SDR	Pegatron PR1450	1.0.3.1p4
Near-RT RIC	Intel Core i7-10700	FlexRIC with E2AP v2.03 and KPM v2.03 [21]
UEs	OnePlus Nord AC2003, Motorola G54 5G, Samsung Galaxy A15 5G	Android 11, 13, 14, 15

B. Case Studies under Diverse Network Conditions

To evaluate the robustness of EExAPP under diverse network conditions, we conduct extensive experiments across varying traffic loads and slicing configurations against the SASC baseline. We define three per-UE target traffic levels generated via iPerf (UDP): light (0.1–1 Mbps), medium (1–5 Mbps), and heavy (5–10 Mbps). Furthermore, we evaluate scenarios with 2, 4, and 8 slices, with the 8 UEs distributed evenly across them.

Fig. 5 illustrates the convergence performance. EExAPP consistently stabilizes around 500 timesteps, significantly faster than SASC, which typically converges after 750 steps. As traffic load and slice count increase, both methods exhibit more significant reward fluctuations. This phenomenon occurs because increasing the number of slices leads to resource fragmentation. With fewer resources dedicated to each slice, the statistical multiplexing gain is diminished, limiting scheduling flexibility. In such cases, even minor under-provisioning can cause QoS violations, which negatively impact the reward.

Moreover, under heavy traffic, the system operates closer to or beyond its capacity, intensifying resource contention and QoS violations (detailed in §IV-D). Concurrently, high buffer occupancy constrains RU sleep opportunities, thereby reducing energy efficiency. These factors collectively degrade reward performance, corroborating our theoretical analysis.

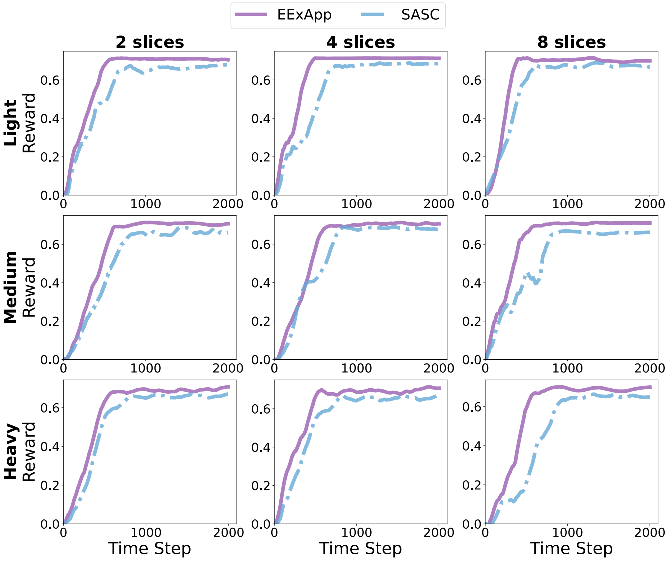

Fig. 5: Convergence performance comparison of EExAPP and SASC under light, medium, and heavy traffic conditions.

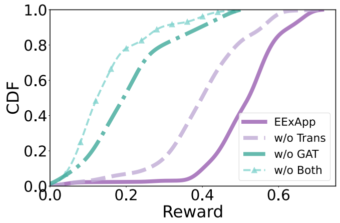

Fig. 6: The reward CDF of ablation studies.

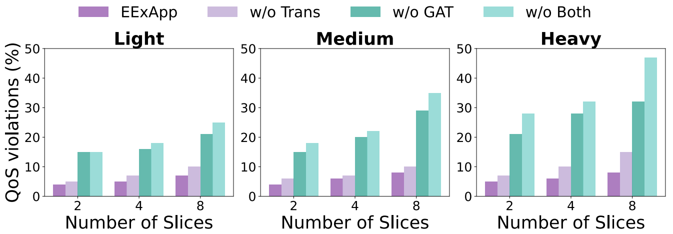

Fig. 7: The QoS violation comparison of ablation studies.

While both models achieve comparable asymptotic rewards under light loads, EExAPP exhibits superior convergence speed. In complex scenarios involving high loads or dense slicing, EExAPP outperforms SASC in both convergence rate and final reward. This demonstrates the sample efficiency of the dual-actor-critic architecture. By decoupling optimization objectives, EExAPP mitigates gradient conflicts inherent in competing tasks. Conversely, SASC suffers from slower adaptation and heightened volatility due to the challenge of balancing trade-offs within a monolithic policy. A detailed comparison of QoS violations is provided in §IV-D.

C. Ablation Studies

In this subsection, we conduct the ablation studies to evaluate the individual contributions of the key components within EExAPP. Specifically, we investigate the impact of two critical modules: the Transformer Encoder and the GAT aggregator. We derive three variants of EExAPP by selectively removing or replacing these components:

- **w/o Trans:** The Transformer Encoder is replaced with a 2-layer MLP, removing the capability to model temporal dependencies and contextual UE states.
- **w/o GAT:** The GAT module used for coordinating dual critics is removed, forcing the actor updates to rely solely on independent critic feedback.
- **w/o Both:** Both Transformer and GAT are removed.

All models were evaluated across nine diverse load and slicing scenarios described in §IV-B. The cumulative distribution function (CDF) of the rewards aggregated across all scenarios is shown in Fig. 6.

The results demonstrate that both the Transformer encoder and the GAT aggregator play critical roles in the overall performance of EExAPP. The full EExAPP model consistently achieves higher rewards than any of its ablated variants. Removing the Transformer module (**w/o Trans**) leads to significant performance degradation, highlighting its necessity in capturing temporal dependencies and contextual user relationships. Similarly, removing the GAT module (**w/o GAT**) results in a notable performance drop, demonstrating the value of coordinated dual-critic feedback. The combined removal of both components (**w/o Both**) yields the worst performance, approaching the lower bound of achievable rewards.

The QoS violation results in Fig. 7 further corroborate these observations. As traffic load increases, all model variants exhibit rising QoS violations, particularly as the number of slices grows. Nevertheless, EExAPP consistently maintains superior robustness with the lowest violation rates under all conditions. Notably, the performance gap between EExAPP and its ablated versions widens significantly under high traffic loads and complex slicing. This indicates that the Transformer and GAT modules are essential for maintaining system stability in saturated network environments.

D. Comprehensive Performance Comparison

To evaluate the performance of EExAPP against SOTA radio energy-saving solutions, we compare it with the following algorithms:

- **Kairos** [5]: An xApp-based single-actor–multi-critic DRL architecture. A centralized actor determines the delay allowance, while multiple distributional critics independently estimate energy savings and QoS compliance for each network slice.
- **O-RAN DRL** [22]: A DRL-based framework that optimizes the trade-off between UE throughput and energy consumption using various PPO and DQN models. For a fair comparison, we adopt the PPO-1 variant, identified as the top-performing model in [22].
- **SASC**: The single-actor–single-critic version of EExAPP, as detailed in §IV-A.

We adapted these baselines to align with our problem formulation and the reward function defined in §II-C. All

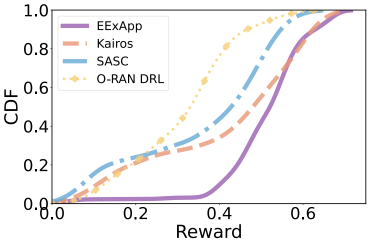

Fig. 8: Reward comparison of EExAPP and SOTA baselines.

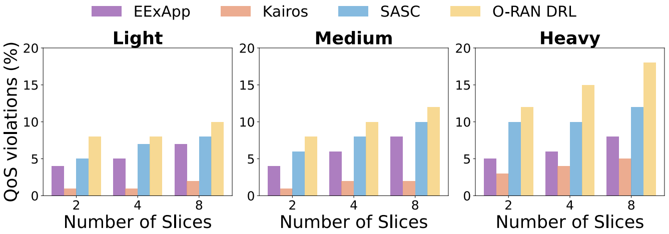

Fig. 9: QoS violation comparison of EExAPP and SOTA baselines.

experiments are conducted under the nine hybrid load and slicing scenarios outlined in §IV-B.

The results, summarized in Fig. 8 and Fig. 9, demonstrate that EExAPP consistently outperforms the baselines in balancing energy efficiency and QoS. Regarding reward distribution (Fig. 8), EExAPP achieves superior cumulative rewards compared to Kairos, SASC, and O-RAN DRL. In terms of QoS violations (Fig. 9), EExAPP maintains consistently low violation rates across all load conditions and slice counts, particularly under medium and heavy traffic.

Kairos achieves the lowest QoS violation rates overall, owing to its distributional critics that explicitly monitor QoS compliance. However, EExAPP surpasses Kairos in overall reward by identifying a more effective trade-off boundary. While Kairos adopts a conservative policy that restricts sleep opportunities to strictly minimize violations, EExAPP leverages the dual-actor architecture to safely extend sleep durations, thereby maximizing energy savings without engaging in excessive QoS violations. In contrast, both SASC and O-RAN DRL underperform, exhibiting lower rewards and higher violation rates. Their performance degrades under complex network conditions, such as high traffic loads or large slice counts, indicating limited robustness and adaptability.

V. Related Work

Radio Energy Saving in 5G: The energy-saving topic in RAN has been extensively investigated, which can be summarized into four main strategies: i) *time-domain strategies* that schedule idle periods via sleep modes, such as discontinuous transmission (DTX) and discontinuous reception (DRX) [23]–[25]; ii) *frequency-domain strategies* that reduce bandwidth usage or deactivate carriers, such as carrier aggregation [26], [27] and carrier shutdown [28]; iii) *power-domain strategies* that optimize power amplifier efficiency, such as transmit power adaptation [29], [30]; iv) *spatial-domain strategies* that employ dynamic antenna and transceiver activation/deactivation such as antenna element adaptation [31]–[34].

However, the efforts to integrate energy-saving techniques into O-RAN systems are still in their early stages. Existing efforts vary widely in their design scope. For instance, works like RLDFS [35] and CFMM [36] focus on function splitting, while BSvBS [37] and EEDRA [38] address resource allocation. Other studies, such as ScalO-RAN [39], investigate compute scaling. Despite this diversity, many approaches remain non-learning-based or rely on static heuristics. While effective in specific settings, such approaches often lack adaptability to dynamic traffic patterns and struggle to optimize long-term trade-offs, such as balancing energy versus QoS.

RL for Energy Optimization in O-RAN: RL-based approaches have recently gained attention within the O-RAN context. Early studies, including the aforementioned RLDFS [35], employed basic Q-learning and Sarsa to minimize energy consumption, while ES-xApp [40] applied DQN to switch off unused radio components. More recent advancements, such as Kairos [5] and O-RAN DRL [22], have extended the scope to advanced sleep mode (ASM) control and joint optimization of throughput, energy, and mobility robustness, leveraging Actor–Critic or PPO strategies. These works demonstrate RL's potential in managing complex trade-offs in O-RAN energy optimization.

Building on this progress, EExAPP advances RL-enabled energy saving in O-RAN through three key contributions: (i) Joint optimization of RU sleeping scheduling and DU resource slicing; (ii) Specialized dual-actor-critic framework to handle conflicting objectives under dynamic traffic; and (iii) Real-world validation via over-the-air experiments, moving beyond purely simulation-based evaluations.

VI. Conclusion

This paper proposed EExAPP, a DRL-based xApp designed for 5G O-RAN systems to jointly optimize RU sleep scheduling and DU resource slicing. By leveraging a dual-actor–dual-critic PPO framework, EExAPP effectively balances the trade-off between energy efficiency and QoS compliance, decoupling the optimization targets to specialized agents. The Transformer-based encoder enables scalable processing of dynamic UE populations, while the bipartite GAT module coordinates the distinct critics to support robust and adaptive policy updates. Extensive over-the-air experiments on a real-world testbed demonstrate that EExAPP achieves substantial reductions in RU energy consumption while maintaining QoS, consistently outperforming SOTA baselines under diverse network conditions. The authors have provided public access to their code at [41].

Acknowledgment

This work was supported in part by NTIA PWSCIF Award 26-60-IF010 and NSF Grant CNS-2312448.

References

[1] S. Han, S. Bian *et al.*, "Energy-efficient 5g for a greener future," *Nature Electronics*, vol. 3, no. 4, pp. 182–184, 2020.

[2] J. Huttunen, M. Pärssinen, T. Heikkilä, O. Salmela, J. Manner, and E. Pongracz, "Base station energy use in dense urban and suburban areas," *IEEE Access*, vol. 11, pp. 2863–2874, 2023.

[3] GSMA, "Mobile net zero, state of the industry on climate action 2022," *White Paper*, 2023.

[4] L. Kundu, X. Lin, and R. Gadiyar, "Toward energy efficient ran: From industry standards to trending practice," *IEEE Wireless Communications*, vol. 32, no. 1, pp. 36–43, 2025.

[5] J. Lozano, J. A. Ayala-Romero, A. Garcia-Saavedra, and X. Costa-Perez, "Kairos: Energy-efficient radio unit control for o-ran via advanced sleep modes," *arXiv preprint arXiv:2501.15853*, 2025.

[6] M. Polese, L. Bonati, S. D'oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[7] P. Yan, J. Lu, H. Zeng, and Y. T. Hou, "Near-real-time resource slicing for qos optimization in 5g o-ran using deep reinforcement learning," *arXiv preprint arXiv:2509.14343*, 2025.

[8] P. Yan, H. Zeng, and Y. T. Hou, "xdiff: Online diffusion model for collaborative inter-cell interference management in 5g o-ran," *IEEE Transactions on Networking*, 2025.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[10] M. Shafi, A. F. Molisch, P. J. Smith, T. Haustein, P. Zhu, P. De Silva, F. Tufvesson, A. Benjebbour, and G. Wunder, "5g: A tutorial overview of standards, trials, challenges, deployment, and practice," *IEEE journal on selected areas in communications*, vol. 35, no. 6, pp. 1201–1221, 2017.

[11] J. F. Santos, A. Huff, D. Campos, K. V. Cardoso, C. B. Both, and L. A. DaSilva, "Managing o-ran networks: xapp development from zero to hero," *IEEE Communications Surveys & Tutorials*, 2025.

[12] B. Brik, H. Chergui, L. Zanzi, F. Devoti, A. Ksentini, M. S. Siddiqui, X. Costa-Pèrez, and C. Verikoukis, "Explainable ai in 6g o-ran: A tutorial and survey on architecture, use cases, challenges, and future research," *IEEE Communications Surveys & Tutorials*, 2024.

[13] 3rd Generation Partnership Project (3GPP), *NR; NR and NG-RAN Overall Description; Stage-2*, 3GPP Std. 3GPP TS 38.300 version 17.2.0 Release 17, Sep. 2022.

[14] ——, *NR; NG-RAN; Architecture description*, 3GPP Std. 3GPP TS 38.401 version 17.2.0 Release 17, Sep. 2022.

[15] ——, *NR; Base Station (BS) radio transmission and reception*, 3GPP Std. 3GPP TS 38.104 version 17.9.0 Release 17, Apr. 2023.

[16] ——, *Study on New Radio (NR) to support non-terrestrial networks*, 3GPP Std. 3GPP TR 38.811 version 15.4.0 Release 15, Oct. 2020.

[17] 3GPP TSG RAN WG1, "NR Ad-Hoc#3: R1-1716574," 3rd Generation Partnership Project (3GPP) TSG RAN WG1 Meeting NR Ad-Hoc#3, R1-1716574, Dec. 2017.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[19] OpenAirInterface, "NR SA Tutorial – OAI CN5G," [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/blob/develop/doc/NR_SA_Tutorial_OAI_CN5G.md, 2015, accessed: June 18, 2015.

[20] ——, "OpenAirInterface 5G Project," [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/-/tree/develop, 2015, accessed: June 18, 2015.

[21] Mosaic5G, "flexRIC: A Flexible and Programmable RAN Intelligent Controller Platform," [Online]. Available: https://gitlab.eurecom.fr/mosaic5g/flexric, 2021, accessed: December 23, 2021.

[22] 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," in *2025 IEEE 22nd Consumer Communications & Networking Conference (CCNC)*. IEEE, 2025, pp. 1–6.

[23] E. M. de Santana, I. M. Guerreiro, L. R. Costa, A. Landström, and A. Simonsson, "Network performance evaluation of a sub-thz downlink system operating under dynamic dtx," *IEEE Transactions on Vehicular Technology*, 2025.

[24] N. Vatanian, G. W. O. da Costa, E. Roth-Mandutz, G. George, and N. Franchi, "Energy savings in 5g-advanced radio access networks: Downlink signaling adaptation," in *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 2024, pp. 1–7.

[25] M. Oikonomakou, A. Khlass, D. Laselva, M. Lauridsen, M. Deghel, and G. Bhatti, "A power consumption model and energy saving techniques for 5g-advanced base stations," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2023, pp. 605–610.

[26] F. Khoramnejad, R. Joda, A. B. Sediq, H. Abou-Zeid, R. Atawia, G. Boudreau, and M. Erol-Kantarci, "Delay-aware and energy-efficient carrier aggregation in 5g using double deep q-networks," *IEEE Transactions on Communications*, vol. 70, no. 10, pp. 6615–6629, 2022.

[27] Z. Wei, H. Liu, X. Yang, W. Jiang, H. Wu, X. Li, and Z. Feng, "Carrier aggregation enabled integrated sensing and communication signal design and processing," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 3, pp. 3580–3596, 2023.

[28] A. De Domenico, D. López-Pérez, W. Li, N. Piovesan, H. Bao, and X. Geng, "Modeling user transfer during dynamic carrier shutdown in green 5g networks," *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5536–5549, 2023.

[29] J. Lu, W. Feng, and D. Pu, "Resource allocation and offloading decisions of d2d collaborative uav-assisted mec systems," *KSII Transactions on Internet and Information Systems (TIIS)*, vol. 18, no. 1, pp. 211–232, 2024.

[30] Q. Wang, C. Qian, P. Yan, S. Zhang, and H. Zeng, "A batteryless wireless microphone using rf backscatter," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 4, pp. 1–18, 2025.

[31] M.-S. Van Nguyen, D.-T. Do, S. Al-Rubaye, S. Mumtaz, A. Al-Dulaimi, and O. A. Dobre, "Exploiting impacts of antenna selection and energy harvesting for massive network connectivity," *IEEE Transactions on Communications*, vol. 69, no. 11, pp. 7587–7602, 2021.

[32] D. Xu, A. Khalili, X. Yu, D. W. K. Ng, and R. Schober, "Integrated sensing and communication in distributed antenna networks," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2023, pp. 1457–1462.

[33] J. Lu, J. Nie, J. Li, W. Feng, Z. Xiong, D. Niyato, and W. Jiang, "Sic-stia-is: An interference management scheme for the uav-assisted heterogeneous network," in *ICC 2023-IEEE International Conference on Communications*. IEEE, 2023, pp. 672–678.

[34] J. Lu, J. Li, F. R. Yu, W. Jiang, and W. Feng, "Uav-assisted heterogeneous cloud radio access network with comprehensive interference management," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 1, pp. 843–859, 2023.

[35] T. Pamuklu, M. Erol-Kantarci, and C. Ersoy, "Reinforcement learning based dynamic function splitting in disaggregated green open rans," in *ICC 2021-IEEE International Conference on Communications*. IEEE, 2021, pp. 1–6.

[36] Ö. T. Demir, M. Masoudi, E. Björnson, and C. Cavdar, "Cell-free massive mimo in o-ran: Energy-aware joint orchestration of cloud, fronthaul, and radio resources," *IEEE Journal on Selected Areas in Communications*, vol. 42, no. 2, pp. 356–372, 2024.

[37] M. Kalntis and G. Iosifidis, "Energy-aware scheduling of virtualized base stations in o-ran with online learning," in *GLOBECOM 2022-2022 IEEE Global Communications Conference*. IEEE, 2022, pp. 6048–6054.

[38] H. Li, X. Tang, D. Zhai, R. Zhang, B. Li, H. Cao, N. Kumar, and A. Almogren, "Energy-efficient deployment and resource allocation for o-ran-enabled uav-assisted communication," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 3, pp. 1128–1140, 2024.

[39] S. Maxenti, S. D'Oro, L. Bonati, M. Polese, A. Capone, and T. Melodia, "Scalo-ran: Energy-aware network intelligence scaling in open ran," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2024, pp. 891–900.

[40] Q. Wang, S. Chetty, A. Al-Tahmeesschi, X. Liang, Y. Chu, and H. Ahmadi, "Energy saving in 6g o-ran using dqn-based xapp," in *2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*. IEEE, 2024, pp. 01–06.

[41] "Eexapp," [Online]. Available: https://github.com/EExApp/EExApp.git, accessed: Jul. 31, 2025.