LLM-Assisted Alpha-Fairness for 6 GHz Wi-Fi/NR-U Coexistence: An Agentic Orchestrator for Throughput, Energy, and SLA

Anonymous Author(s)
Affiliation
Address
email

Abstract

Unlicensed 6 GHz is becoming a primary workhorse for high-capacity access, with Wi-Fi and 5G NR-U competing for the same channels under listen-before-talk (LBT) rules. Operating in this regime requires decisions that jointly trade throughput, energy, and service-level objectives while remaining safe and auditable. We present an agentic controller that separates *policy* from *execution*. At the start of each scheduling epoch the agent summarizes telemetry (per-channel busy and baseline LBT failure; per-user CQI, backlog, latency, battery, priority, and power mode) and invokes a large language model (LLM) to propose a small set of interpretable knobs: a fairness index α, per-channel duty-cycle caps for Wi-Fi/NR-U, and class weights. A deterministic optimizer then enforces feasibility and computes an α-fair allocation that internalizes LBT losses and energy cost; malformed or unsafe policies are clamped and fall back to a rule baseline. In a 6 GHz simulator with two 160 MHz channels and mixed Wi-Fi/NR-U users, LLM-assisted policies consistently improve energy efficiency while keeping throughput competitive with a strong rule baseline. One LLM lowers total energy by 35.3% at modest throughput loss, and another attains the best overall trade-off, finishing with higher total bits ($+3.5\%$) and higher bits/J ($+12.2\%$) than the baseline. We release code, per-epoch logs, and plotting utilities to reproduce all figures and numbers, illustrating how transparent, policy-level LLM guidance can safely improve wireless coexistence.

1 Introduction

Unlicensed 6 GHz spectrum is rapidly becoming a workhorse for high-capacity wireless access, with Wi-Fi 6E/7 and 5G NR-U expected to coexist in the same channels Wang et al. [2022]. The resulting environment is shaped by LBT rules, bursty traffic, heterogeneous device batteries and priorities, and tight latency targets for interactive and safety-critical applications. Conventional schedulers are typically crafted around a fixed objective, such as sum-rate maximization or proportional fairness, and rely on static heuristics to avoid collisions. While effective in specific regimes, these designs struggle to adapt when operating conditions drift, and they provide limited knobs to navigate the three-way tension between throughput, energy, and service-level objectives Liu et al. [2025].

State-of-the-art solutions for managing spectrum coexistence between technologies like Wi-Fi and NR-U predominantly fall into two categories: static rule-based schedulers and complex learning-based systems like deep reinforcement learning (DRL) Kalahe-Wattege and Beltran [2024]Zhang et al. [2023b]. Rule-based schedulers are reliable and predictable but are fundamentally brittle; they are designed with fixed priorities, such as maximizing overall throughput, and struggle to dynamically adapt when network conditions and objectives change Kalahe-Wattege and Beltran [2024]. For example, they cannot easily pivot to prioritize energy efficiency or meet a specific user's urgent latency demand when the network becomes congested . On the other hand, while DRL agents can learn sophisticated policies to manage these trade-offs, they often operate as "black boxes," making their decisions difficult to interpret, audit, or trust in a production network Zhang et al. [2023b] . Furthermore, DRL systems require extensive, often impractical, online training cycles to converge on an effective policy.

Our paper solves this critical gap by introducing an agentic controller that separates high-level reasoning from low-level execution. It leverages a LLM to interpret the complex, multi-faceted network state and propose a simple, human-understandable set of policy "knobs"—such as the fairness level and duty-cycle caps Hao et al. [2024] Wang et al. [2024]. This approach avoids the rigidity of fixed rules and the opacity of DRL, providing a solution that is simultaneously adaptive, auditable, and capable of intelligently balancing the conflicting objectives of throughput, energy efficiency, and service level agreements (SLA) on a dynamic, per-epoch basis.

At the start of each scheduling epoch, the agent encodes the observable state—per-channel busy levels and baseline LBT failure probabilities for both Wi-Fi and NR-U, together with per-user CQI, backlog, latency target, battery state, task priority, and power mode—into a compact JSON summary. A high-level policy then chooses a small set of interpretable knobs: a fairness index, per-channel duty-cycle caps for each stack, and priority weights across traffic classes. This policy can be instantiated either as a deterministic rule or as a LLM prompted to reason about energy-aware, latency-aware spectrum sharing.

We evaluate the agent in a 6 GHz simulator with two 160 MHz channels and mixed Wi-Fi/NR-U populations. Across moderate and high offered loads, LLM-assisted policies consistently reduce cumulative energy and improve energy efficiency (bits/J) relative to a strong rule baseline, while maintaining competitive or superior cumulative throughput. In a representative scenario, one LLM variant lowers total energy by more than a third while the other achieves the best overall trade-off, finishing with higher total bits and the highest bits/J.

The remainder of this paper is structured as follows. In Section II, we detail the system model and formulate the multi-objective resource allocation problem. Section III presents the design of the LLM-assisted agentic controller. We describe our simulation environment and the baseline rule-based scheduler used for comparison in Section IV. In Section V, we present a comprehensive evaluation of our proposed system, analyzing its performance across key metrics of network throughput, energy efficiency, and SLA satisfaction against the baseline. Finally, we conclude the paper in Section VI.

2 System Model and Problem Formulation

2.1 System Model

We consider a 6 GHz unlicensed band shared by Wi-Fi and 5G NR-U. Time is slotted into fixed scheduling epochs of length Δ seconds (default $\Delta = 0.1$), and all control decisions are recomputed once per epoch $t = 1, 2, \ldots$ The system comprises a finite set of channels $\mathcal{C}$ (two 160 MHz channels in the default configuration) and two coexisting technology stacks $\mathcal{T} = \{\text{Wi-Fi}, \text{NR-U}\}$. Each channel $c \in \mathcal{C}$ has bandwidth B_c and is characterized by exogenous LBT conditions that vary slowly over time. Specifically, for each stack $t \in \mathcal{T}$ and channel c, we track a sensed busy fraction $b_{t,c}(t) \in [0, 1]$ and a baseline LBT failure probability $f_{t,c}(t) \in [0, 1]$, both subjected to small random jitter between epochs to emulate environmental dynamics. These quantities are taken as inputs to the scheduler and are observable before each decision.

Users are partitioned by stack, $\mathcal{U} = \mathcal{U}_{\text{Wi-Fi}} \cup \mathcal{U}_{\text{NR-U}}$. A user $i \in \mathcal{U}$ is described at epoch t by a channel quality indicator $q_i(t) \in \{0, \ldots, 15\}$, a battery state $B_i(t) \in [0, 1]$, a queue backlog $Q_i(t)$ in bits, a latency target D_i in milliseconds, a task priority class $k_i \in \{\text{emergency, high, normal, bulk}\}$, and a discrete power mode $p_i \in \{\text{low, med, high}\}$. New traffic arrivals $A_i(t)$ are injected into the queue each epoch according to a truncated Gaussian process with a configurable mean; the queue dynamics obey

$$Q_i(t{+}1) \;=\; \max\big\{\, 0,\; Q_i(t) + A_i(t) - S_i(t) \,\big\}, \tag{1}$$

where $S_i(t)$ denotes the number of bits served to user i during epoch t Wang et al. [2020]. The simulation exposes these elements as first-class state variables that are evolved by a lightweight environment model prior to each scheduling step.

Physical-layer throughput is modeled via a CQI-to-spectral-efficiency mapping. For user i on channel c, the spectral efficiency (bits/s/Hz) is

$$s_{i,c}(t) \;=\; \mathrm{SE}\big(q_i(t)\big)\cdot \eta_{p_i}, \tag{2}$$

where $\mathrm{SE}(\cdot)$ is a standard 16-level table and η_{p_i} scales the efficiency under the user's power mode. If $\tau_{i,c}(t) \in [0,1]$ is the airtime fraction allotted to i on c during epoch t, then the pre-LBT raw rate is

$$r_{i,c}(t) \;=\; s_{i,c}(t)\, B_c\, \tau_{i,c}(t). \tag{3}$$

Shared-channel contention and regulatory LBT constraints induce a stack–channel loss that we capture with a smooth proxy. Let $\tau_{t,c}(t) = \sum_{i\in\mathcal{U}_t} \tau_{i,c}(t)$ be the aggregate airtime of stack t on channel c in epoch t. The loss fraction applied uniformly to all users of (t,c) is

$$\ell_{t,c}(t) \;=\; \min\Big\{\, 0.95,\; f_{t,c}(t) \;+\; 0.6\,\tau_{t,c}(t)\, b_{t,c}(t) \;+\; 0.2\,\big(\tau_{t,c}(t) + b_{t,c}(t) - 1\big)_+ \Big\}, \tag{4}$$

with $(x)_+ = \max\{x, 0\}$. The post-LBT goodput for user i on c is then

$$g_{i,c}(t) \;=\; r_{i,c}(t)\,\big(1 - \ell_{t_i,c}(t)\big), \qquad S_i(t) \;=\; \Delta \sum_{c\in\mathcal{C}} g_{i,c}(t). \tag{5}$$

Energy is modeled through a per-bit cost that depends on the power mode and the achieved spectral efficiency. Writing $P(p_i)$ for the mode-dependent transmit power and $e_{i,c}(t) = P(p_i)/s_{i,c}(t)$ for joules per bit, the per-epoch energy is

$$E_i(t) \;=\; \sum_{c\in\mathcal{C}} e_{i,c}(t)\, S_i(t). \tag{6}$$

The above link, loss, and energy models mirror the implementation used by our simulator, where (4) is realized as a differentiable proxy to capture LBT-driven collisions/backoff at the stack–channel granularity.

Service-level latency is expressed as an instantaneous rate requirement. Given backlog $Q_i(t)$ and latency target D_i, the minimum rate that avoids deadline slippage within epoch t is

$$\rho_i(t) \;=\; \min\Big\{ \tfrac{Q_i(t)}{\Delta},\; \tfrac{Q_i(t)}{D_i/1000} \Big\}, \tag{7}$$

and an SLA hit is registered whenever $\sum_c g_{i,c}(t) \ge \rho_i(t)$. The policy layer that precedes optimization provides three high-level knobs per epoch: a fairness index $\alpha \in \{0,1,2\}$, per-channel duty-cycle caps $u_c^{\text{Wi-Fi}}, u_c^{\text{NR-U}} \in [0,1]$, and priority weights $\{w_k\}_{k\in\{\text{emergency,high,normal,bulk}\}}$. Caps are constrained by sensed load through a headroom rule of the form $u_c^t \le 1 - \gamma\, b_{t,c}(t)$ with $\gamma \in (0,1)$, ensuring that aggregate scheduled airtime remains feasible under exogenous activity. These quantities are produced either by a deterministic rule or by a LLM from a compact JSON state summary, and are clamped to safe ranges before optimization.

2.2 Problem Formulation

Let $x_{i,c}(t) \in \{0,1\}$ indicate whether user i is scheduled on channel c during epoch t. Each user is bound to at most one channel per epoch, $\sum_c x_{i,c}(t) \le 1$, and receives a nonnegative airtime $\tau_{i,c}(t) \in [0,1]$ that is consistent with the assignment via $\tau_{i,c}(t) \le x_{i,c}(t)$. Per-stack duty caps provided by the policy enforce

$$\sum_{i\in\mathcal{U}_{\text{Wi-Fi}}} \tau_{i,c}(t) \;\le\; u_c^{\text{Wi-Fi}}, \qquad \sum_{i\in\mathcal{U}_{\text{NR-U}}} \tau_{i,c}(t) \;\le\; u_c^{\text{NR-U}}, \quad \forall c \in \mathcal{C}, \tag{8}$$

and interact with LBT through (4)–(5). The per-user post-LBT rate in epoch t is $x_i(t) = \sum_c g_{i,c}(t)$, and the per-epoch energy is $E_i(t)$ in (6). A standard weighted α-fair utility is adopted Wang and Zhou [2019],

$$U_\alpha(x) \;=\; \begin{cases} \log x, & \alpha = 1, \\ \dfrac{x^{\,1-\alpha}}{1-\alpha}, & \alpha \ne 1, \end{cases} \qquad x > 0, \tag{9}$$

and each user i is endowed with a base weight $\theta_i = \dfrac{w_{k_i}}{0.5 + \beta(B_i)}$ that increases with priority and penalizes low battery via a monotone function $\beta(\cdot)$. The single-epoch allocation problem can then be stated as

$$\begin{aligned} \max_{\{x_{i,c},\,\tau_{i,c}\}} \quad & \sum_{i\in\mathcal{U}} \theta_i \, U_\alpha\!\left(\sum_{c\in\mathcal{C}} g_{i,c}(t)\right) \;-\; \lambda \sum_{i\in\mathcal{U}} E_i(t) \\ \text{s.t.} \quad & \text{link and LBT coupling in (3)–(5),} \\ & \text{caps in (8)}, \quad \tau_{i,c}(t) \in [0,1],\; x_{i,c}(t) \in \{0,1\},\; \sum_c x_{i,c}(t) \le 1. \end{aligned} \tag{10}$$

Here $\lambda \ge 0$ trades off energy against throughput in the objective; in our implementation the energy term is effectively captured by the interaction of (6) with the weights θ_i and per-duty utility densities used during channel selection, while the reported α-fair utility remains the primary figure of merit. Hard per-epoch latency guarantees may be included as $\sum_c g_{i,c}(t) \ge \rho_i(t)$ for selected flows, but we favor a pragmatic approach in which minimum "urgent" grants are applied procedurally before the proportional α-fair split on the remaining budget—an approach that preserves tractability while honoring latency-sensitive traffic.

Problem (10) is solved independently at each epoch using policy-provided knobs $(\alpha, \{u_c^t\}, \{w_k\})$, yielding the airtime variables $\{\tau_{i,c}(t)\}$, post-LBT rates $\{g_{i,c}(t)\}$, and per-epoch metrics (throughput, energy, and SLA hit rate) that feed back into the next epoch via (1). This separation between a high-level, possibly LLM-driven policy and a verifiable optimizer enables safe orchestration: policy outputs are clamped to feasible ranges, and the optimizer enforces hard constraints at execution time.

3 Method: Agent Architecture

This work adopts an agentic architecture that separates high-level spectrum *policy* from low-level *optimization*. The policy layer proposes a fairness index α, per-channel duty-cycle caps for Wi-Fi and NR-U, and task-class priority weights from observable telemetry; the optimizer then enforces feasibility and computes an α-fair allocation while accounting for LBT losses, energy, and latency. The separation ensures that the intelligence responsible for strategic trade-offs is modular and replaceable (rule-based or LLM-driven), whereas the executor is verifiable and deterministic. The implementation follows this design in a single-epoch loop with environmental evolution between epochs.

3.1 Telemetry encoding and policy interface

At the beginning of each epoch of length Δ, the environment exposes a compact JSON state containing channel descriptors (bandwidth B_c, sensed busy $b_{t,c}$, and baseline LBT failure $f_{t,c}$ for each stack t on channel c) and user descriptors (CQI q_i, battery level B_i, backlog Q_i, latency target D_i, task priority k_i, and power mode p_i). The function `build_state_json` produces this state and adds lightweight hints such as candidate α values. A policy consumes this JSON and returns three objects: an $\alpha \in \{0, 1, 2\}$; duty caps $\{u_c^{\text{Wi-Fi}}, u_c^{\text{NR-U}}\}$; and priority weights $\{w_k\}$ for classes `emergency/high/normal/bulk`. The project ships two variants of the policy interface: a rule baseline that biases caps toward the busier stack while reserving headroom, and an LLM policy that emits a JSON policy either via a chat completion in JSON mode or via a Responses API call constrained by a JSON Schema; in both cases the returned values are coerced to safe ranges prior to optimization.

3.2 Safety, feasibility, and fallbacks

The fairness index is restricted to $\{0, 1, 2\}$; duty caps are clipped to $[0, 1]$ and to a busy-aware headroom $u_c^t \le 1 - \gamma\, b_{t,c}$ (with $\gamma \in (0, 1)$ implemented as 0.5 in the code); and priority weights are confined to a bounded interval $[0.1, 10]$. If the LLM call fails or returns malformed JSON, the system falls back to the deterministic rule policy, ensuring that every epoch yields a feasible control. Optional textual rationales from the LLM are preserved for auditability but do not affect the optimizer.

3.3 Epoch solver: two-stage optimization

Given the policy knobs, the optimizer solves the epoch using a two-stage scheme tailored to coexistence with shared LBT losses.

The first stage assigns each user to one channel by maximizing a *utility density* computed under a small probe airtime τ_0, which approximates the value per unit of duty cycle while internalizing energy and latency costs. For user i on channel c, the pre-loss rate is $r_{i,c} = s_{i,c} B_c \tau_0$, where $s_{i,c}$ is the CQI- and power-mode–dependent spectral efficiency. The stack-channel loss is modeled by a smooth proxy:

$$\ell_{t,c} \;=\; \min\{0.95,\; f_{t,c} + 0.6\,\tau_{t,c}\, b_{t,c} + 0.2\,(\tau_{t,c} + b_{t,c} - 1)_+\}\,, \tag{11}$$

with $\tau_{t,c}$ being the aggregate duty of stack t on channel c. The probe goodput is $g_{i,c} = r_{i,c}(1 - \ell_{t_i,c})$, and the corresponding energy consumed during the probe is $E^{\text{probe}}_{i,c}$. The assignment score reflects a direct trade-off between this goodput and the energy cost, defined as:

$$\Phi_{i,c} \;=\; \frac{1}{\tau_0}\left(\underbrace{w_{k_i}\cdot\frac{g_{i,c}}{10^6}\cdot w_{\text{lat}}(D_i)}_{\text{Reward: Weighted Goodput (Mbps)}} \;-\; \underbrace{\beta(B_i)\cdot E^{\text{probe}}_{i,c}}_{\text{Cost: Battery-Scaled Energy}} \right), \tag{12}$$

where w_{k_i} is the user's priority weight, $w_{\text{lat}}(D_i)$ is a multiplier that increases the reward for latency-sensitive users (e.g., those with $D_i \leq 50$ ms), and $\beta(B_i)$ is a penalty factor that increases as battery level B_i decreases. The channel with the maximal score $\Phi_{i,c}$ is chosen for user i. This stage has complexity $O(|\mathcal{U}|\,|\mathcal{C}|)$ and captures the primary cross-channel trade-offs.

The second stage performs within-channel allocation under the duty caps provided by the policy. For each channel and stack, the available duty budget $\leq u^t_c$ is split in two passes. A first pass grants *urgent minimums* to latency-critical or high-priority users, allocating the minimal duty required to meet their instantaneous rate requirement $\rho_i = \min\{Q_i/\Delta,\, Q_i/(D_i/1000)\}$. A second pass distributes the residual budget via a weighted α-fair rule, where user i receives a portion of the airtime proportional to:

$$\omega_i \;=\; \left(\frac{w_{k_i}}{0.5 + \beta(B_i)}\right) \cdot \big(\text{served}_i + \varepsilon\big)^{-\alpha}. \tag{13}$$

Here, served_i is the service a user has already received within the epoch (from the urgent grant) and $\varepsilon > 0$ is a small constant to ensure stability. After all duties are provisionally assigned, the final aggregate duties are used to recompute the stack-channel losses $\ell_{t,c}$, from which the final per-user goodput and energy consumption are determined.

3.4 SLA evaluation, queue update, and logging

Following allocation, the achieved per-user rate $\sum_c g_{i,c}$ is compared against ρ_i to determine SLA hits in the current epoch. The served bits are subtracted from backlogs to update Q_i for the next epoch, ensuring tight coupling between control and traffic dynamics. In multi-epoch mode, the driver `run_multi_epoch` evolves channels and baselines with small Gaussian jitters, injects arrivals with a configurable mean, repeatedly invokes the epoch solver.

3.5 LLM-Assisted Decision Making

At the beginning of each epoch, the agent summarizes the observable telemetry into a compact JSON state that includes per-channel descriptors (bandwidth, sensed busy fractions, and baseline LBT failure rates for each stack) and per-user descriptors (CQI, battery level, backlog, latency target, task priority, and power mode). This serialization, produced by `build_state_json`, is passed to a large language model that proposes high-level control knobs: a fairness index $\alpha \in \{0, 1, 2\}$, per-channel duty-cycle caps for Wi-Fi and NR-U, and task-class priority weights. We implement two invocation modes to ensure robustness: a chat-completions call constrained to JSON output, and a Responses-API call that enforces a JSON Schema with strict types and bounds. In both modes the model is prompted as a spectrum policy orchestrator and asked to trade off latency, energy, and fairness while avoiding per-user micromanagement. The LLM may return brief rationales, which are logged for audit but are not used downstream in optimization.

The raw policy is never executed directly. Instead, `coerce_policy_from_llm` enforces feasibility and safety by clamping α to $\{0, 1, 2\}$, projecting duty caps to $[0, 1]$ and to a busy-aware headroom of the form $u_c^t \le 1 - \gamma\, b_{t,c}$ (with $\gamma \in (0, 1)$), and restricting priority weights to a bounded interval that prevents extreme allocations. Any parsing failure, schema violation, or out-of-range proposal triggers a deterministic rule fallback that biases caps toward the empirically busier stack while reserving headroom, guaranteeing that every epoch yields a valid policy even under LLM faults.

Given the sanitized knobs $(\alpha, \{u_c^t\}_{c,t}, \{w_k\}_k)$, the optimizer solves the epoch in two stages. First, it assigns each user to a single channel by maximizing a probe-time utility density that internalizes post-LBT goodput, energy cost, deadline pressure, and priority/battery weights. Second, within each channel and stack, it grants minimal duties to satisfy urgent latency targets and then allocates the remaining budget according to a weighted α-fair rule. After duties are finalized, stack–channel LBT losses are realized and per-user goodputs and energies are computed. The selected α and per-epoch metrics (throughput, energy, SLA hit rate) are recorded, and the agent advances to the next epoch with updated queues. This integration makes the LLM responsible only for transparent, high-level decisions, while a verifiable executor enforces hard constraints at run time.

The core of the interaction between the agent and the LLM is a structured JSON object that serves as the complete state representation provided at the start of each scheduling epoch. The JSON object is organized into two primary keys: `channels` and `users`. `channels`: An array of objects detailing the physical state and contention level of each frequency channel. For each channel, we provide its bandwidth (`bw_mhz`), the measured busy-time contributed by Wi-Fi and NR-U (`busy_wifi`, `busy_nru`), and the baseline LBT failure probability. `users`: An array of objects representing the state of each active user. For each user, the prompt specifies their technology (`tech`), channel quality indicator (`cqi`), data backlog in bits (`backlog_bits`), remaining time to meet their SLA deadline (`deadline_s`), battery percentage (`battery_pct`), and assigned service priority class (`priority`).

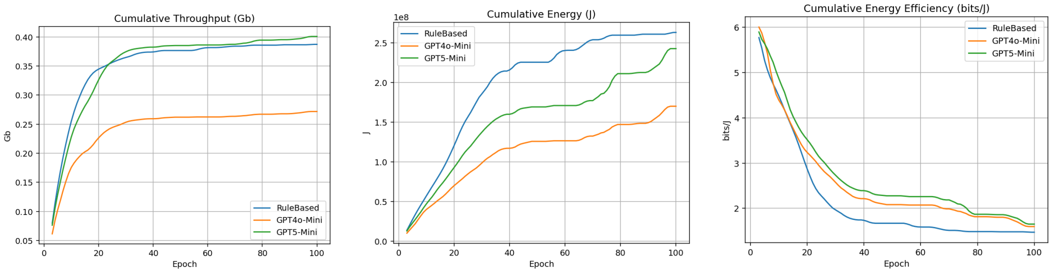
4 Experimental Setup and Results

4.1 Experimental Setup

We evaluate the agent in a simulated 6 GHz unlicensed band with two 160 MHz channels shared by Wi-Fi and NR-U. Each experiment spans $T{=}100$ scheduling epochs of length $\Delta{=}0.1$ s (total horizon 10 s) Ghosh [2023]. The default user population includes 16 Wi-Fi and 12 NR-U stations with heterogeneous channel quality indicators (CQI), battery levels, queue backlogs, latency targets, task priorities, and power modes. Channel descriptors include sensed busy fractions and baseline LBT failure probabilities for both stacks, all subject to small Gaussian jitter between epochs to emulate environmental dynamics. Traffic arrivals are injected every epoch as truncated Gaussians, and queue evolution follows the standard Lindley recursion. The simulator exposes these elements as first-class state variables (User, Channel, Env) and advances them via `step_env` before each decision; the single-epoch solver is called from `one_epoch_allocate`, and multi-epoch orchestration from `run_multi_epoch`. The code also logs per-epoch throughput (served bits), energy (J), SLA hit rate, and the fairness index α. The code is available at: https://github.com/claudwq/LLM-Assisted-Alpha-Fairness-for-6-GHz-Wi-Fi-NR-U-Coexistence.git

The policy layer is either rule-based or LLM-driven. The rule baseline computes per-channel duty-cycle caps for the two stacks, reserving headroom as a function of sensed busy, and then chooses the best-throughput $\alpha \in \{0, 1, 2\}$ each epoch (“benevolent” baseline). The LLM policy sees the same telemetry as a compact JSON, selects a single α, per-channel caps, and class weights, and is then clamped by `coerce_policy_from_llm` for safety before the optimizer is invoked. The optimizer itself is deterministic: it assigns a single channel per user via a utility-density score that internalizes post-LBT goodput, energy cost, and deadline pressure, and then performs a within-channel weighted α-fair split under the cap constraints, realizing LBT loss and energy using the final aggregate duties. The complete path and data schema are implemented in `llm_spectrum_agent_fairness.py` and its LLM interface variant.

We compare three methods: **RuleBased** (no LLM, benevolent α), **GPT4o-Mini** (LLM-assisted policy), and **GPT5-Mini** (LLM-assisted policy) Zhang et al. [2023a]. All experiments use seed 2025 and the default arrival and jitter settings in the code unless otherwise noted.



(a) Throughput

(b) Energy

(c) Energy Efficiency

Figure 1: Moderate load (40 Mb/s) results.

5 Results Analysis

We report results for three policies—*RuleBased* (benevolent $\alpha \in \{0, 1, 2\}$ chosen per epoch), *GPT4o-Mini*, and *GPT5-Mini*—over $T=100$ epochs with $\Delta=0.1$ s. Metrics are computed directly from the simulator logs: per-epoch served bits, energy (J), and SLA hit rate; we plot cumulative throughput (Gb), cumulative energy (J), and cumulative energy efficiency (bits/J).

5.1 Moderate Offered Load (40 Mb/s)

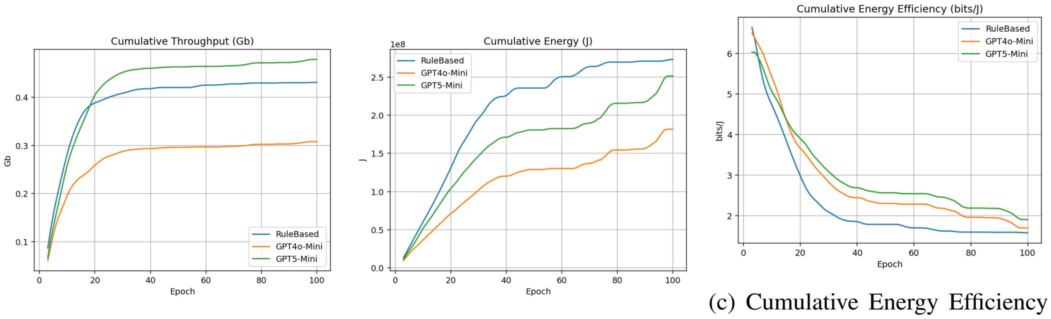
Under the moderate arrival rate, cumulative throughput rises rapidly during the first 20–30 epochs as the scheduler drains initial backlogs, then transitions to an arrival-limited regime in which curves flatten. In this setting, *GPT5-Mini* ultimately surpasses the baseline in total delivered bits while spending less energy, whereas *GPT4o-Mini* attains the lowest energy but at the cost of reduced throughput. The cumulative energy plots show that *RuleBased* expends the most energy across the horizon; this aligns with its tendency to choose $\alpha=0$ and to push higher duties into collision-prone regions. The cumulative energy-efficiency trajectories confirm the advantage of LLM-assisted control: both LLM policies maintain higher bits/J than the rule baseline for most of the horizon, with *GPT5-Mini* finishing highest. Intuitively, the LLMs propose headroom-aware caps and a fairness regime closer to $\alpha=1$, which lowers collision losses without starving progress, yielding better energy efficiency at comparable or higher cumulative bits.

5.2 High Offered Load (150 Mb/s)

With higher offered traffic the system remains service-limited for longer, and the differences between policies become more pronounced. The throughput curves indicate that *GPT5-Mini* sustains the fastest cumulative growth and finishes with the highest total bits, while *GPT4o-Mini* again trades some throughput for substantial energy savings. Total energy consumption is highest for the rule baseline across the entire horizon, reflecting aggressive duty usage that amplifies LBT loss; both LLM policies keep cumulative energy lower, and *GPT5-Mini* delivers a favorable balance of bits and joules. The energy-efficiency curves mirror these trends: LLM-assisted control dominates the baseline throughout most of the run, with *GPT5-Mini* providing the strongest long-horizon bits/J and *GPT4o-Mini* offering the best energy containment when energy is the primary objective.

5.3 Interpretation and Takeaways

Across both load regimes, LLM-assisted policies consistently lower cumulative energy and improve bits/J relative to the rule baseline, while *GPT5-Mini* achieves the best overall trade-off by pairing strong throughput with reduced energy. The qualitative shape of the curves is consistent with the agent's design: LLM-proposed duty caps, combined with a fairness choice nearer to proportional fairness, keep the system away from congestion-dominated operating points where additional duty yields little goodput but incurs substantial energy. In the moderate-load setting, backlogs drain by mid-horizon and per-epoch throughput approaches zero for all methods; the cumulative differences observed up to that point therefore reflect more judicious early-phase decisions by the LLM policies. In the high-load setting, where the system remains busy, the LLM advantage persists throughout the horizon, indicating better long-run operating points under sustained traffic.



(a) Cumulative Throughput (Gb)

(b) Cumulative Energy (J)

(c) Cumulative Energy Efficiency (bits/J)

Figure 2: High offered load (150 Mb/s) over 100 epochs. From left to right: cumulative throughput, cumulative energy, and cumulative energy efficiency.

6 Conclusion

This paper introduced an LLM-assisted spectrum agent for Wi-Fi/NR-U coexistence in the 6 GHz band. The core design cleanly separates high-level reasoning from verifiable execution: the policy layer—instantiated by a rule or by an LLM—chooses an α-fairness regime, per-channel duty caps, and class weights from compact telemetry, while a deterministic optimizer enforces hard constraints and realizes post-LBT goodput and energy. This interface makes the role of the LLM transparent and auditable and guarantees safe control through clamping and rule fallback. Across moderate and high offered loads, experiments show that LLM-assisted policies reduce cumulative energy and raise energy efficiency (bits/J) relative to a benevolent rule baseline, while maintaining competitive or superior cumulative throughput. The gains are most pronounced in the early and mid horizon, where headroom-aware caps and fairness choices near proportional fairness keep the system away from collision-dominated operating points. In a representative 100-epoch scenario, one LLM reduces total energy by more than a third, and another achieves the best overall trade-off with higher total bits and the highest bits/J among all methods.

References

Monisha Ghosh. Evolution of sharing in 6 ghz. *IEEE Wireless Communications*, 30(5):4–5, 2023. doi: 10.1109/MWC.2023.10325444.

Nan Hao, Yuangang Li, Kecheng Liu, Songtao Liu, Yingzhou Lu, Bohao Xu, Chenhao Li, Jintai Chen, Ling Yue, Tianfan Fu, et al. Artificial intelligence-aided digital twin design: A systematic review. 2024.

Rasika Nilaweera Kalahe-Wattege and Fernando Beltran. Enhancing throughput for 5g nr-u and wifi networks in 6ghz shared-spectrum. In *2024 International Symposium on Networks, Computers and Communications (ISNCC)*, pages 1–6, 2024. doi: 10.1109/ISNCC62547.2024.10759054.

Sicheng Liu, Qun Wang, Zhuwei Qin, Weishan Zhang, Jingyi Wang, and Xiang Ma. Irs assisted decentralized learning for wideband spectrum sensing. *arXiv preprint arXiv:2504.01344*, 2025.

Qun Wang and Fuhui Zhou. Fair resource allocation in an mec-enabled ultra-dense iot network with noma. In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6, 2019. doi: 10.1109/ICCW.2019.8757173.

Qun Wang, Le Thanh Tan, Rose Qingyang Hu, and Yi Qian. Hierarchical energy-efficient mobile-edge computing in iot networks. *IEEE Internet of Things Journal*, 7(12):11626–11639, 2020. doi: 10.1109/JIOT.2020.3000193.

Qun Wang, Haijian Sun, Rose Qingyang Hu, and Arupjyoti Bhuyan. When machine learning meets spectrum sharing security: Methodologies and challenges. *IEEE Open Journal of the Communications Society*, 3:176–208, 2022. doi: 10.1109/OJCOMS.2022.3146364.

Yue Wang, Tianfan Fu, Yinlong Xu, Zihan Ma, Hongxia Xu, Bang Du, Yingzhou Lu, Honghao Gao, Jian Wu, and Jintai Chen. Twin-gpt: Digital twins for clinical trials via large language model. *ACM Trans. Multimedia Comput. Commun. Appl.*, July 2024. ISSN 1551-6857. doi: 10.1145/3674838. URL `https://doi.org/10.1145/3674838`. Just Accepted.

Chaoning Zhang, Chenshuang Zhang, Sheng Zheng, Yu Qiao, Chenghao Li, Mengchun Zhang, Sumit Kumar Dam, Chu Myaet Thwal, Ye Lin Tun, Le Luang Huy, et al. A complete survey on generative ai (aigc): Is chatgpt from gpt-4 to gpt-5 all you need? *arXiv preprint arXiv:2303.11717*, 2023a.

Xiang Zhang, Arupjyoti Bhuyan, Sneha Kumar Kasera, and Mingyue Ji. Distributed power allocation for 6-ghz unlicensed spectrum sharing via multi-agent deep reinforcement learning. In *2023 IEEE International Conference on Industrial Technology (ICIT)*, pages 1–6, 2023b. doi: 10.1109/ICIT58465.2023.10143125.

Responsible AI Statement

This work adheres to the NeurIPS Code of Ethics. The agent is explicitly designed so that generative components never execute free-form actions: the large language model (LLM) proposes only a small set of interpretable, bounded policy knobs (fairness index α, per–channel duty caps, and priority weights) from a compact, non-personal telemetry summary. All proposals are validated against a JSON schema and then clamped to safety ranges, including busy-aware headroom on duty caps; malformed or out-of-range outputs trigger a deterministic rule fallback. Execution is handled by a verifiable optimizer that enforces constraints at run time and records decisions for audit. The simulator contains no personally identifiable information; it uses synthetic traffic and environment jitter. Potential negative impacts include unsafe spectrum use, unfair service to specific traffic classes, or excessive energy consumption if the policy is misconfigured. To mitigate these risks, we (i) restrict the LLM's authority to high-level proposals with hard constraints; (ii) log every epoch's knobs and outcomes for traceability; (iii) provide conservative defaults and seed control; (iv) discuss limits of our LBT-loss proxy and fairness settings; and (v) refrain from any real-world transmission or device control in this study. Any future deployment must include regulatory compliance checks (e.g., LBT requirements), site-specific validation, and monitoring for distribution shift, with the rule policy available as a safe fallback.

Reproducibility Statement

We provide an anonymized artifact containing code, configuration files, and scripts to regenerate every figure and table. The simulator exposes fixed seeds and prints the exact run configuration; the optimizer is deterministic given inputs. For the main results, we include per-epoch CSV logs (`served_bits`, `energy`, `sla_hit_rate`, `alpha_used`) produced by `llm_spectrum_agent_fairness.py` in multi-epoch mode, and a plotting script (`plot_three_runs_overlay.py`) that creates all figures and a summary CSV. The artifact documents the Python version and library hashes, lists command lines to reproduce the 40 Mb/s and 150 Mb/s experiments, and specifies default hyperparameters (epoch length, number of epochs, user counts, arrival process, jitter magnitudes, seeds, and model identifiers masked for double-blind review). Because the LLM only emits bounded numeric knobs and the executor is deterministic, results are stable across runs with the same seed; for completeness we also report partial cumulatives (25/50 epochs) and provide the raw CSVs used to render each plot.

AI Contribution Disclosure Checklist

AI Contribution Disclosure

This work uses large language models (LLMs) as *policy planners* and as assistants for writing and code refactoring. Model identities are anonymized as *LLM-A* and *LLM-B* for double-blind review. The execution engine that applies spectrum decisions is deterministic and constraint-enforcing; it does not depend on generative outputs beyond a small set of bounded numeric knobs.

Roles across the research lifecycle. Problem framing, system modeling, and the design of the policy–optimizer split were led by the authors. LLMs assisted ideation by suggesting alternative fairness/energy/latency trade-offs and by helping refine prose and notation. Simulator and optimizer implementations were authored by the researchers; LLMs were used to draft code skeletons, to propose interface shapes (e.g., JSON fields for policy knobs), and to streamline plotting and CSV post-processing. For spectrum control itself, the LLMs served as policy planners: given a compact telemetry summary, they proposed a fairness index α, per-channel duty caps for Wi-Fi/NR-U, and traffic-class weights. Authors retained authority over safety and feasibility: proposals were parsed under a strict schema, clamped to allowable ranges, and replaced by a rule policy upon any violation or parsing failure. Experiments, ablations, and quantitative analysis were specified and verified by the authors; LLMs assisted with script creation and draft figure captions. The manuscript's first drafts for several sections were produced with LLM assistance and then edited for technical accuracy and clarity by the authors.

Safeguards and limits on AI authority. Generative models do not issue low-level commands or modify the executor. Their authority is restricted to high-level, interpretable knobs: a discrete $\alpha \in \{0, 1, 2\}$, duty-cycle caps projected to $[0, 1]$ and further limited by busy-aware headroom $u_c^t \leq 1 - \gamma\, b_{t,c}$, and bounded priority weights. Any malformed JSON, schema mismatch, or out-of-range value triggers an automatic fallback to a deterministic rule policy. At run time, a verifiable optimizer enforces constraints, realizes LBT losses from the final aggregate duties, and computes energy from mode-dependent per-bit costs. The system logs the chosen knobs and resulting metrics every epoch, ensuring traceability and audit.

Data, privacy, and safety considerations. All experiments are simulation-based with synthetic traffic; no personal data are used. The work does not transmit over real radios or control physical infrastructure. Potential risks include unfair service to certain traffic classes or excessive energy usage if policy knobs were mis-set; these are mitigated by hard clamping, conservative defaults, per-epoch logging, and a safe rule fallback. Any future deployment would require regulatory compliance checks (e.g., LBT rules), site-specific validation, monitoring for distribution shift, and human oversight.

Transparency and artifacts. We provide an anonymized artifact with source code, prompts, schema definitions, exact commands, seeds, per-epoch CSV logs, and plotting scripts. All figures and numbers in the paper are generated programmatically from these logs; LLMs did not fabricate results. The division of responsibilities between AI systems and humans is thus transparent, auditable, and reproducible.