CAPS: Link-Aware Reinforcement Learning for Spatial Reuse in Dense Wi-Fi 6 Networks

Haonan Chi
haonan.chi@griffithuni.edu.au
School of ICT,
Griffith University,
Brisbane, Australia

Wee Lum Tan
w.tan@griffith.edu.au
School of ICT,
Griffith University,
Brisbane, Australia

Hui Tian
hui.tian@griffith.edu.au
School of ICT,
Griffith University,
Brisbane, Australia

***Abstract*—Dense IEEE 802.11ax Wireless Local Area Networks (WLANs) can improve spectral efficiency through spatial reuse, but the practical gain depends on how the Overlapping Basic Service Set Packet Detection (OBSS/PD) threshold, Transmit Power (TP), and the Modulation and Coding Scheme (MCS) level are configured. Existing spatial reuse methods commonly optimize AP-level channel-access parameters, while rate control is typically delegated to local algorithms. This paper proposes *Centralized Adaptive Per-AP Spatial Reuse with MCS Control* (CAPS), a centralized reinforcement-learning controller that jointly optimizes per-AP TP & OBSS/PD settings and per-link MCS level in dense single-channel multi-BSS Wi-Fi 6 networks. CAPS uses two coordinated actors: an AP-level spatial reuse actor for TP and OBSS/PD configuration, and a link-level MCS actor for AP–STA link rate selection. A centralized critic evaluates the complete network configuration using a throughput–fairness reward. Extensive simulations on three increasingly dense network topologies clearly show that our proposed CAPS algorithm outperforms two representative spatial reuse algorithms and the default Wi-Fi 6 standard. In the largest and densest network topology, CAPS achieves the highest aggregate network throughput–with up to 13.7% improvement over Wi-Fi 6 and 6.9% over competing spatial reuse algorithms–while preserving high throughput fairness across BSSs and ensuring strong delivery reliability.**

***Index Terms*—Wi-Fi 6, spatial reuse, OBSS/PD, transmit power control, link rate adaptation, reinforcement learning**

I. Introduction

Dense deployments of IEEE 802.11 Wireless Local Area Networks (WLANs) are increasingly constrained by contention and interference among co-channel Basic Service Sets (BSSs). In conventional Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA), devices defer transmission when the medium is sensed as busy, even if a concurrent transmission may be decodable at the intended receiver. This conservative channel-access behavior avoids many collisions but reduces spatial reuse in dense Overlapping Basic Service Set (OBSS) deployments.

IEEE 802.11ax, marketed as Wi-Fi 6, introduces BSS coloring and OBSS Packet Detection (OBSS/PD)-based spatial reuse mechanisms to increase concurrency in dense deployments [1], [2]. BSS coloring enables a device to distinguish intra-BSS and inter-BSS frames, while OBSS/PD allows a device to ignore sufficiently weak inter-BSS transmissions and continue its own channel-access procedure. However, the standard defines mechanisms and parameter ranges rather than a deployment-specific algorithm for selecting spatial reuse configurations. The resulting control problem is difficult because the OBSS/PD, Transmit Power (TP), and Modulation and Coding Scheme (MCS) decisions are coupled. Raising the OBSS/PD threshold can increase channel-access opportunities, but it can also increase interference at receivers. Reducing TP can reduce the interference footprint, but will also weaken the signal strength on the intended AP–STA[1] link. Selecting a higher MCS can increase the physical-layer data rate, but it requires stronger link quality and is more vulnerable to interference [3].

Learning-based spatial reuse optimization has made important progress. Existing studies include centralized Multi-Armed Bandit (MAB) search for optimal TP & OBSS/PD settings [4], decentralized MAB-based spatial reuse [5], deep reinforcement learning for carrier-sensing and TP control [6], [7], and multi-AP cooperative access control [8]. Recent work has also recognized that spatial reuse depends on receiver-specific interference relationships. For example, Du et al. [9] incorporate link-specific state information and jointly determine transmission rate and power for IEEE 802.11bn spatial reuse. Nevertheless, most coordinated, centralized, and decentralized spatial reuse methods still independently optimize AP or BSS-level parameters such as Packet Detect (PD), Carrier Sensing Threshold (CST), TP and OBSS/PD [10], [11], while MCS rate selection is typically delegated to local rate control algorithms [12], [13].

This paper proposes *Centralized Adaptive Per-AP Spatial Reuse with MCS Control* (CAPS), a centralized reinforcement-learning controller that jointly optimizes per-AP TP & OBSS/PD settings and per-link MCS level in dense single-channel multi-BSS Wi-Fi 6 networks. CAPS is motivated by the observation that spatial reuse decisions in dense Wi-Fi networks are inherently coupled across BSSs. A configuration that improves one BSS may degrade another BSS, especially in asymmetric topologies where APs experience different levels of contention, interference, and receiver-side vulnerability. CAPS adopts a centralized learning and per-AP control structure to address this problem. The centralized CAPS controller learns

[1]In this work, an access point (AP) forms a BSS and serves a set of associated stations (STAs)

979-8-3195-0167-7/26/$31.00 ©2026 IEEE

from network-wide measurements collected across all APs, allowing it to capture the coupled effects of spatial reuse decisions across different BSSs. The learned policy derives AP-specific TP and OBSS/PD actions from the global network state, and each AP applies its assigned configuration rather than a common network-wide setting. This design follows the principle of centralized decision making with per-AP execution: global observations are used to coordinate the network, while the resulting control actions remain specific to individual APs. CAPS also incorporates link-specific MCS control for each AP–STA link. Spatial reuse gains depend not only on creating additional transmission opportunities, but also on ensuring that concurrent transmissions remain decodable under the resulting signal-to-interference-plus-noise ratio (SINR). The selected MCS must therefore be robust to the interference conditions induced by the AP-specific TP and OBSS/PD settings. By jointly controlling TP, OBSS/PD, and MCS, CAPS coordinates medium access, interference management, and PHY-rate robustness, enabling increased concurrency to translate into effective throughput gains.

Our primary contributions in this paper are as follows:

- We formulate dense Wi-Fi network spatial reuse as a joint AP-level and link-level control problem, where TP and OBSS/PD are AP-level actions and MCS is an AP–STA link-level action.
- We design CAPS, a centralized reinforcement learning controller that coordinates AP-level spatial reuse and link-level rate selection under one centralized critic.
- We evaluate CAPS across three increasingly dense multi-BSS network topologies and compare its performance with the centralized MAB [4], decentralized selfish MAB [5], and Wi-Fi 6 [1] spatial reuse algorithms. Our results show that CAPS consistently outperforms the other three algorithms by achieving the highest aggregate network throughput in all scenarios, while preserving high throughput fairness across BSSs and achieving consistent delivery reliability.

The remainder of this paper is organized as follows. Section II reviews related work, while Section III presents the design of our proposed CAPS algorithm. The experimental evaluation of our CAPS algorithm is presented in Section IV, followed by the conclusion in Section V.

II. Related Work

A. Spatial Reuse in IEEE 802.11ax

IEEE 802.11ax [1] spatial reuse relies on BSS coloring and OBSS/PD to allow selected inter-BSS transmissions to be ignored when they are sufficiently weak. Zhu [2] highlighted that improper TP and carrier-sensing threshold selection can cause spatial reuse to backfire, where a new transmission opportunity is created but the resulting SINR is insufficient for decoding. Karakoc [3] showed that OBSS/PD threshold adjustment has an impact on rate selection because the selected MCS must remain decodable under the resulting interference condition. Link-aware spatial reuse has also been studied in LSR [14], which considers mutual interference between transmission links when selecting spatial reuse opportunities. These studies motivate a controller that jointly accounts for channel access, interference footprint, and link robustness.

B. Learning-Based Spatial Reuse Control

Bardou [4] formulated the IEEE 802.11ax TP & OBSS/PD configuration as a centralized MAB problem, using candidate sampling and Thompson Sampling to search for high-performing configurations while reducing starvation. Wilhelmi [5] proposed a decentralized MAB-based spatial reuse approach, where each WLAN independently adjusts channel, TP, and carrier-sensing threshold. Deep reinforcement learning has also been applied to Wi-Fi access control. Huang [6] used Deep Q-Networks (DQN) for CCA threshold and TP optimization, and later proposed a three-tier deep learning method for channel access, TP, and CCA threshold selection [7]. Zhang [8] used Deep Deterministic Policy Gradient (DDPG) for multi-AP access control, jointly tuning contention window and OBSS/PD. Our proposed work, CAPS, differs from these works by jointly learning AP-level spatial reuse actions and AP–STA link-level MCS actions. Rather than treating rate control as an external mechanism, CAPS exposes MCS as a learnable per-link action and evaluates the resulting joint network configuration using a centralized critic.

C. Rate Adaptation and Hybrid Action Spaces

Rate adaptation is a long-standing WLAN problem. Minstrel and Minstrel-HT [12], [13] used measured performance statistics, probing, and retry chains to select rates (or MCS) for each remote station. This per-station behavior is important for dense WLANs because different links under the same AP may experience different SINR. From a reinforcement-learning perspective, CAPS has a heterogeneous action space: TP and OBSS/PD are continuous or quasi-continuous, while MCS is a discrete index. Hybrid actor–critic methods have been proposed for structured and parameterized action spaces [15]. Multi-actor centralized-critic structures, such as MADDPG [16], also show the value of using centralized information to train multiple action-producing policies in coupled environments. CAPS adapts these ideas to Wi-Fi spatial reuse by decomposing the heterogeneous action space into an AP-level spatial reuse actor and a link-level MCS actor, while using one centralized critic to evaluate the full TP, OBSS/PD, and MCS configuration.

III. CAPS Design

This section presents the proposed Centralized Adaptive Per-AP Spatial Reuse with MCS Control algorithm (CAPS). CAPS is a centralized two-actor reinforcement-learning controller that jointly determines AP-level spatial reuse parameters and link-level physical-layer rate decisions. This structure follows the general idea of hybrid actor–critic designs for structured action spaces, where multiple policy modules are trained under a shared critic [15]. In CAPS, the first spatial reuse actor controls the TP and OBSS/PD threshold of each AP, while the second MCS actor selects the MCS for each AP–STA link. A centralized critic evaluates the complete network-level action.

A. System Model Overview

We consider a dense single-channel IEEE 802.11ax network with K APs. Each AP forms one Basic Service Set (BSS) and serves a set of associated stations (STAs). The STA set associated with AP k is denoted by $\mathcal{S}_k$, where $k \in \{1, \ldots, K\}$. Downlink saturated traffic is transmitted from APs to their associated STAs. The total number of AP–STA links is

$$C = \sum_{k=1}^{K} |\mathcal{S}_k| \tag{1}$$

The controller operates at fixed decision epochs with an interval $\Delta t = 0.1$s. At each decision epoch t, CAPS receives aggregated network measurements from the previous interval. CAPS then constructs AP-level observations for the spatial reuse actor. This actor selects the TP and OBSS/PD threshold settings for each AP. The selected AP-level action is then used as part of the conditioning input for the link-level MCS actor, which selects the MCS for each AP–STA link. The AP-level and link-level actions are concatenated into one joint network action, forming the resultant network configuration for the next interval.

The objective is to increase useful delivered throughput while avoiding severe unfairness and excessive frame loss. Therefore, CAPS treats spatial reuse not only as a channel-access problem, but also as a receiver-side reliability problem. A spatial reuse configuration is useful only when the additional, concurrent transmission opportunities are converted into successfully delivered application-layer traffic.

B. Observation Space

CAPS uses two observation spaces: AP-level observations for spatial reuse control and link-level observations for MCS control.

1) AP-Level Observation for Spatial Reuse Control: The first actor operates at the AP level. For AP k at each decision epoch t, the AP-level observation contains local BSS measurements, the AP's own previous spatial reuse action, and the previous spatial reuse actions of the other APs. It is defined as

$$\mathbf{o}_t^{(k)} = \left[x_t^{(k)},\, J_t,\, F_t^{(k)},\, \Gamma_t^{(k)},\, P_{t-1}^{(k)},\, \theta_{t-1}^{(k)},\, \mathbf{c}_{t-1}^{(k)}\right]^{\mathrm{T}} \tag{2}$$

where $x_t^{(k)}$ is the throughput of AP k, J_t is the Jain's fairness index, $F_t^{(k)}$ is the AP's attempt-level success ratio (ASR), and $\Gamma_t^{(k)}$ is the minimum STA-side SINR across all the STAs associated with AP k. The terms $P_{t-1}^{(k)}$ and $\theta_{t-1}^{(k)}$ are the TP and OBSS/PD threshold settings applied by AP k in the previous interval. The vector $\mathbf{c}_{t-1}^{(k)}$ provides the previous spatial reuse actions of all other APs in the network:

$$\mathbf{c}_{t-1}^{(k)} = \left[P_{t-1}^{(j)},\, \theta_{t-1}^{(j)}\right]_{j\in\{1,\ldots,K\},\, j\neq k} \tag{3}$$

By observing the previous TP and OBSS/PD choices of neighboring APs, the shared actor can learn more coordinated AP-level behavior. Note that although the observation is AP-level, the Jain's fairness index is a global network fairness indicator of the throughput distribution across all the APs. CAPS uses the Jain's fairness index to prevent each AP's decision from becoming purely selfish.

At decision epoch t, the full AP-level observation is the ordered collection

$$\mathbf{O}_t^{\mathrm{AP}} = \left[\mathbf{o}_t^{(1)},\, \mathbf{o}_t^{(2)},\, \ldots,\, \mathbf{o}_t^{(K)}\right] \tag{4}$$

2) Link-Level Observation for MCS Control: The second actor operates at the AP–STA link level. For a link (k, c), where STA $c \in \mathcal{S}_k$, the link-level environment observation is

$$\mathbf{z}_t^{(k,c)} = \left[P_{t-1}^{(k)},\, \theta_{t-1}^{(k)},\, m_{t-1}^{(k,c)},\, d_t^{(k,c)},\, ret_t^{(k,c)},\, \gamma_t^{(k,c)},\, x_t^{(k,c)}\right]^{\mathrm{T}} \tag{5}$$

where $P_{t-1}^{(k)}$ and $\theta_{t-1}^{(k)}$ are the previous TP and OBSS/PD threshold settings of AP k, $m_{t-1}^{(k,c)}$ is the previous selected MCS for link (k, c), $d_t^{(k,c)}$ is the MCS fallback indicator for the link, $ret_t^{(k,c)}$ is the frame retransmission attempts count for the link, $\gamma_t^{(k,c)}$ is the link's reported SINR, and $x_t^{(k,c)}$ is the link throughput.

The CAPS design incorporates an MCS retry-aware fallback mechanism whereby if a frame transmission using the selected MCS were to fail, frame retransmission attempts will use a lower MCS value. The selected MCS value is reduced by one level for each retransmission attempt, until the frame retransmission succeeds or until the retry limit is reached. In Equation (5), $d_t^{(k,c)}$ is the MCS fallback indicator that represents the outcome of the retry-aware fallback mechanism in the previous interval. It is basically the difference between the selected MCS and the actual MCS with which a frame transmission is successful. Note that in Equation (5), the previously selected MCS $m_{t-1}^{(k,c)}$ is retained as a separate observation because the MCS fallback indicator can have different meanings at different absolute MCS levels. For example, falling back by two levels from a high MCS may still indicate a relatively strong link, whereas falling back by two levels from a low MCS may indicate a weak or high-interference link.

The link-level observation also uses link throughput $x_t^{(k,c)}$ to directly measure the effective delivered performance of the AP–STA link. Reliability information is still retained through the frame retransmissions attempts count $ret_t^{(k,c)}$ and the MCS fallback indicator $d_t^{(k,c)}$. This allows the MCS actor to distinguish between links that deliver high throughput with little fallback and links where frame transmissions only succeed after repeated retransmissions with lower MCS.

The current AP-level action selected by the spatial reuse actor is also used by the MCS actor as a conditioning input. Hence, the complete input to the MCS actor for link (k, c) is

$$\left[\mathbf{z}_t^{(k,c)},\, \mathbf{a}_t^{(k),\mathrm{SR}}\right] \tag{6}$$

where $\mathbf{a}_t^{(k),\mathrm{SR}} = [P_t^{(k)}, \theta_t^{(k)}]^{\mathrm{T}}$ is the current AP-level action selected by the spatial reuse actor. This term is not an environment measurement; it is the current decision produced by the first spatial reuse actor. It is included because the learned MCS selection for a link is also dependent on the TP and OBSS/PD setting under which the next transmission will be attempted.

The complete link-level observation is the ordered collection

$$\mathbf{Z}_t^{\mathrm{link}} = \left[\mathbf{z}_t^{(k,c)}\right]_{k=1,\ldots,K;\, c\in\mathcal{S}_k} \tag{7}$$

C. Two-Actor Action Design

CAPS uses two coordinated actors and one centralized critic. The first actor controls AP-level spatial reuse, while the second actor controls link-level MCS.

1) Spatial Reuse Actor: The spatial reuse actor is denoted by μ_ϕ^{SR}, where ϕ represents the actor parameters. It is shared across APs. For AP k, it maps the AP-level observation to a two-dimensional normalized output:

$$\begin{aligned}\mathbf{u}_t^{(k),\mathrm{SR}} &= \mu_\phi^{\mathrm{SR}}(\mathbf{o}_t^{(k)}) \\ &= \left[u_{P,t}^{(k)},\, u_{\theta,t}^{(k)}\right]^{\mathrm{T}}, \quad u_{P,t}^{(k)}, u_{\theta,t}^{(k)} \in [-1,1]\end{aligned} \tag{8}$$

Then, the normalized outputs are mapped to physical control values:

$$P_t^{(k)} = \frac{u_{P,t}^{(k)}+1}{2}(P_{\max}-P_{\min}) + P_{\min} \tag{9}$$

$$\theta_t^{(k)} = \frac{u_{\theta,t}^{(k)}+1}{2}(\Theta_{\max}-\Theta_{\min}) + \Theta_{\min} \tag{10}$$

The resulting AP-level spatial reuse action is

$$\mathbf{a}_t^{(k),\mathrm{SR}} = \left[P_t^{(k)},\, \theta_t^{(k)}\right]^{\mathrm{T}} \tag{11}$$

Applying the same actor to all APs reduces the number of trainable parameters and encourages the actor to learn a reusable AP-level decision rule. The executed action remains AP-specific because each AP provides a different local observation and a different global action context.

2) Link-Level MCS Actor: The link-level MCS actor is denoted by μ_η^{MCS}, where η represents its parameters. It is shared across AP–STA links. For link (k,c), it receives the link-level observation and the associated AP-level action, and outputs a normalized, continuous MCS control value:

$$u_{m,t}^{(k,c)} = \mu_\eta^{\mathrm{MCS}}\left(\mathbf{z}_t^{(k,c)},\, \mathbf{a}_t^{(k),\mathrm{SR}}\right), \quad u_{m,t}^{(k,c)} \in [-1,1] \tag{12}$$

The continuous MCS actor output is treated as a relaxed control variable during actor updates. However since MCS is a discrete index, the MCS actor output is mapped and quantized into an initial MCS for execution:

$$m_t^{(k,c)} = \mathrm{round}\left[\frac{u_{m,t}^{(k,c)}+1}{2}(M_{\max}-M_{\min}) + M_{\min}\right] \tag{13}$$

The value is clipped to $[M_{\min}, M_{\max}]$ according to the feasible MCS range before it is applied as the initial MCS for each corresponding AP–STA link; the MCS retry-aware fallback mechanism may subsequently reduce this MCS during retries if a frame transmission fails.

3) Joint Action: The complete joint action is

$$\mathbf{A}_t = \left[\{P_t^{(k)}, \theta_t^{(k)}\}_{k=1}^{K}, \{m_t^{(k,c)}\}_{k=1}^{K,c\in\mathcal{S}_k}\right] \tag{14}$$

D. Centralized Critic

The centralized critic is denoted by Q_ψ, where ψ represents the critic parameters. It receives the full AP-level observation, the full link-level observation, and the complete joint action:

$$Q_\psi\left(\mathbf{O}_t^{\mathrm{AP}},\, \mathbf{Z}_t^{\mathrm{link}},\, \mathbf{A}_t\right) \approx \mathbb{E}\left[\sum_{\ell=0}^{\infty}\gamma^\ell r_{t+\ell}\right] \tag{15}$$

The critic outputs one scalar action-value estimate for the complete network configuration. It is centralized because the coupling affects multiple AP's TP, OBSS/PD, or MCS decision and the channel quality of multiple AP–STA links. A local critic would not observe enough context to assign credit when one AP's spatial reuse action affects another AP's link reliability or throughput.

E. Reward and Training

CAPS is trained using Deep Deterministic Policy Gradient (DDPG)-style off-policy actor–critic learning [17]. A transition stored in the replay buffer has the form

$$e_t = \left(\mathbf{O}_t^{\mathrm{AP}},\, \mathbf{Z}_t^{\mathrm{link}},\, \mathbf{A}_t,\, r_t,\, \mathbf{O}_{t+1}^{\mathrm{AP}},\, \mathbf{Z}_{t+1}^{\mathrm{link}},\, d_t\right) \tag{16}$$

where d_t is a terminal indicator. The base reward r_t^{base} is a weighted harmonic mean of normalized aggregate throughput and normalized Jain's fairness index:

$$r_t^{\mathrm{base}} = \frac{(1+\lambda)\widehat{T}_t\widehat{J}_t}{\lambda\widehat{J}_t + \widehat{T}_t + \epsilon} \tag{17}$$

where $\widehat{T}_t = \mathrm{clip}_{[0,1]}(T_t/T_{\mathrm{ref}})$ is the normalized aggregate throughput, $\widehat{J}_t = \mathrm{clip}_{[0,1]}((J_t - 1/K)/(1-1/K))$ is the normalized Jain's fairness index, T_t is the aggregate network throughput, T_{ref} is the aggregate throughput reference used for normalization, $1/K$ is the minimum Jain's fairness value when one BSS receives all throughput, λ controls the throughput–fairness emphasis, and ϵ avoids division by zero. A larger λ places more emphasis on throughput while retaining the harmonic-mean penalty for severe unfairness.

To discourage the MCS actor from always selecting overly aggressive MCS values and relying on the MCS retry-aware fallback mechanism, CAPS subtracts a fallback penalty from the base reward:

$$r_t = \mathrm{clip}\left(r_t^{\mathrm{base}} - \beta_{\mathrm{fb}}\overline{d}_t,\, 0,\, 1\right) \tag{18}$$

where β_{fb} is the fallback-penalty coefficient and

Algorithm 1 CAPS

Require: Training episodes N_{ep}; decision epochs per episode T_{ep}; replay buffer $\mathcal{D}$; control interval Δt.
Ensure: Trained spatial reuse actor μ_{ϕ}^{SR} and MCS actor $\mu_{\eta}^{\mathrm{MCS}}$.

Initialize spatial reuse actor, MCS actor, centralized critic, and their target networks.
Initialize replay buffer $\mathcal{D}$ and measurement buffers.
for episode $= 1$ to N_{ep} **do**
  Reset the environment and initialize AP/link configurations.
  for $t = 1$ to T_{ep} **do**
    Collect interval measurements from network.
    Update throughput, fairness, ASR, SINR, retry, and fallback statistics.
    Construct AP observations $\mathbf{O}_t^{\mathrm{AP}}$ and link observations $\mathbf{Z}_t^{\mathrm{link}}$.
    if random warm-up is active **then**
      Sample random joint action $\mathbf{A}_t$.
    else
      Use μ_{ϕ}^{SR} to select per-AP TP & OBSS/PD actions.
      Append each AP action to its associated link observations.
      Use $\mu_{\eta}^{\mathrm{MCS}}$ to select per-link MCS actions.
      Add exploration noise and form joint action $\mathbf{A}_t$.
    end if
    Apply $\mathbf{A}_t$ in network for the next control interval Δt.
    Execute retry-aware MCS fallback inside the rate-control layer.
    Compute reward r_t and observe the next state.
    Store transition e_t in replay buffer $\mathcal{D}$.
    if $|\mathcal{D}| \geq B$ and warm-up has completed **then**
      Update networks using Algorithm 2.
    end if
  end for
end for

$$\overline{d}_t = \frac{1}{C} \sum_{k=1}^{K} \sum_{c \in \mathcal{S}_k} d_t^{(k,c)} \tag{19}$$

is the mean fallback indicator across all AP-STA links. The fallback penalty is used during training to discourage inefficient high-MCS policies that depend on repeated fallback.

During training, sampled mini-batches are used to update the centralized critic and both actors. Target copies of the spatial reuse actor, MCS actor, and critic are used to compute stable temporal-difference targets. During evaluation, exploration noise is disabled, replay-buffer updates are stopped, and the trained actors are executed deterministically. Algorithm 1 and Algorithm 2 show the pseudocode detailing the CAPS training procedure.

IV. Experimental Evaluation

A. Experimental Settings

We compare the performance of CAPS against the default Wi-Fi 6 standard [1] and two representative spatial reuse algorithms: Bardou [4] and Wilhelmi [5]. Bardou's approach is a centralized MAB-based method that optimizes TP and OBSS/PD while minimizing starvation. On the other hand, Wilhelmi's approach is a decentralized selfish MAB method where each AP learns the optimal TP and OBSS/PD settings by using its own performance feedback. A summary of the compared methods is provided in Table I.

We evaluate the spatial reuse performance of CAPS, Bardou [4], Wilhelmi [5], and Wi-Fi 6 [1] in three network topologies with progressively increasing complexity and node density as seen in Table II and Fig. 1. The topologies are designed to gradually increase the number of overlapping BSSs and the degree of inter-BSS interference coupling. The APs are placed on grid layouts with fixed adjacent spacing, while the STAs are randomly placed around the associated AP.

All experiments are conducted in ns-3 with ns3-gym as the interface between the packet-level simulator and the Python reinforcement-learning controller [18]. Table III summarizes the network simulation settings. The results presented in this paper are averaged over 30 simulation runs. All APs and STAs share a single channel, with the AP transmitting saturated downlink UDP traffic to its associated STAs. A-MPDU aggregation is disabled to isolate the effects of per-link MCS selection and retry behavior from aggregation-dependent batching within the 0.1 s control interval. In the default Wi-Fi 6 settings, all APs use BSS Coloring, TP and OBSS/PD settings to enable concurrent transmissions. Under these settings, whenever an AP transmits to one of its STAs, neighboring BSSs experience some level of interference, and the spatial distribution of STAs affects whether the AP's transmission succeeds. Since the STA positions are randomized, each AP faces a distinct local environment, allowing us to compare performance differences across the four algorithms.

Table IV lists the hyperparameters that our proposed CAPS algorithm used for the reported experiments. They are not claimed to be universally optimal; rather, they provide the training configuration under which the reported experimental

Algorithm 2 CAPS Actor–Critic Update

Require: Replay buffer $\mathcal{D}$; mini-batch size B; discount factor γ; target update rate τ.

Sample a mini-batch of transitions from $\mathcal{D}$:

$$(\mathbf{O}_i^{\mathrm{AP}}, \mathbf{Z}_i^{\mathrm{link}}, \mathbf{A}_i, r_i, \mathbf{O}_{i+1}^{\mathrm{AP}}, \mathbf{Z}_{i+1}^{\mathrm{link}}, d_i).$$

Use the target spatial reuse actor to compute next AP-level actions.
Append each target AP action to the corresponding next link observations.
Use the target MCS actor to compute next MCS actions.
Concatenate target AP-level and link-level actions into $\mathbf{A}'_i$.
Compute the temporal-difference target:

$$y_i = r_i + \gamma(1 - d_i) Q_{\psi'}(\mathbf{O}_{i+1}^{\mathrm{AP}}, \mathbf{Z}_{i+1}^{\mathrm{link}}, \mathbf{A}'_i).$$

Update critic Q_{ψ} by minimizing:

$$L_Q = \frac{1}{B} \sum_i \left(Q_{\psi}(\mathbf{O}_i^{\mathrm{AP}}, \mathbf{Z}_i^{\mathrm{link}}, \mathbf{A}_i) - y_i \right)^2.$$

Update both actors by maximizing the centralized critic value:

$$L_{\mu} = -\frac{1}{B} \sum_i Q_{\psi}(\mathbf{O}_i^{\mathrm{AP}}, \mathbf{Z}_i^{\mathrm{link}}, \mathbf{A}_{\mu,i}).$$

Soft-update all target networks:

$$\phi' \leftarrow \tau\phi + (1-\tau)\phi', \quad \eta' \leftarrow \tau\eta + (1-\tau)\eta', \quad \psi' \leftarrow \tau\psi + (1-\tau)\psi'.$$

TABLE I: Compared methods

Method	Spatial reuse control	MCS handling
CAPS	Learned TP & OBSS/PD	Learned per-link MCS
Bardou [4]	Centralized MAB	Minstrel-HT
Wilhelmi [5]	Decentralized selfish MAB	Minstrel-HT
Wi-Fi 6 [1]	Default TP & OBSS/PD	Minstrel-HT

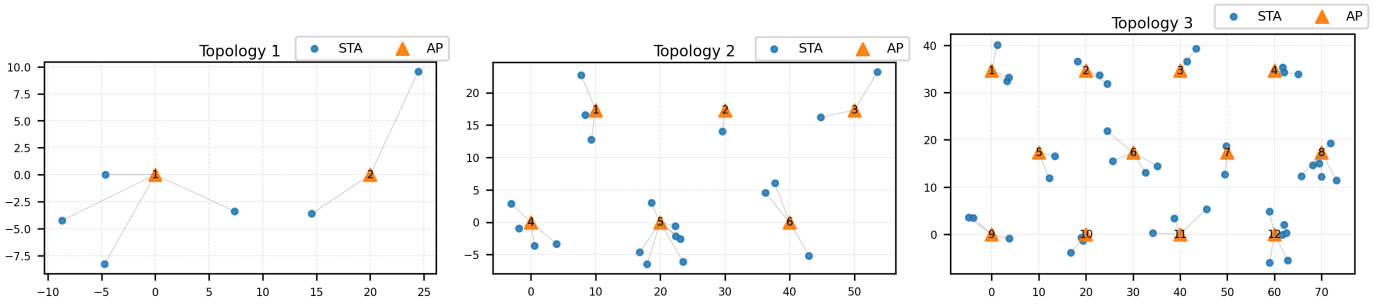


Fig. 1: Three network topology layouts with associated AP-STA pairs. The distances between AP-STA pairs are in meters.

TABLE II: Evaluation topologies

Topology	APs	STAs	Purpose
T1	2	6	Small interference-coupling case
T2	6	20	Medium dense multi-BSS case
T3	12	40	Dense multi-BSS stress-test case

TABLE III: Network simulation parameters

Parameter	Value
Simulator	ns-3.42 with ns3-gym
Learning framework	PyTorch
Wi-Fi standard	IEEE 802.11ax / Wi-Fi 6
A-MPDU	Disabled
Band and bandwidth	5 GHz, 20 MHz
Traffic	Saturated UDP downlink
Packet size	1500 bytes
Evaluation duration	120s
Propagation model	LogDistancePropagationLossModel
Path-loss exponent	4.0
Reference distance	1m
Reference loss	46.6777 dB
Error model	TableBasedErrorRateModel
TP range	[1, 21] dBm, default = 21dBm
OBSS/PD range	[-82, -62] dBm, default = -62dBm
MCS range	HE-MCS 0–11

TABLE IV: CAPS training hyperparameters

Parameter	Value
Training mode	Joint Actor 1 / Actor 2 training
Actor 1 learning rate	1×10^{-5}
Actor 2 learning rate	2×10^{-5}
Critic learning rate	4×10^{-4}
Control interval	0.1s
Discount factor γ	0.98
Target update rate τ	0.003
Batch size	256
Replay buffer capacity	64000 transitions
Random warm-up	1200 steps
Gradient warm-up	1200 steps
Actor hidden size	256
MCS actor hidden size	256
Critic hidden size	1024
Actor exploration σ	0.30 $\rightarrow$ 0.01, decay 0.93
MCS exploration σ_m	0.20 $\rightarrow$ 0.01, decay 0.95
Reward λ	10.0
Fallback penalty β_{fb}	0.1
Maximum fallback depth	up to 7 MCS levels

results were obtained.

In terms of performance metrics, we report the aggregate network throughput, BSS-level Jain's fairness index, and per-AP attempt-level success ratio. The aggregate network throughput is the sum of the individual AP's throughput in the network. The BSS-level Jain's fairness index is computed over the BSS throughput vector $\mathbf{x} = [x^{(1)}, \ldots, x^{(K)}]$:

$$J(\mathbf{x}) = \frac{\left(\sum_{k=1}^{K} x^{(k)}\right)^2}{K \sum_{k=1}^{K} \left(x^{(k)}\right)^2}. \tag{20}$$

The per-AP attempt-level success ratio is computed as

$$F^{(k)} = \frac{N_{\text{rx}}^{(k)}}{N_{\text{tx}}^{(k)}}, \quad N_{\text{tx}}^{(k)} > 0, \tag{21}$$

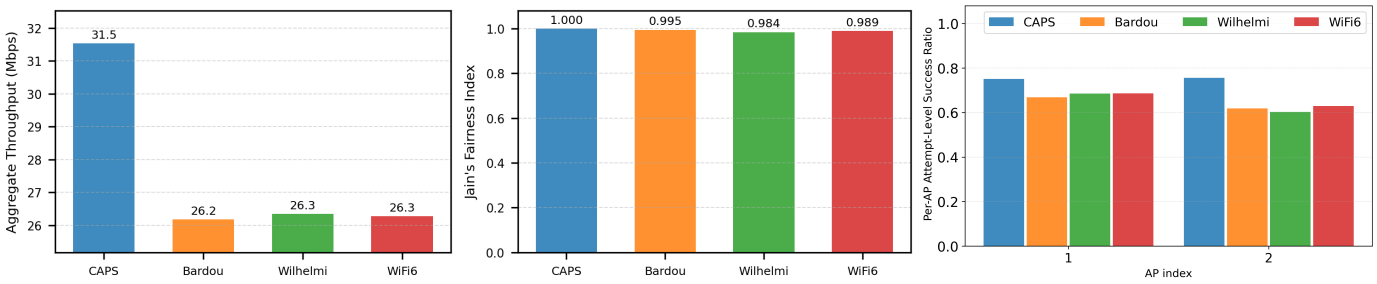
where $N_{\text{tx}}^{(k)}$ is the number of MAC-layer data-frame transmission attempts by AP k, including retransmissions, and $N_{\text{rx}}^{(k)}$ is the number of successfully received MAC-layer data frames associated with AP k. These metrics jointly indicate whether the evaluated algorithms produce useful spatial reuse, whether the improvement is fairly distributed across BSSs, and whether the selected TP, OBSS/PD, and MCS configurations convert channel-access attempts into successful frame deliveries.

TABLE V: Network throughput and BSS fairness results.

Topology	Metric	CAPS	Bardou	Wilhelmi	Wi-Fi 6
T1	Thr. (Mbps)	31.5	26.2	26.3	26.3
T1	Fairness	1.000	0.995	0.984	0.989
T2	Thr. (Mbps)	84.0	72.9	68.5	75.8
T2	Fairness	0.916	0.918	0.952	0.771
T3	Thr. (Mbps)	153.7	145.6	143.8	135.2
T3	Fairness	0.895	0.884	0.868	0.855

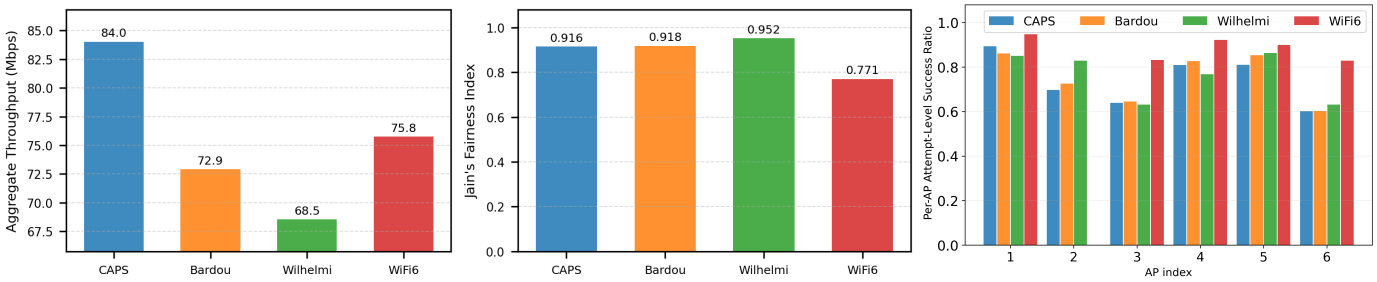
B. Simulation Results

Table V summarizes aggregate network throughput and BSS-level Jain's fairness index results over three increasingly dense topologies. Overall, CAPS achieves the highest aggregate network throughput among the evaluated methods while maintaining high throughput fairness between APs. Compared with Wi-Fi 6, CAPS improves throughput by 19.8%, 10.8%, and 13.7% in T1, T2, and T3, respectively. Compared with Bardou and Wilhelmi, CAPS achieves consistent throughput gains; its BSS fairness is also higher in T1 and T3, while in T2 it is slightly lower than the two methods. These results indicate that CAPS's joint AP-level TP & OBSS/PD control and link-level MCS selection provide a clear advantage compared to the other evaluated methods.



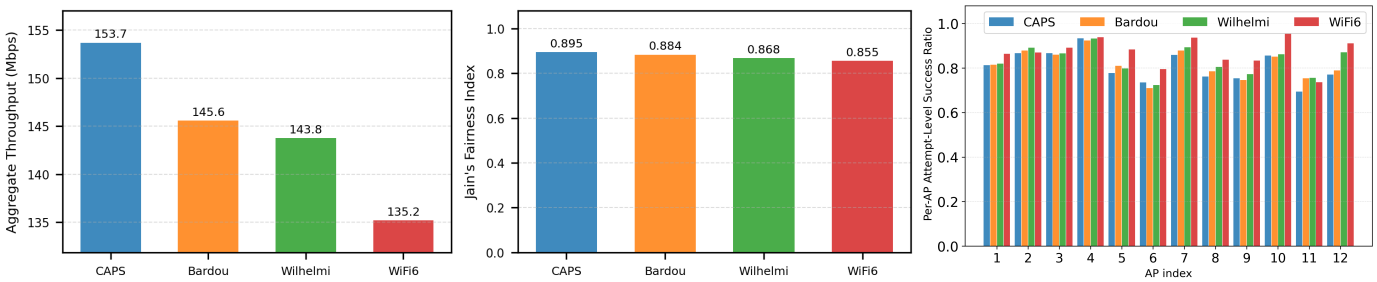
(a) Aggregated Network Throughput (b) Jain's Fairness Index (c) Per-AP Attempt-Level Success Ratio

Fig. 2: Topology 1 Performance Metrics



(a) Aggregated Network Throughput (b) Jain's Fairness Index (c) Per-AP Attempt-Level Success Ratio

Fig. 3: Topology 2 Performance Metrics



(a) Aggregated Network Throughput (b) Jain's Fairness Index (c) Per-AP Attempt-Level Success Ratio

Fig. 4: Topology 3 Performance Metrics

1) Topology 1: This is the smallest scenario and contains two APs and six STAs. Although this topology has fewer interference relationships than the larger scenarios, it is useful as a sanity check because an effective spatial reuse controller should improve throughput without creating obvious BSS starvation. The Topology 1 results in Fig. 2 show that CAPS achieves the highest aggregate network throughput of 31.5 Mbps while the three competing methods' throughput values are lower at around 26 Mbps. CAPS also achieves perfect BSS fairness and the highest attempt-level success ratio (ASR), as shown in Fig. 2b and Fig. 2c. In this small network topology, spatial-reuse opportunities and link reliability are aligned: CAPS can improve concurrency while maintaining sufficient receiver-side link quality.

2) Topology 2: This scenario increases the network size to six APs and twenty STAs. This topology introduces stronger coupling between BSSs: a configuration that benefits one AP can potentially interfere with the receiver-side reliability of another AP. It is therefore a more challenging test for spatial reuse. In Topology 2, Fig. 3a shows that CAPS achieves the highest aggregate network throughput of 84.0 Mbps, outperforming Wi-Fi 6 by 10.8%, Bardou by 15.2%, and Wilhelmi by 22.6%. Although Bardou's and Wilhelmi's approaches obtain slightly higher BSS fairness, their aggregate network throughput is lower. Wi-Fi 6 has the lowest BSS fairness, indicating that the default Wi-Fi 6 spatial-reuse settings do not fairly distribute throughput across BSSs: Fig. 3c shows that in the Wi-Fi 6 scenario, AP2 achieved an ASR of zero indicating total throughput starvation.

3) Topology 3: This is the densest scenario, with twelve APs and forty STAs, where local decisions can easily create network-wide interference and unfairness. Fig. 4a shows that CAPS achieves the highest aggregate network throughput of 153.7 Mbps while maintaining the highest fairness of 0.895 compared to the three competing methods. Although Fig. 4c indicates that CAPS's attempt-level success ratio is sometimes lower than that of competing methods, this reflects the more aggressive spatial-reuse configuration learned by CAPS: additional concurrent

transmissions improve airtime utilization (by selecting higher MCS) and increase aggregate network throughput, but expose some links to stronger inter-BSS interference. The competing methods are more conservative and therefore achieve higher attempt-level success ratios, while sacrificing transmission opportunities and overall throughput. These results also support CAPS centralized joint control design: as the number of APs and links increases, AP-level-only or static spatial-reuse policies become less effective. CAPS improves both throughput and fairness by coordinating AP-level channel access and interference control with link-level rate selection.

Across all three network topologies, CAPS is able to achieve the highest aggregate network throughput while maintaining high BSS fairness because it jointly controls AP-level interference exposure and link-level rate robustness. The spatial reuse actor adjusts the AP-level TP and OBSS/PD settings, which changes both the interference footprint and the set of feasible concurrent transmissions. The link-level MCS actor then selects the link-specific MCS for each AP–STA link under that AP-level configuration. This decomposition is important because an AP-level spatial reuse decision can change the SINR of multiple AP–STA links. Treating link-level rate control as independent from spatial reuse would ignore the effects of the TP and OBSS/PD settings on link robustness. Overall, these experimental results indicate that CAPS achieves a favorable throughput–fairness trade-off under the evaluated settings. They also support the main design choice of this paper: AP-level spatial reuse and link-level rate control should be coordinated under one network-level learning objective.

V. Conclusion

This paper has presented CAPS, a centralized reinforcement-learning controller for dense IEEE 802.11ax WLANs. CAPS jointly optimizes per-AP TP & OBSS/PD settings and per-link MCS level using two coordinated actors and a centralized critic. CAPS is designed to increase useful delivered throughput while avoiding severe unfairness and excessive frame loss. Through the joint optimization of AP-level spatial reuse parameters and link-level physical-layer rate decisions, CAPS is able to coordinate channel access, interference control, and link robustness so that spatial reuse concurrency produces useful throughput.

Experimental results across three increasingly dense single-channel network topologies show that CAPS consistently achieves the highest aggregate network throughput compared to competing spatial reuse methods. In the largest and densest network topology, CAPS improves aggregate network throughput by 13.7% over Wi-Fi 6, 5.6% over a centralized MAB scheme, and 6.9% over a decentralized selfish MAB scheme, while also achieving the highest BSS-level fairness. These results support the central claim of this paper: Wi-Fi spatial reuse and link-level rate control should be coordinated rather than optimized independently. Future work will investigate the performance of CAPS in dynamic network scenarios with STA mobility and varying traffic loads.

References

[1] *IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems Local and Metropolitan Area Networks–Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 1: Enhancements for High-Efficiency WLAN*, IEEE Std. 802.11ax, 2021.

[2] D. Zhu and S. Luan, "Spatial reuse in IEEE 802.11ax: Whether and how to use in practice," *KSII Transactions on Internet and Information Systems*, vol. 15, no. 12, pp. 4617–4638, 2021.

[3] A. Karakoç, H. B. Yilmaz, and M. Ş. Kuran, "More WiFi for Everyone: Increasing spectral efficiency in WiFi6 networks using a distributed OBSS/PD mechanism," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 31, no. 3, pp. 660–677, 2023.

[4] A. Bardou, T. Begin, and A. Busson, "Improving the spatial reuse in IEEE 802.11ax WLANs: A multi-armed bandit approach," in *Proceedings of the 24th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '21. Association for Computing Machinery, 2021, pp. 134–143.

[5] F. Wilhelmi, S. Barrachina-Muñoz, B. Bellalta, C. Cano, A. Jonsson, and G. Neu, "Potential and pitfalls of multi-armed bandits for decentralized spatial reuse in WLANs," *Journal of Network and Computer Applications*, vol. 127, pp. 26–42, 2019.

[6] Y. Huang and K.-W. Chin, "A deep Q-network approach to optimize spatial reuse in WiFi networks," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 6, pp. 6636–6646, 2022.

[7] Y. Huang and K.-W. Chin, "A three-tier deep learning-based channel access method for WiFi networks," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 1, pp. 90–106, 2023.

[8] H. Zhang, R. He, X. Fang, and L. Zhou, "DDPG-based multi-AP cooperative access control in dense Wi-Fi networks," in *Proceedings of the 2023 IEEE 98th Vehicular Technology Conference (VTC2023-Fall)*, 2023, pp. 1–6.

[9] M. Du, R. Yan, P. Liu, Z. Guo, and X. Sun, "Deep reinforcement learning based spatial reuse for IEEE 802.11bn," in *Proceedings of the 2025 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2025, pp. 1–6.

[10] F. Wilhelmi, B. Bellalta, S. Szott, K. Kosek-Szott, and S. Barrachina-Muñoz, "Coordinated multi-armed bandits for improved spatial reuse in Wi-Fi," in *Proceedings of the 2025 IEEE International Conference on Machine Learning for Communication and Networking (ICMLCN)*. IEEE, 2025, pp. 1–6.

[11] F. Wilhelmi, B. Bellalta, M. Casasnovas, A. Kijanka, and M. Calvo-Fullana, "Decentralized spatial reuse optimization in Wi-Fi: An internal regret minimization approach," 2026. [Online]. Available: https://arxiv.org/abs/2602.08456

[12] F. Fietkau, "minstrel_ht: New rate control module for 802.11n," Linux wireless mailing-list post archived by LWN.net, Mar. 2010, accessed: 2026-06-13. [Online]. Available: https://lwn.net/Articles/376765/

[13] T. Y. Arif, R. Munadi, and Fardian, "Evaluation of the Minstrel-HT rate adaptation algorithm in IEEE 802.11n WLANs," *International Journal of Simulation: Systems, Science and Technology*, vol. 18, no. 1, pp. 11.1–11.7, 2017.

[14] H. Lee, H.-S. Kim, and S. Bahk, "LSR: Link-aware spatial reuse in IEEE 802.11ax WLANs," in *Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1–6.

[15] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, ser. IJCAI-19. International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 2279–2285.

[16] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems 30*. Curran Associates, Inc., 2017.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proceedings of the 4th International Conference on Learning Representations*, 2016.

[18] P. Gawłowicz and A. Zubow, "ns-3 meets OpenAI Gym: The playground for machine learning in networking research," in *Proceedings of the 22nd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '19. Association for Computing Machinery, 2019, pp. 113–120.