Neuro-Cognitive Radios for Dynamic Spectrum Access

Rodrigo P. Ferreira
Pritzker School of Molecular Engineering
University of Chicago
Chicago, IL, USA
rpferreira@uchicago.edu

Pedro Lustosa Rege Botelho
Dep of Electrical and Computer Engineering
Northwestern University
Evanston, IL, USA
plrbotelho.ai@gmail.com

Yubo Zhang
Dep of Electrical and Computer Engineering
Northwestern University
Evanston, IL, USA
yubozhang2023@u.northwestern.edu

Igor Kadota
Dep of Electrical and Computer Engineering
Northwestern University
Evanston, IL, USA
kadota@northwestern.edu

Abstract

Neuromorphic computing is an emerging brain-inspired information processing paradigm that is well-suited for energy-efficient, real-time, and adaptive applications such as Dynamic Spectrum Access (DSA). In this paper, we develop for the first time a neuromorphic-based learning architecture to address a challenging decentralized DSA scenario where multiple source-destination pairs share a limited number of spectrum bands. Sources, called Neuro-Cognitive Radios (NCR), run spiking neural network (SNN)-based reinforcement learning (RL) architectures that adapt their transmission strategy over time, in a decentralized manner, aiming to maximize their own throughput while striving for network-wide fairness. We evaluate NCR in several network settings and compare with an equivalent Deep Q-Network (DQN) architecture that uses multi-layer perceptrons as opposed to spiking neurons. Our simulation results show that NCR outperforms DQN in terms of fairness while keeping a competitive throughput level, thereby demonstrating the potential of neuromorphic computing to address DSA problems.

1 Introduction

The importance of Dynamic Spectrum Access (DSA) for next-generation wireless networks has been highlighted in the U.S. National Spectrum Strategy [1], in the OSTP National Spectrum Research and Development Plan [2], in the International Telecommunication Union Report [3], and in many other recent reports. DSA in decentralized wireless networks with multiple source-destination pairs is especially challenging as it requires efficient, adaptive, and fair transmission scheduling policies that can be deployed on resource-constrained radios.

In recent years, Reinforcement Learning (RL) has gained attention as an effective method for facilitating DSA in decentralized wireless networks. Through interactions with both the environment and other sources, RL enables sources (also called agents) to learn and adapt their transmission strategies over time. However, most RL-based solutions proposed in the literature (see recent survey in [4]) use complex architectures (e.g., Dueling Deep Q Networks with Likelihood Hysteretic Implicit Quantile Network [14]), complex training procedures (e.g., centralized training and decentralized execution [5, 6, 8, 9]), and/or rely on the capability of sources to coordinate transmissions by sharing information with other sources implicitly or explicitly [10–13], thereby limiting the applicability of these RL-based solutions to real-world DSA scenarios with resource-constrained radios.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI and ML for Next-Generation Wireless Communications and Networking (AI4NextG).

Neuromorphic computing (NMC) is an emerging information processing paradigm inspired by the structure and function of the human brain, mimicking its sparse and event-driven communication, where neurons fire only when necessary, as described in section 3.1. NMC has been successfully applied to numerous applications [15–17] that require energy-efficient, real-time adaptation to dynamic scenarios including: (i) drone navigation and perception under tight energy budgets [19]; (ii) visual SLAM using "event cameras" that capture scenes asynchronously, consuming less power [20]; and (iii) power-efficient sensing using micro-Doppler radars [22]. We believe that these inherent features of NMC makes it a promising approach to address challenging DSA scenarios.

To the best of our knowledge, this is the first paper to bring neuromorphic methods to DSA. We propose Neuro-Cognitive Radio (NCR): a spiking neural network (SNN)-based RL architecture that learns to adapt its transmission strategy over time, aiming to maximize its own throughput while striving for network-wide fairness. NCRs training and execution is fully decentralized and does not rely on information sharing among sources for coordinating transmissions.

Over the next sections, we formally describe the DSA problem (section 2), explain our NCR solution in greater detail (section 3), discuss our numerical experiments (section 4), and our remarks on how to leverage and further expand our model's capabilities (section 5).

2 Dynamic Spectrum Access in Decentralized Wireless Networks

We consider a decentralized wireless communication network with M source-destination pairs transmitting packets via N orthogonal frequency bands. We assume that sources always have packets to transmit and that destinations are constantly listening to all bands. In each time slot $t \in \{1, ..., \mathcal{T}\}$, each source $m \in \{1, ..., M\}$ takes an action $a_m(t) \in \{0, 1, ..., N\}$, where $a_m(t) = 0$ means that source m is idle, and $a_m(t) = n \geq 1$ represents a transmission in band n during time slot t.

We then define $o_m(t)$ as the outcome of source m's action in time slot t. There are three possibilities for $o_m(t)$: if source m idles during t, then $o_m(t) = 0$; if only source m transmits in the selected band, then the transmission is successful and $o_m(t) = 1$; finally, $o_m(t) = -1$ when two or more sources transmit in the same band, leading to a packet collision.

We consider a DSA problem in which sources share no information to coordinate transmissions. At any given time slot t, source m knows only about its current and previous actions and outcomes—$\{a_m(k)\}_{k \leq t}$ and $\{o_m(k)\}_{k \leq t}$, respectively. Furthermore, sources have no prior knowledge about the network topology or size M.

3 Proposed Solution: Neuro-Cognitive Radio (NCR)

We present some fundamental concepts in neuromorphic computing followed by a detailed explanation on how to apply such ideas to our specific DSA problem.

3.1 Fundamentals of Neuromorphic Computing

Drawing inspiration from the human brain, neuromorphic computing is fundamentally different from its digital (von Neumann) counterpart on many levels. Apart from spike versus binary data representation, other major differences include operation (parallel or sequential) and organization (co-located or separated memory and processing units) [15, 16]. Moreover, while digital processors are time-driven (synchronous), neuromorphic processors are event-driven (asynchronous)—a major feature that has been leveraged to build energy-efficient computing platforms [16].

The building block of neuromorphic computing is the neuron. There are multiple models that accurately describe the neuron. Some optimize for higher biological fidelity (Hodgkin-Huxley, Morris-Lecar) whereas others focus on computational efficiency (Izhikevich, AdEx IF). Our work uses the leaky-integrate-and-fire (LIF) neuron due to its simplicity [18].

The LIF model approximates the neuron's membrane potential $U(t)$ to a low-pass filter circuit made of a resistor R and a capacitor C—which is valid from a biological standpoint—evolving over time as

$$\tau \frac{dU(t)}{dt} = -U(t) + RI(t), \tag{1}$$

where $\tau = RC$ is the circuit's time constant, and $I(t)$ is the current flowing through the circuit at any given time t. If the current $I(t)$ is constant over time, we obtain

$$U(t) = RI + (U_0 - RI)e^{-t/\tau}, \tag{2}$$

where U_0 is the membrane potential at $t = 0$. Defining the decay rate as $\beta = e^{-1/\tau}$, we can rewrite $U(t)$ in a discrete time domain via the forward Euler method as follows

$$U[t] = \beta U[t-1] + (1-\beta)I[t] \tag{3}$$

From a ML perspective, it is useful to write $I[t]$ as $WX[t]$, where W is the weight matrix and $X[t]$ is a vectorized input decoupled from effects of β. By doing so, we can write

$$U[t] = \beta U[t-1] + WX[t] - S[t-1]\theta, \tag{4}$$

which has three terms: decay ($\beta U[t-1]$), input ($WX[t]$), and reset ($S[t-1]\theta$), in which $S[t] = 1$ if $U[t] > \theta$, and $S[t] = 0$ otherwise.

A practical way to understand $U[t]$ is that the membrane potential decays over time according to its β factor and increases whenever a spike arrives at the neuron. At any time slot t, if the membrane potential $U[t]$ reaches its threshold value θ, the neuron fires a spike to its neighboring neurons and its potential resets to zero. In our model, both β and θ are tunable hyperparemeters—to be discussed in greater detail in sub-section 3.2.

3.2 Neuro-Cognitive Radio: States, SNN Architecture, Training Pipeline, and Rewards

In this section, we describe our proposed solution to the DSA problem, dubbed Neuro-Cognitive Radio (NCR). Each source is a NCR running a neuromorphic agent. In time slot t, the agent's state ($s_m(t)$) includes its own actions, outcomes, and binary time references from the previous T time slots. The binary time references [14] are represented in modulo 16, *i.e.*, mod(t,16), using 4 bits, allowing agents to find transmission patterns of different lengths by adaptively ignoring bits. Mathematically, for each $k \in \{t-T, \dots, t-1\}$, the state $s_m(t)$ will include time-ref(k), one-hot($a_m(k)$), $o_m(k)] \in \mathbb{Z}^F$, where the feature size F is given by $F = 4 + A + 1$, and A is the action space size, *i.e.*, $A = N + 1$ with all frequency bands and idle. Then, for $k \in \{t-T, \dots, t-1\}$, we have $s_m(t) \in \mathbb{R}^{T \times F}$ which is then converted into a torch tensor with dimension $1 \times T \times F$ to facilitate our model training pipeline.

Now, we describe the architecture of the SNN, which is composed of an input layer, one hidden layer, and an output layer. For each time index $k \in \{t-T+1, ..., t\}$, the input projection is done through a hidden linear layer (size H) that gives us $z_k = W_{in}s_m(k) + b_{in} \in \mathbb{R}^H$, where W_{in} and b_{in} are the input weight matrix and bias vector, respectively. Then z_k acts as input to a series of "spiking blocks". Each block is made of a linear layer (size H) followed by a LIF layer—analogously to the traditional structure of a multi-layer perceptron. The dimensions of the SNN architecture are $(input_{dim}, hidden_{dim}, output_{dim}) = (F, F, N+1)$.

Now, we explain how the learning process works through the spiking backbone. Let L be the total amount of spiking blocks and let's initialize the first block ($l = 1$) input embedding as $h_{0,k} = z_k$. For each block $l \in \{1, ..., L\}$, we have the following forward pass mechanism: (1) linear step as $a_{l,k} = W_l h_{l-1,k} + b_l$; (2) leaky-integrate as $\tilde{U}_{l,k} = \beta_l U_{l,k-1} + (1-\beta_l)a_{l,k}$; (3) threshold as $S_{l,k} = \mathcal{H}(\tilde{U}_{l,k} - \theta_l)$; (4) reset as $U_{l,k} = \tilde{U}_{l,k} - \theta_l S_{l,k}$; and (5) spike output as $h_{l,k} = S_{l,k} \in \{0,1\}^H$. In this case, the subscript l, k stands for the spiking block l at a given time index k, $\mathcal{H}$ is the Heaviside function, and U, β, θ, S have the same definitions as those from sub-section 3.1.

The forward pass output is the last step in the last layer, *i.e.*, $z = S_{L,T} \in \{0,1\}^H$, which remains in a spiking format. Then, to map it back to the problem's original output space (*i.e.*, the number of bands), we use the Q-value concept—a way to represent our action quality at a given time slot k. We do so via another linear layer: $Q(s,a) = W_{out}z + b_{out} \in \mathbb{R}^A$, where $A = N + 1$.

To select an action, the agent turns a decision score into a choice through the ε-greedy policy, by taking $\text{argmax}_a Q(s,a)$ with probability $1-\varepsilon$—aiming to balance the exploitation of promising bands and the exploration of new ones. Then, for each action, we save sequences (s, a, r, s', d) of length T, where r is the immediate reward (to be defined later in this sub-section) received after taking action a, s' is the updated state, and d is the "done" flag. After saving the (s, a, r, s', d) sequences, and sampling a batch with size (B, T, F), we define the target (y) and the loss ($\mathcal{L}$) functions as $y = r + \gamma(1-d)\max_{a'} Q(s', a')$ and the loss function $\mathcal{L} = (y - Q_{\text{online}}(s,a))^2$.

Now, we describe backpropagation. As described above, the forward pass uses the Heaviside function, which is not differentiable. A common way to address this issue is to differentiate a function with similar format—also known as the surrogate gradient approach. We write $dS_t/d\tilde{U}_t \approx g_\zeta(\tilde{U}_t - \theta)$, with surrogate gradient as $g_\zeta(u) = (1/\zeta)(1 + (u/\zeta)^2)^{-1}$ so that ζ is related to the "shape" of our surrogate function.

Finally, we use Adam optimizer, update the hyperparameters, and periodically copy Q_{online} to the actual target. Having completed a cycle, we update our states $s \leftarrow s'$, and continue to the next time slot. It is worth mentioning that throughout this process we do not use a distinct Q_{target} when computing $\max_{a'} Q(s', a')$.

Now, we describe the reward of agent m at the end of time slot t which is given by

$$r_m(t) = \begin{cases} 0.15 \times (1.5 - w_m(t)) \text{ , if } o_m(t) = 1, \text{[successful transmission]} \\ -1 \times w_m(t) \text{ , if } o_m(t) = -1, \text{[collision between two or more agents]} \\ -0.6 \text{ , if } o_m(t) = 0 \text{ and } \sum_{k=t-L}^{t} a_m(k) = 0, \text{[idle for more than } L \text{ slots]} \\ 0.03 \text{ , otherwise} \end{cases} \tag{5}$$

with $w_m(t)$ representing the weight associated with agent m during time slot t, where $w_m(t) = 0.4W_1 + 0.6W_2/(N-1)$, $W_1 = \sum_{k=t-L}^{t-1} \mathbb{I}_{\{a_m(k)=a_m(t)\}}(2^{k-t}|o_m(k)|)$, $W_2 = \sum_{n=0}^{N} \mathbb{I}_{\{n \neq a_m(t)\}} \sum_{k=t-L}^{t-1} \mathbb{I}_{\{a_m(k)=n\}}(2^{k-t}|o_m(k)|)$. The value of W_1 increases with the number of transmissions in the recent past $k \in \{t-L, \ldots, t-1\}$ using band $a_m(t)$. The value of W_2 increases with the number of transmissions in the recent past $k \in \{t-L, \ldots, t-1\}$ using bands other than $a_m(t)$. The multiplicative factor 2^{k-t} gives more importance to recent events. A high weight $w_m(t)$ reduces the reward of a successful transmission and increases the collision penalty.

It is worth noting that due to our focus on spiking architectures and strict page limit, we do not run experiments with additional DSA baseline methods. For completeness, future work shall include, for example, centralized-training RL methods [5, 6] as well as discretized tabular Q-learning (or value-iteration) approaches [7].

4 Results: Comparison of Neuro-Cognitive Radio and Deep Q-Network

We evaluate the performance of our proposed Neuro-Cognitive Radio solution, which utilizes a spiking neural network (SNN) backbone, against a baseline deep Q-network (DQN) approach (similar to [8, 9]) across several distinct DSA scenarios with different numbers of agents M and bands N. The baseline DQN architecture is identical to the SNN architecture described in section 3.2, with the same dimensions $(input_{dim}, hidden_{dim}, output_{dim}) = (F, F, N+1)$, but using multi-layer perceptrons instead of spiking neurons. All experiments are conducted over a time interval sufficient to achieve full convergence within the specified time slots. All experiments use the same hyperparameters outlined in Table 1. Performance is assessed using two key metrics: Jain's fairness index [23], which measures how equitably the spectrum is shared among agents (with 1 indicating perfect fairness), and total throughput, defined as the moving average number of successful transmissions per time slot (windowed by the last 500 time slots) normalized by the number of bands (N), reflecting overall spectrum utilization efficiency (with 1 indicating perfect throughput).

Table 2 summarizes the comparative results. In all scenarios, the SNN-based NCR achieves substantially higher fairness indices compared to DQN, while maintaining nearly equivalent total throughput. For instance, for $M = 5$ agents and $N = 1$ bands, fairness improves from 0.5289 (DQN) to 0.9459 (NCR), with throughput remaining at 0.9720. This indicates that our SNN architecture enables more equitable access without sacrificing efficiency. Similar trends hold for $M = 7$ and $N = 5$ (fairness: 0.8041 to 0.9732) and $M = 10$ and $N = 8$ (fairness: 0.8670 to 0.9706), where throughput reduces marginally from 0.9860 to 0.9688 and 0.9843 to 0.9685, respectively. These results highlight the SNN's ability to promote spectrum sharing in decentralized networks through its event-driven processing of time-dependent transmission patterns.

In addition to the main results highlighted in Table 2, we also perform several additional experiments, with $M \geq 2$ agents and $N \geq 2$ bands (Figure 1) to further illustrate NCR's capabilities. Across all 45 network settings, all using the same hyperparameters from Table 1, we can observe a high Jain's fairness index: greater than 0.95 in more than 50% of the experiments—achieving its lowest value

Table 1: Hyperparameters used in all experiments.

Parameter	Value
LIF neuron decay factor (β_0)	0.25
LIF neuron threshold (θ_0)	0.03
Discount factor (γ)	0.9
Exploration rate (ϵ)	5×10^{-2}
Exploration rate decay (ϵ_{decay})	6.5×10^{-6}
Minimum exploration rate (ϵ)	8×10^{-3}
Learning rate ($\tilde{\mu}$)	5×10^{-4}
Buffer size	500
Target network's replace rate	50
Batch size (B)	64
Reward history length (L)	16
Temporal length (T)	15

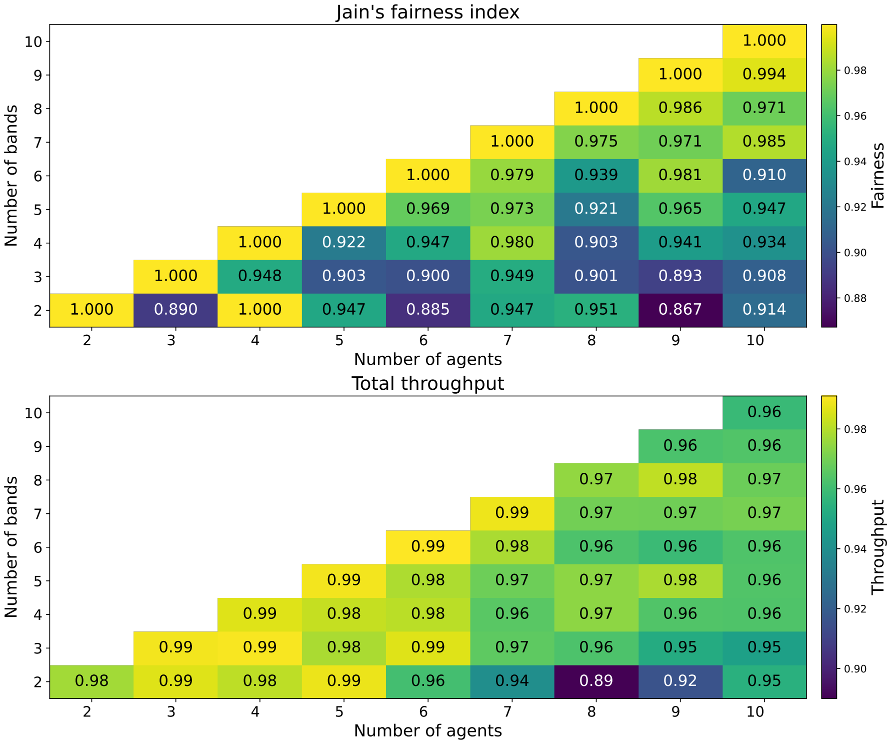


Figure 1: Jain's fairness index and total throughput across all 45 experiments conducted using NCR.

(0.8670) in the 9 agents and 2 bands setting. Moreover, we also observe high throughput—greater or equal to 0.95 in 42 (out of 45) experiments.

To further illustrate the dynamics of the network settings in Table 2, Figure 2 depicts the evolution of individual agent success rates (throughput per agent), along with aggregate collision and idle rates, over time. Rates are computed as moving averages over 500 time steps for smoothness. In the plots on the left side of Figure 2, across all scenarios, SNN agent success rates exhibit initial fluctuations during exploration but rapidly converge to similar values, demonstrating the SNN's ability to find suitable transmission patterns for every agent, leading to superior fairness results. For $M = 8$ agents and $N = 2$ band (Figure 2a), all eight agents stabilize around a success rate of approximately 0.25, with collisions dropping sharply to near 0 and idle rates remaining low. For $M = 7$ and $N = 5$ (Figure 2c), all agents converge to rates around 0.6, reflecting fair distribution of the 5 bands. For

Table 2: Comparison of DQN and NCR across network settings with different numbers of agents and bands. Both RL architectures have identical structures and use the hyperparameters outlined in Table 1 in every experiment.

#Agents	#Bands	Deep Q-Network (DQN)		Neuro-Cognitive Radio (NCR)	
M	N	Fairness Index	Total Throughput	Fairness Index	Total Throughput
10	8	0.8670	0.9843	0.9706	0.9685
9	3	0.6393	0.9527	0.8929	0.9520
7	5	0.8041	0.9860	0.9732	0.9688
8	2	0.6544	0.9790	0.9513	0.9680
4	2	0.6651	0.9060	0.7721	0.9830
3	2	0.8723	0.9860	0.8864	0.9760

Table 3: Reward-based ablation study using the CP1 reward structure. Comparison of DQN and NCR across network settings with different numbers of agents and bands. Both RL architectures have identical structures and use the hyperparameters outlined in Table 1 in every experiment.

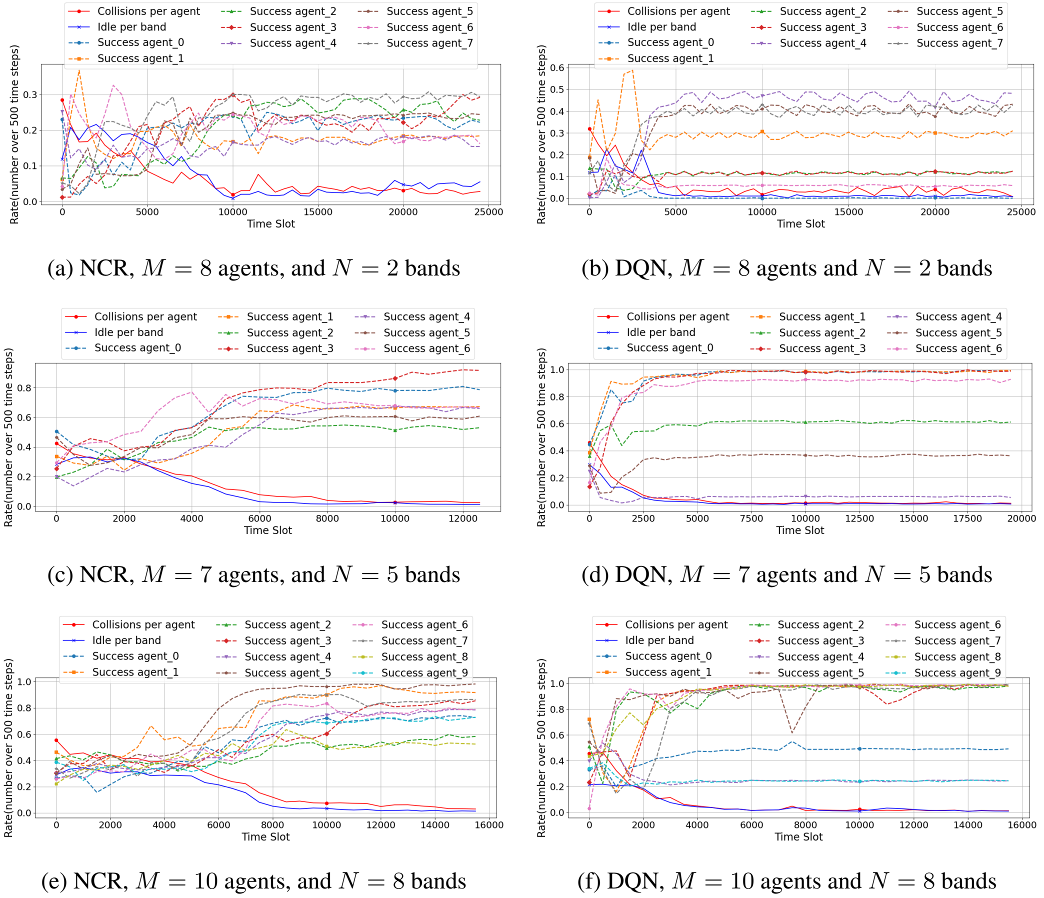
#Agents	#Bands	Deep Q-Network (DQN)		Neuro-Cognitive Radio (NCR)	
M	N	Fairness Index	Total Throughput	Fairness Index	Total Throughput
10	9	0.9000	1.0000	0.9000	0.9913
10	7	0.7000	1.0000	0.7000	0.9866
10	5	0.5000	1.0000	0.5000	0.9844
10	3	0.3000	1.0000	0.3000	0.9713
9	2	0.2222	1.0000	0.2222	0.9760
8	2	0.2500	1.0000	0.2500	0.9860
7	2	0.2857	1.0000	0.2857	0.9730
6	5	0.8333	1.0000	0.8333	0.9872
5	4	0.8000	1.0000	0.8000	0.9925
2	2	1.0000	1.0000	1.0000	0.9860

$M = 10$ and $N = 8$ (Figure 2e), all agents settle at 0.8 each, with minimal variance. Collision and idle rates decrease steadily, contributing to high throughput.

In contrast, in the plots on the right side of Figure 2, DQN agents shows greater disparity in their performance, explaining the lower fairness indices in Table 2. For $M = 8$ and $N = 2$ (Figure 2b), success rates diverge, with some agents dominating at 0.4 – 0.5 while others languishing below 0.2 and agent 0 remaining silent, leading to persistent collisions and uneven resource use. For $M = 7$ and $N = 5$ (Figure 2d), although convergence is better, noticeable gaps remain (e.g., rates from 0.1 to 0.2). For $M = 10$ and $N = 8$ (Figure 2f), agents stabilize with more spread than in SNN, such as some at 0.2. These plots underscore Neuro-Cognitive Radio's superiority in achieving balanced, adaptive spectrum access through neuromorphic principles.

Finally, we conduct a reward-based ablation study. On top of the main reward structure proposed in Eq. 5, we also consider a traditionally used scheme known as Collision Penalty 1 (CP1) [8, 9] defined as $r_m(t) = +3$ if there is a successful transmission, $r_m(t) = -1$ if collision, and $r_m(t) = 0$ otherwise. Table 3 presents such results. As we can see, both DQN and NCR lead to approximately the same fairness and throughput results across all experiments. As expected, given CP1's simple reward structure, the system tends to converge to an equilibrium state in which N agents occupy distinct bands, while the remaining $M - N$ ones remain idle, *i.e.*, J converges to N/M.

We highlight that the only difference between NCR and DQN is that the first utilizes spiking neurons while the second utilizes multi-layer perceptrons. Taken together, the results in Table 2 and Figure 2 suggest that a neuromorphic (spiking) backbone is effective for fair, decentralized DSA: it achieves a favorable fairness–throughput tradeoff while producing stable per-agent behavior suitable for deployment on low-power neuromorphic hardware.



(a) NCR, $M = 8$ agents, and $N = 2$ bands

(b) DQN, $M = 8$ agents and $N = 2$ bands

(c) NCR, $M = 7$ agents, and $N = 5$ bands

(d) DQN, $M = 7$ agents and $N = 5$ bands

(e) NCR, $M = 10$ agents, and $N = 8$ bands

(f) DQN, $M = 10$ agents and $N = 8$ bands

Figure 2: Idle rate, collision rate, and agent's throughput over time t. Comparison between Neuro-Cognitive Radio (left column) and DQN (right column) architectures across different scenarios.

5 Final Remarks

In this paper, we introduced Neuro-Cognitive Radio, a novel neuromorphic-based approach that addresses challenging DSA scenarios by leveraging spiking neural networks to enable efficient, fair, and decentralized spectrum sharing without requiring centralized training or explicit information exchange among agents. Our SNN backbone—built on leaky-integrate-and-fire neurons—incorporates temporal dependencies through a sliding window of past actions and outcomes, achieving superior performance in fairness while maintaining high throughput across diverse network configurations.

Our numerical experiments show that Neuro-Cognitive Radio consistently outperforms the baseline DQN method in Jain's fairness index—for example, improving from 0.6544 to 0.9513 in the scenario with 8 agents and 2 bands—without significant loss in total throughput. This balance is particularly evident in the convergence patterns of agent success rates, where Neuro-Cognitive Radio promotes equitable access even in congested environments. These results underscore the potential of neuromorphic computing for DSA, capitalizing on its event-driven, parallel processing to address the energy and latency constraints of future resource-constrained wireless devices, such as those in vehicular networks or IoT ecosystems.

Looking ahead, Neuro-Cognitive Radio serves as a foundational step toward deploying spiking models on neuromorphic hardware platforms, which could further enhance energy efficiency compared to traditional von Neumann architectures. Future work could explore scaling to larger networks (e.g., hundreds of agents), integrating real-world channel impairments like fading or interference, and hybrid approaches combining SNNs with other RL variants for enhanced robustness. Additionally, empirical validation on physical testbeds would bridge the gap between simulation and practical implementation, paving the way for neuromorphic solutions in next-generation wireless systems.

References

[1] National Telecommunications and Information Administration. "National Spectrum Strategy." 2023. Online: `https://www.ntia.gov/sites/default/files/publications/national_spectrum_strategy_final.pdf`

[2] National Science and Technology Council. "National Spectrum Research and Development Plan." 2024. Online: `https://www.nitrd.gov/pubs/National-Spectrum-RD-Plan-2024.pdf`

[3] International Telecommunication Union. "Report ITU-R SM.2405-1: Spectrum management principles, challenges and emerging issues." 2021. Online: `https://www.itu.int/hub/publication/r-rep-sm-2405-1-2021/`

[4] Guimarães, Francisco R. V., José Mairton B. da Silva Jr., Charles Casimiro Cavalcante, Gabor Fodor, Mats Bengtsson, and Carlo Fischione. "Machine Learning for Spectrum Sharing: A Survey." arXiv preprint arXiv:2411.19032v1, 2024. Online: `https://arxiv.org/html/2411.19032v1`

[5] Lu, Ziyang, Chen Zhong, and M. Cenk Gursoy. "Dynamic Channel Access and Power Control in Wireless Interference Networks via Multi-Agent Deep Reinforcement Learning." *IEEE Transactions on Vehicular Technology*, vol. 71, no. 2, pp. 1588–1601, 2022. doi:10.1109/TVT.2021.3131534

[6] Chang, Hao-Hsuan, Yifei Song, Thinh T. Doan, and Lingjia Liu. "Federated Multi-Agent Deep Reinforcement Learning (Fed-MADRL) for Dynamic Spectrum Access." *IEEE Transactions on Wireless Communications*, vol. 22, no. 8, pp. 5337–5348, 2023. doi:10.1109/TWC.2022.3233436

[7] Wang, Ziyu, Tom Schaul, Matteo Hessel, Hado Hasselt, Marc Lanctot, Nando Freitas. "Dueling Network Architectures for Deep Reinforcement Learning" In Proceedings of the 33rd International Conference on Machine Learning, PMLR 48:1995-2003, 2016.

[8] Naparstek, Oshri, and Kobi Cohen. "Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access." *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2019. doi:10.1109/TWC.2018.2879433

[9] Yu, Yiding, Taotao Wang, and Soung Chang Liew. "Deep-Reinforcement Learning Multiple Access for Heterogeneous Wireless Networks." *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1277–1290, 2019. doi:10.1109/JSAC.2019.2904329

[10] Sohaib, Muhammad, Jongjin Jeong, and Sang-Woon Jeon. "Dynamic Multichannel Access via Multi-Agent Reinforcement Learning: Throughput and Fairness Guarantees." *IEEE Transactions on Wireless Communications*, vol. 21, no. 6, pp. 3994–4008, 2022. doi:10.1109/TWC.2021.3126112

[11] Xu, Yue, Jianyuan Yu, and R. Michael Buehrer. "The Application of Deep Reinforcement Learning to Distributed Spectrum Access in Dynamic Heterogeneous Environments With Partial Observations." *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4494–4506, 2020. doi:10.1109/TWC.2020.2984227

[12] Bokobza, Yoel, Ron Dabora, and Kobi Cohen. "Deep Reinforcement Learning for Simultaneous Sensing and Channel Access in Cognitive Networks." *IEEE Transactions on Wireless Communications*, vol. 22, no. 7, pp. 4930–4946, 2023. doi:10.1109/TWC.2022.3230872

[13] Janiar, Siavash Barqi, and Vahid Pourahmadi. "Deep-Reinforcement Learning for Fair Distributed Dynamic Spectrum Access in Wireless Networks." In *Proceedings of CCNC*, 2021.

[14] Zhang, Yubo, Pedro Botelho, Trevor Gordon, Gil Zussman, and Igor Kadota. "Fair Dynamic Spectrum Access via Fully Decentralized Multi-Agent Reinforcement Learning." arXiv preprint arXiv:2503.24296 (2025).

[15] Schuman, Catherine D., Shruti R. Kulkarni, Maryam Parsa, J. Parker Mitchell, Prasanna Date, and Bill Kay. "Opportunities for neuromorphic computing algorithms and applications." Nature Computational Science 2, no. 1 (2022): 10-19.

[16] Eshraghian, Jason K. "Deep Learning with Spiking Neural Nets." In Selected Topics in Intelligent Chips with Emerging Devices, Circuits and Systems, pp. 164-180. River Publishers, 2023.

[17] Parpart, Gavin, Sumedh Risbud, Garrett Kenyon, and Yijing Watkins. "Implementing and benchmarking the locally competitive algorithm on the loihi 2 neuromorphic processor." In Proceedings of the 2023 International Conference on Neuromorphic Systems, pp. 1-6. 2023.

[18] Hunsberger, Eric, and Chris Eliasmith. "Spiking deep networks with LIF neurons." arXiv preprint arXiv:1510.08829 (2015).

[19] Chowdhury, Sayeed Shafayet, Deepika Sharma, Adarsh Kosta, and Kaushik Roy. "Neuromorphic computing for robotic vision: algorithms to hardware advances." Communications Engineering 4, no. 1 (2025): 152.

[20] Tenzin, Sangay, Alexander Rassau, and Douglas Chai. "Application of event cameras and neuromorphic computing to VSLAM: A survey." Biomimetics 9, no. 7 (2024): 444.

[21] Gupta, Abhishek, Aldo Pacchiano,, Yuexiang Zhai, Sham M. Kakade, and Sergey Levine,"Unpacking reward shaping: Understanding the benefits of reward engineering on sample complexity," arXiv preprint arXiv:2210.09579 (2022).

[22] Intel Corporation. "Intel Launches World's Largest Neuromorphic Supercomputer as Brain-Inspired AI Goes Mainstream." 2025. Online: https://semiconductorinsight.com

[23] R. Jain, D.-M. Chiu, and W. Hawe, *A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Systems*, Technical Report DEC-TR-301, 1984.