Policy-based Fully Spiking Reservoir Computing for Multi-Agent Distributed Dynamic Spectrum Access

Nima Mohammadi, Lingjia Liu and Yang Yi

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838849

***Abstract*—In the midst of the machine learning revolution, there is hope to thrive the ever-growing demand for limited spectrum resources imposed by the growth of wireless devices with a paradigm shift to more intelligent ways to manage and share the radio spectrum. This requirement mandates very energy-efficient solutions that can tackle the rapid changes of the wireless environment. This work considers spiking neural networks, which have been shown to drastically reduce the energy consumption compared to conventional neural networks in a reinforcement learning setup designed for the dynamic spectrum sharing scenario. Moreover, the temporal aspect of the problem and the necessity of sample efficiency motivates incorporating liquid state machines into this design. However, the agents' state- and time-variant inputs impose a burden of a posteriori hyperparameter optimization for liquid state machines, rendering the deployment of reliable models whose reservoirs operate in favorable regimes very challenging in such a setting. Therefore, we employ a homeostatic learning rule for adaptively tuning small-world reservoir connections to maintain near-chaotic behavior during operation. Simulation results prove the performance of the introduced solution compared with several existing techniques.**

***Index Terms*—Dynamic Spectrum Access, Spiking Neural Networks, Neuromorphic Computing, Multi-Agent Reinforcement Learning, Bio-inspired Computing**

I. Introduction

With the advent and deployment of 5G technology, there are already discussions about beyond 5G wireless systems with the main distinction of employing AI-based solutions at every layer. The new generation of wireless technology promises to extend the benefits of AI, with reinforcement learning (RL) playing a major role. RL has been widely shown to be capable of solving many information processing problems in areas, including but not limited to autonomous robot control, marketing strategy optimization, and wireless communications and networking [1], [2].

It is no mystery that spectrum is presently regarded as one of the most valuable commodities. The value of the wireless spectrum goes higher as it inevitably becomes more and more scarce. Spectrum sharing, first explored through the concept of Cognitive Radio, is an attempt to prevent underutilization of the wireless spectrum by allowing access to license-exempt secondary users (SUs) conditional on interference protection to primary users (PUs). However, traditional CR was sought insufficient by both the uncertain operators and SUs who could at best achieve a QoS similar to unlicensed spectrum. Since the inception of CR, flexible spectrum access has been studied in more depth. Similar to many other disciplines, Machine Learning has found its way to enable more efficient and effective access to the spectrum. In a Dynamic Spectrum Access (DSA) scenario, users become aware of their surrounding spectrum usage via a procedure known as spectrum sensing. Across a wideband of spectrum, it becomes more critical for the sensing time to be optimally utilized [1]. To this end, we employ a policy-based RL algorithm that allows the users to dynamically and autonomously adjust their operation.

After formulating DSA as an RL problem, there are choices to be made regarding the RL framework and the network architecture in the case of deep reinforcement learning (DRL), among many. Deep Q-networks have shown to be able to tackle many reinforcement learning problems, even surpassing the human-level performance in many complex tasks. This is achieved by combining the Q-learning framework [3] with neural networks, allowing for efficient learning of policies over complex large-scale state and action spaces. For DSA, the authors of [2] investigate two variants of DQN, namely deep recurrent Q-networks (DRQN) and deep echo state Q-networks (DEQN), where DEQN with echo state networks (ESNs) is shown to provide a better solution given the limited amount of training data available for the highly dynamic environment.

Neural networks are computationally and memory-wise intensive operations to be performed on data, but their efficiency aspects are often outshined by their accuracy. Having limited power budget, especially on battery-operated devices, puts the viability of DEQN and DRQN for DSA into question. In this work, we introduce a model built on spiking neural networks (SNNs) and neuromorphic computing (NC), which alleviates this issue by taking a rather more biological approach toward computation that diverges from the ubiquitous von-Neumann architecture by coupling memory and computational units and operating on information conveyed through rare discrete events. We show that our introduced energy-efficient design outperforms existing methods in both energy consumption and achieved throughput. To this end, we take the innovative approach of employing a fully-spiking model with liquid state machines (LSM) to take into account the temporal aspect of the problem and a homeostatic mechanism to make sure the LSM reservoir does not move into undesirable regimes.

The remainder of this paper is organized as follows: Section II describes the system model. Section III-A and III-B introduce the RL approach and formulate the problem. Section III-C describes the neural model and coding. Section III-D de-

N. Mohammadi and L. Liu are with Wireless@VT, and along Y. Yi are with Bradley Department of Electrical and Computer Engineering at Virginia Tech. This work is supported in part by the National Science Foundation (NSF) under grants CCF-1937487 and CNS-2003059. The corresponding author is L. Liu (ljliu@ieee.org).

scribe the spiking reservoir network and the regulatory mechanism. In section IV we apply this method to construct a CR network and draw comparisons with other competing model. Finally, we conclude and some future works are presented.

II. Problem Definition

For our DSA setup, we assume a wireless environment comprised of M primary users, each with a dedicated wireless channel. It is also assumed that cross-channel interference is negligible. Furthermore, the environment includes N secondary users competing to gain access to the M channel such that low interference is caused to the PUs. This work aims to introduce an energy-efficient adaptive access strategy for SUs that aims to maximize spectrum utilization while minimizing the undesirable interference to the operation of the primary network.

The goal of UEs is to transmit the data from the transmitter to the receiver over the wireless link in the span of discretized time slots. To this end, the radio should know its current performance, for which signal-to-interference and noise ratio (SINR) is employed. Then, based on this quality measure, the agent would reflect on its previous actions and their results. The SINR obtained by user i on channel m at time slot $t \in \mathbb{N}$ is defined as the ratio of the power of the desired signal of interest in the numerator to the sum of the interference power and background noise, written as

$$\mathrm{SINR}_m^i[t] = \frac{P^i \cdot \left|H^i[t]\right|^2}{\sum_{j \in \Phi_m, i \neq j} P^j \cdot \left|H^{ji}[t]\right|^2 + N_m} \tag{1}$$

where P^i denote the transmit power of user i and $\Phi_m \subseteq [N+M]$ is a set of users transmitting on channel m. $H^i[t]$ and $H^{ji}[t]$ denote the time-varying channel gains of the desired link for user i and the interference link between the user j's transmitter and the user i's receiver. In addition to this, we have N_m as the background noise. Upon low SINR, the PU will broadcast a warning signal, which may be due to either low channel gain of desired link (i.e., small numerator) or strong interference from other users (i.e., large denominator).

The reason DSA works is that the PUs usually do not fully utilize their licensed part of the spectrum giving rise to the so-called 'spectrum holes.' Then, the aim is to detect these spectrum holes, which can be in turn temporarily used by the SUs. However, in highly dynamic wireless networks, it is not realistic for SUs to obtain information regarding the activation of the PUs or the competing SUs. Instead, the SUs have to rely on spectrum sensing to avoid transmission over occupied channels. However, many factors, including the quality of the wireless links, the transmission powers of other transmitters, and the background noise, influence spectrum sensing and prevent it from being an accurate indicator of spectrum opportunities. To make matters worse, all of these contributing factors are time-variant, meaning that the obtained information can quickly become outdated.

Spectrum sensing is performed by the SUs, prior to transmission, in order to prevent interference to the active PUs. Unfortunately, power and complexity constraints prevent sensing multiple channels at the same time. We assume that a SU is only able to sense and operate on a single channel over each period T but may freely move to other channels across periods. We base the identification of spectrum opportunities on the commonly-used approach of energy detection. The energy of signal on channel m received by SU i is calculated as,

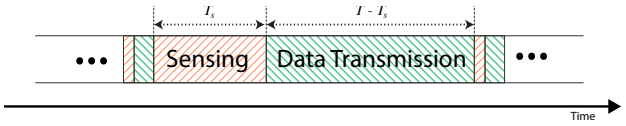

Fig. 1. Alternating operation modes of half-duplex SU agents over time.

$$E_m^i[t] = \sum_{t'=t}^{t+T_s-1} \left|y_m^i[t']\right|^2 \tag{2}$$

where t denotes the starting time of the period and T_s is the sensing duration. $y_m^i[t']$ is the received signal at time t'. The sensing procedure is followed by $T - T_s$ time slots for data transmission, as depicted in Fig. 1. The spectrum sensing and the data transmission are alternatively and iteratively repeated. The received signal $y_m^i[t']$ depends on activity state and power of PU m, the sensing link between the transmitters of PU m and SU i, and the background noise, represented as

$$y_m^i[t'] = \begin{cases} \sqrt{P^m} \cdot H^{mi}[t'] + \omega_m[t'] & \text{Active PU } m \\ \omega_m[t'] & \text{Inactive PU } m \end{cases} \tag{3}$$

where P^m is the transmit power of PU associated to the channel, $H^{mi}[t]$ is the channel gain of the sensing link and $\omega_m[t'] \sim \mathcal{CN}(0, N_m)$ is the background noise.

Finding a threshold value that distinguishes the active and inactive state of a PU m can be very challenging for a SU due to the fact that it depends on many time-variant factors such as the channel gain of the sensing link and the random background noise that inherently brings uncertainty to the process. Furthermore, the sensing link is between the PU and the SU transmitters, whereas the interference link is between the PU transmitter and the SU receiver. This along the underlying temporal correlation of DSA networks motivates us to incorporate AI-based solutions to increase the overall utilization of the wireless network by better identifying the spectrum opportunities that may be used by the secondary users.

III. Methodology

In this section we elaborate on various components and design choices that bring us to the introduced model, starting with the RL framework before proceeding with the description of the underlying SNN, neural coding and the resevoir component.

A. Reinforcement Learning Framework

RL refers to goal-oriented machine learning algorithms that train themselves via interaction with an environment. The algorithm/agent performs actions and modifies its behavior based on the feedback (reward or punishment) it receives. This

scenario, which can naturally describe the decision-making of SUs, is formulated as a Markov Decision Process (MDP) comprised of the quintuple $(\mathcal{S}, \mathcal{A}, P, R,$ and $\gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the discrete action space, R is the reward function providing the immediate observed reward r_t for taking action $a_t \in \mathcal{A}$ while at state $s_t \in \mathcal{S}$, and $\gamma \in [0, 1]$ is the discount factor. In addition, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ provides the transition probability $P(s_{t+1} \mid s_t, a)$. The agent follows a policy $\pi(s_t)$ to select action a_t at each step to maximize the objective function

$$J(\pi) := \mathbb{E}\left[\Sigma_{t=0}^{T} \gamma^t r_t \mid \pi\right] \tag{4}$$

denoting the expected cumulative reward following a policy π. Then, an optimal policy $\pi^\star$ is any policy such that

$$\pi^\star \in \arg\max_{\pi \in \Pi} J(\pi). \tag{5}$$

In value-based Q-learning, e.g., DEQN and DRQN, the goal is to learn a state-action value function $Q(s_t, a_t)$ that guides the implicit policy[1] in choice of actions by estimating the expected return of each state-action pair. DQN, an extension of Q-learning, parameterizes and models the $Q(.)$ function via a neural network. Alternatively, instead of learning the Q-function one may directly parameterize the policy and adjust it toward improvement of the performance irrespective of the value function. In this work, we opt for the second approach, where the policy network is a spiking neural network parameterized by θ and adjusted in proportion to the gradient of the performance to maximize $J(\pi_\theta)$ via gradient ascent,

$$\theta_{i+1} \leftarrow \theta_i + \eta \cdot \nabla_\theta J(\pi_{\theta_i}) \tag{6}$$

where $\eta > 0$ is the step size. Our method is classified as a policy-gradient (PG) algorithm and is able to learn stochastic policies, is more effective with a high-dimensional or continuous action space and typically converges faster, although often prematurely [4]. Furthermore, another advantage of our technique is that it would need half the model size compared to DEQN and DRQN setups, with one single policy network instead of the two separate evaluation and target networks of DQN, which is again aligned with our efficiency concerns.

B. Formulation of the DSA problem for RL setup

To adopt an RL solution for the dynamic spectrum sharing problem, we need to define the inputs and outputs of each agent, including the states determined by the observations made from the environment, the actions selected by the agents, and the reward function that stipulate how well the agent accomplishes its task. At the period $n \in \mathbb{Z}^+$, the state of SU i is denoted by

$$s^i[n] = \left(E^i[n], C^i[n]\right) \tag{7}$$

where $E^i[n]$ is the received signal energy for the sensing time slots $[nT \,..\, nT + T_s)$ of the sensed channel indicated by the categorical variable $C^i[n]$ with cardinality M. The energy part of the state vector, $E^i[n]$, is calculated by $E^i_m[nT]$ in Eq. (2). Furthermore, the action of the SU i at nth period is denoted as

$$a^i[n] = \left(o^i[n], p^i[n]\right) \tag{8}$$

where $o^i[n] \in \{$'Active', 'Idle'$\}$ represents the SU i mode of operation during the transmission time slots $[nT+T_s \,..\, (n+1)T)$ in period n. In addition, $p^i[n] \in [1 \,..\, M]$ indicates which channel should be sensed during the sensing time slots of the next period, $[(n+1)T \,..\, (n+1)T + T_s)$. That is, the logic is divided into two parts, with $p^i[n]$ and $o^i[n]$ taking account for channel switching and decision to access or stay idle, respectively.

We design a reward signal based on the possible interference to PUs and the achieved modulation and coding scheme (MCS) of 3GPP LTE/LTE-A. MCS depends on the link quality, measured here via SINR, which determines the channel quality indicator (CQI), and indicates how many useful bits are transmitted per symbol.

During the transmission time slots $[nT + T_s \,..\, (n+1)T)$, the average spectral efficiency of a primary or a secondary user i operating on channel m is calculated by

$$\eta^{PU/SU}_{i,m}[n] = \frac{1}{T - T_S} \sum_{t'=nT+T_s}^{(n+1)T-1} e^i_m\left[t'\right] \tag{9}$$

where e^i_m denotes the spectral efficiency of user i on channel m at time slot t'.

$$r^i[n] = \begin{cases} -2, & \text{if } \eta^{\text{PU}}_m[n] < \tau \\ -1, & \text{else if SU } i \text{ is idle} \\ min\{3, \lfloor \eta^{\text{SU}}_i[n] \rfloor\} & \end{cases} \tag{10}$$

where τ is the threshold below which the PU m will have emitted a warning signal. Besides receiving the warning signal, which as mentioned is not necessarily due to collision but possibly low quality of the wireless link, an SU staying idle is also being penalized by receiving a negative reward to encourage the agent to seek spectrum opportunities. In case the corresponding PU does not experience intolerable interference from any SU, the reward signal $r^i[n]$ will be a positive integer value within the $[0 \,..\, 3]$ range, monotonically increasing with the average spectral efficiency of SU i.

C. Supervised Spiking Neural Model

The policy network is composed of a reservoir, delineated in the next section, whose output is fed to a feed-forward SNN, referred to as the readout model, with one intermediate layer. Spiking neurons are computational units that perform temporal and spatial weighted integration of spikes coming through their synaptic connections. The output spiking activities of a neuron at time t can be written as $S(t) = \Theta(V_{mem}(t) - V_{th})$, where V_{th} is the firing threshold potential above which a spike is emitted, and Θ is the Heaviside step function. Among many neural models, the widely-used phenomenological model, the leaky integrate and fire (LIF), is chosen for this work. LIF

[1]Starting with an ϵ-greedy policy to allow selection of initial exploratory actions, and gradually becoming a deterministic policy.

neurons provide a good balance between computational efficiency and biological plausibility, where the frequency and timing of the spikes are assumed to carry the information, and the shape and amplitude of the spikes are discarded, giving rise to rich temporal dynamics with binary signals. As a result of the event-driven nature of SNNs, with the extremely low-consumption idle state, the energy consumption is mainly determined by the number of spikes generated by its neurons. Given the sparse coding schemes, this allows SNNs to reduce the energy consumption compared to ANNs drastically.

Compared to ANNs, the training of SNNs, i.e. adjusting the synaptic weights, is much more challenging and has been the topic of many studies. In this work, we employ Approximate Gradient Descent (AGD) which offers a backprop-like algorithm to train the readout model. Mature training algorithms for ANNs such as backpropagation [5] cannot be exploited by SNNs due to the non-differentiable spike activities and also the fact that spiking neurons operate in the temporal domain. To this end, we employ Surrogate Gradients (SG), which take care of non-differentiability, in combination with backpropagation through time (BPTT) algorithm. The main idea is to use different so-called 'surrogate' gradients for parameter optimization and overcome the vanishing gradient problem, instead of actually changing the nonlinearity itself [6].

Besides the training mechanism for the SNN, a vital aspect of NC systems is the neural encoder, which transforms sensory information into spike trains. Different coding schemes have been proposed with different impacts on energy consumption and overall performance of the model [7]. We test our design with both Poisson rate coding and inter-spike-interval temporal coding, where the information is carried via the firing rate or through the time intervals between consecutive spikes, respectively.

D. Liquid State Machines

Reservoir Computing is an umbrella term used to describe the computational framework developed independently under the names of Echo State Network (ESN) and Liquid State Machine (LSM) [8]. While both architectures share a core idea, they are distinguished by the neural model they use, with LSMs using the spiking neural, hence being more energy efficient. Both models are composed of a population of neurons recurrently interconnected, giving rise to three layers, namely the input layer W_I, the reservoir W_R, and the output/readout layer W_O. In reservoir computing, only the readout layer gets trained, while the input and the reservoir layers are randomly initialized and fixed. The intuition for constraining the supervised training process to only the last layer was born from the observation that in training RNNs, the weights showing most change are the ones in the last layer. This consequently makes reservoir computing more efficient in terms of training time and a promising solution for the emerging field of Neuromorphic Computing.

Although foregoing training on the connections within the reservoir brings gains to accuracy and efficiency, finding proper values for the hyperparameters of the reservoir can still be a challenge. The topology and the initialization mechanism of the reservoir weights can significantly impact the performance of the network, and careless selection of them may render the whole model useless. Hyperparameter optimization incurs high computational cost and needs to be performed before employment of the model, making it impractical to incorporate into our design with space- and time-variant inputs.

The problem with an improper W_R is the emergence of supercritical or subcritical dynamics, which during recursion leads to explosion or fading of the internal states, respectively. To circumvent this issue, for echo state networks, certain algebraic properties of the transition matrix W_R have to be satisfied, such as echo state property (ESP) which guarantees that the effect of initial conditions should vanish as time proceeds. Unfortunately, similar sufficient properties for LSMs have not been identified, and the commonly used evolutionary algorithms for searching hyperparameter space of LSMs would not be applicable for deploying our DSA solution on specialized hardware.

Although topology still needs to be carefully chosen, fixing the connectivity pattern to a small-world-like topology, found in cortical neural circuits and shown to enhance ESP [9], allows us to only focus on the weights. In this work, we employ the newly proposed P-Critical homeostasis that enables tuning of the mean branching factor $\bar{\sigma}$ during runtime [10]. The branching factor of a neuron spiking n_{pre} times with n_{post} post-synaptic spikes is defined as $\sigma = n_{\text{pre}} / n_{\text{post}}$. A $\bar{\sigma}$ slightly below one can bring the network to the desired edge of chaos. To this end, P-Critical augments the network with auxiliary neurons that causes depreciation of growing synaptic weights of associated neurons to regulate the branching factor.

IV. Experiment

A. Setup

The channel gains are generated based on the B1 scenario of the WINNER II channel model, which describes an environment of Manhattan-like streets with outdoor antennas whose heights are below the tops of surrounding buildings [11]. The number of PUs and SUs are set to $M = 4$ and $N = 6$, respectively, with the UEs randomly scattered on a $2000m \times 2000m$ plane and a constraint on the distance between the corresponding transmitter-receiver pairs to be in the range of $400 - 450$ meters. In this setup, there are M desired links for PUs, and a total of $N \times N$ links between pairs of SUs, $N^2 - N$ of which are interference links. Also, there are $2 \times M \times N$ interference links from SUTs to PURs, and from PUTs to SURs. As described in section II the cross-channel interference is assumed to be negligible, hence the omission of these links. Finally, there are $M \times N$ sensing links from PUTs to SUTs.

The bandwidth of each channel is set to 5 MHz with random Gaussian noise generated 17 dB above the -174 dBm thermal noise floor. Each period T takes ten time slots, with the first two time slots allocated to sensing, where each slot takes one millisecond. The transmission powers of both PUs and SUs are

set to 500 mW. Two PUs, namely PU1 and PU3, are activated every $3T$ periods, whereas the remaining two are activated every $4T$ periods. The simulation runs for 600 seconds, that is a total of 60,000 time slots.

B. RL Agent

Each SUT is assigned an RL agent that interacts with the environment and adapts its policy based on its observations. The same learning rate and $\epsilon-$greedy exploration schedules were employed as [2]. For the sake of comparison, the number of neurons in LSTM kernel of DRQN and the number of units in reservoirs of DEQN and our introduced method are all set to 32.

For our method, the RL agent is composed of two spiking neural networks, a recurrent reservoir and feed-forward network. The reservoir neurons are arranged in a tri-dimensional space, forming minicolumns that are connected to each other in macrocolumnar organization. The synapses within the reservoir are realized randomly where the probability of two neurons being connected is related to their Euclidean distance, following a small-world topology exhibited in cortical neural connectivity.

C. Simulation Results

The main objective of training RL agents for DSA is to increase the throughput of SUs without harming the throughput of the PUs. The throughput being the number of transmitted bits per second, we define the system throughputs for PUs and SUs as the sum of throughputs for users within the corresponding group. This goal is achieved by SUs predicting the occupancy of communication channels and attempting to access the ones that will not cause interference.

We compare the results of our introduced model, denoted in this section as PG+LSM, with DRQN (LSTM Q-network) and DEQN (ESN Q-network).

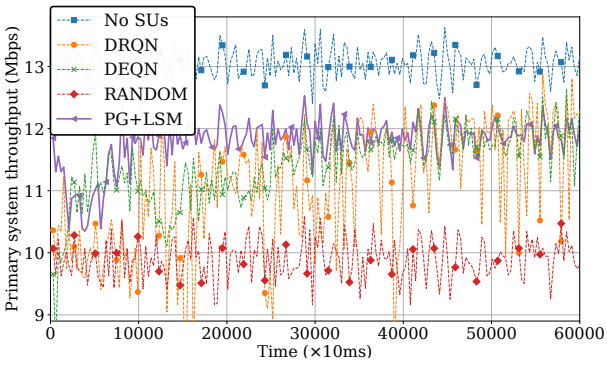

Fig. 2. Comparison of system throughput of PUs

The system throughputs of PUs and SUs are depicted in Fig. 2 and Fig. 3, respectively. For better comparison, we have plotted the PU throughputs in the absence of any secondary user as an upper limit. We also include curves for the case where the policy of SUs is to randomly choose an action, as a baseline. We expect for the agents to interact with the environment,

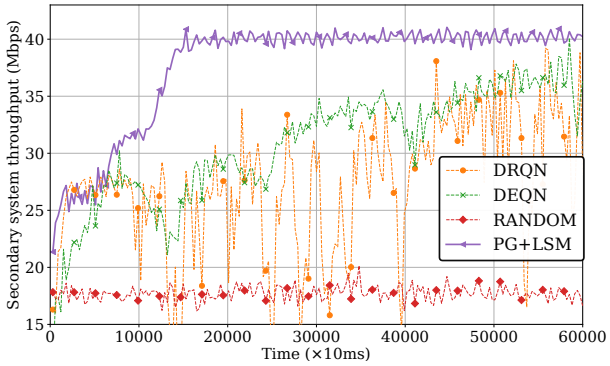

Fig. 3. Comparison of system throughputs of SUs

tune their parameters, and observe their performance, behaving similar to the random baseline and approaching the case where PUs have no interference from SUs. From these plots, we observe that our method outperforms DRQN and DEQN with regard to both performance indicators. While the PU throughput of our model and DEQN reach the same level, our model reaches that at a faster pace. However, the improvement is more pronounced in the case of SU throughputs, where the fast convergence of our method promises quicker adaptability resulting in higher data rates.

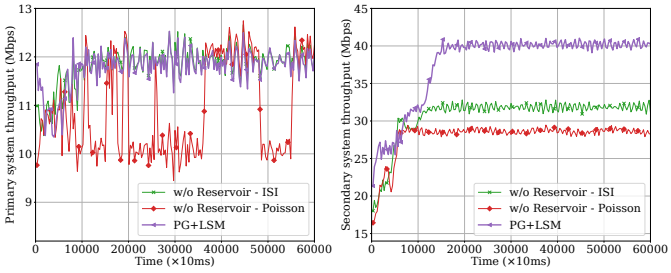

Fig. 4. The performance of SNN-based policy networks with and without Liquid State Machines.

To see the impact of having a reservoir with the P-CRITICAL regulatory mechanism, we compare three runs in Fig. 4. One of the advantages of AGD in training an SNN is its flexibility with various neural codings, and our observations with more than 6000 runs show that this also applies to the case with LSMs, despite the drastically different modes of operation for Poisson rate encoding and ISI temporal encoding. The result of our model reported here is obtained with ISI encoding to transform the stimuli to spike trains, however similar architecture with rate coding achieve roughly the same performance level. Fig. 4 indicates that having an LSM has a staggering effect on the performance. Furthermore, Fig. 5 depicts how the introduced model has caused lower warnings to be emitted by PUs during the 60000 simulated time slots compared to the competition.

D. Energy-Efficiency Analysis

We assume a half-duplex SU system where the SU may not perform transmission and spectrum sensing simultaneously.

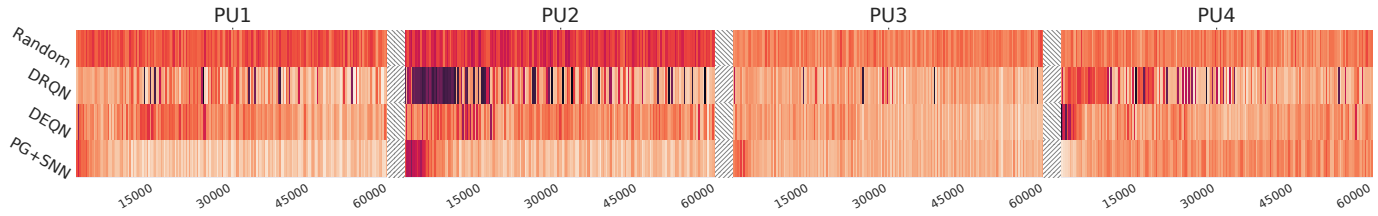

Fig. 5. PU warning frequencies across time. Darker stripes indicate more warnings emitted by the primary user.

The periodic operation of a SU consists of a T_s sensing period followed by a $T - T_s$ transmission period where the SU may be either on or in sleep mode. In other words, the total energy consumption of the SU is defined as

$$\begin{aligned} E_{total} &= E_{sensing} + E_{on} + E_{sleep} \\ &= P_{sensing}T_{sensing} + P_{on}T_{on} + P_{sleep}T_{sleep} \end{aligned} \tag{11}$$

where $E_{sensing}$ would depend on the inference energy of the underlying neural network. With the all-or-nothing pulses, the SNNs replace the expensive multiply-accumulate (MAC) operations with cheaper accumulate (AC) operations [12]. The energy-saving implications of SNNs are also rooted in various coding schemes that result in sparse activation patterns across the network, for which we have relied on ISI encoding [13]. Although AC operations required for SNNs consume tens of times less than MACs for conventional neural networks ($E_{MAC} = 3.2$pJ and $E_{AC} = 0.1$pJ), note that the input action potentials are presented over a period of T_{SNN} timesteps in SNNs [14]. Higher T_{SNN} allows the function approximation of rate-encoded SNN to approach the ANNs' but decreases the energy-efficiency gains since higher spike counts and consequently FLOPS are required for higher-intensity inputs. The temporal encoding maintains the frequency of actions potentials and makes better use of the simulation time steps, eventually lowering the latency. The energy consumption ratio E_{ANN}/E_{SNN} can be calculated as $E_{MAC}/(E_{AC} \times A \times T)$. In our simulation with latency of $T = 10$ timesteps, $A \approx 0.05\%$ active neurons are found at each step, hence a $\sim$60 times improvement with regard to energy consumption.

V. Conclusions and Future Works

In this article, we introduced a fully-spiking reinforcement learning scheme powered by liquid state machines for dynamic spectrum sharing and access scenarios. The model is shown to exceed more than one order of magnitude improvement in energy consumption, and also to outperform state-of-the-art methods with regard to utilization of the spectrum. We showed that homeostatic plasticity could warrant operational LSMs with small-world-like topologies, in lieu of cost-prohibitive criteria, to learn a proper spectrum access strategy in the highly dynamic wireless networks.

One clear advantage of policy-based RL methods is amenability for continuous-control problems. In this regard, we may devise a more flexible collaborative spectrum access system that also controls the transmission power of agents. In Spatial Spectrum Sensing, we attempt to increase the number of agents simultaneously transmitting on a single channel by minimizing the coverage overspill of secondary transmitters. To this end, we should have fine-grained control over the transmission power. Such extension would increase spectrum utilization, decrease the power consumption of each device, and would be aligned with our attempt at energy efficiency via spiking neural networks. Furthermore, a federated learning scheme [15], realization of our model on neuromorphic architectures and evaluation on a USRP testbed, are ongoing works.

References

[1] H.-H. Chang, H. Song, Y. Yi, J. Zhang, H. He, and L. Liu, "Distributive dynamic spectrum access through deep reinforcement learning: A reservoir computing-based approach," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1938–1948, 2019.

[2] H.-H. Chang, L. Liu, and Y. Yi, "Deep echo state q-network (DEQN) and its application in dynamic spectrum sharing for 5g and beyond," *IEEE Trans. Neural Netw. and Learning Syst.*, 2020.

[3] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[4] L. Wang, Q. Cai, Z. Yang, and Z. Wang, "Neural policy gradient methods: Global optimality and rates of convergence," *arXiv preprint arXiv:1909.01150*, 2019.

[5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[6] E. O. Neftci, H. Mostafa, and F. Zenke, "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks," *IEEE Signal Process. Mag.*, vol. 36, no. 6, pp. 51–63, 2019.

[7] H. Zheng, N. Mohammadi, K. Bai, and Y. Yi, "Low-power analog and mixed-signal ic design of multiplexing neural encoder in neuromorphic computing," in *22nd Intl Symp. Quality Electronic Design (ISQED)*, pp. 154–159.

[8] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, 2002.

[9] Y. Kawai, J. Park, and M. Asada, "A small-world topology enhances the echo state property and signal propagation in reservoir computing," *Neural Networks*, vol. 112, pp. 15–23, 2019.

[10] I. Balafrej and J. Rouat, "P-critical: A reservoir autoregulation plasticity rule for neuromorphic hardware," *arXiv:2009.05593*, 2020.

[11] Y. d. J. Bultitude and T. Rautiainen, "Ist-4-027756 winner ii d1. 1.2 v1. 2 winner ii channel models," *EBITG, TUI, UOULU, CU/CRC, NOKIA, Tech. Rep*, 2007.

[12] B. Rueckauer and S.-C. Liu, "Conversion of analog to spiking neural networks using sparse temporal coding," in *2018 IEEE Intl Symp. on Circuits and Syst. (ISCAS)*, pp. 1–5.

[13] C. Zhao, B. T. Wysocki, C. D. Thiem, N. R. McDonald, J. Li, L. Liu, and Y. Yi, "Energy efficient spiking temporal encoder design for neuromorphic computing systems," *IEEE Trans. Multi-Scale Comput. Syst.*, vol. 2, no. 4, pp. 265–276, 2016.

[14] P. Panda, S. A. Aketi, and K. Roy, "Toward scalable, efficient, and accurate deep spiking neural networks with backward residual connections, stochastic softmax, and hybridization," *Frontiers in Neuroscience*, vol. 14, 2020.

[15] N. Mohammadi, J. Bai, Q. Fan, Y. Song, Y. Yi, and L. Liu, "Differential privacy meets federated learning under communication constraints," *IEEE Internet of Things Journal*, 2021.