Learning-based Spectrum Sensing in Cognitive Radio Networks via Approximate POMDPs

Bharath Keshavamurthy, Nicolò Michelusi

Abstract

In this paper, a novel spectrum sensing and access strategy is proposed, wherein a cognitive radio learns the time-frequency correlation model defining the occupancy behavior of incumbents via the Baum-Welch algorithm, and concurrently devises an optimal spectrum sensing and access strategy that exploits this learned correlation model, under spectrum sensing constraints. The optimal access and sensing strategy is optimized via an approximate point-based Partially Observable Markov Decision Process (POMDP) method with fragmentation and Hamming distance state filters, which facilitates control of the trade-off between secondary network throughput and incumbent interference. Numerical results demonstrate improvements of 60% over correlation-coefficient based clustering, 25% over Neyman-Pearson Detection, 6% over Viterbi, and 7% over adaptive deep Q-network. **[NM: I believe that DARPA is for the single agent case correct? Whereas EPS32 is on multiagent? In that case i would move as follows..]** The proposed single-agent scheme is implemented on the DARPA Spectrum Collaboration Challenge SC2 platform, demonstrating superior performance over competitors on a real-world TDWR-UNII WLAN scenario emulation. The proposed solution is extended to a distributed multi-agent settings with neighbor discovery and channel access rank allocation, which delivers 43% boost over cooperative time difference-SARSA, 84% over greedy distributed learning under pre-allocation**[NM: what do you mean?]**, and 324% over distributed opportunistic g-statistics with ACKs**[NM: too many details, readers will not be able to follow unless they happen to know which specific scheme you are talking about. Can you rephrase in a more familiar way that can be understood by most readers?]**. Finally, the implementation feasibility of the proposed multi-agent spectrum sensing and access **[NM: do you also have access or only sensing with ESP32?]** is demonstrated on an ad-hoc distributed wireless platform of ESP32 radios.**[NM: and? what does this implementation demonstrate?]**

Index Terms

Part of this research has been submitted to IEEE ICC 2021.**[NM: provide reference in the biblio and cite here]**

Part of this research has been funded in part by NSF under grant CNS-1642982.

The authors are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN.

Email: {bkeshava, michelus}@purdue.edu

Hidden Markov Model, Cognitive Radio, Spectrum Sensing, POMDP

I. Introduction

Cognitive radios have been touted as instrumental in solving resource-allocation problems in resource-constrained radio environments. The adaptive computational intelligence of these radios facilitates the dynamic allocation of scarce network resources, particularly, the spectrum. With fifth-generation (5G) cellular technologies in widespread deployment today [**?**], [**?**], a multitudinous array of devices have been or will be brought into the wireless communication ecosystem, resulting in an enormous strain on the available spectrum resources. Dynamic Spectrum Access (DSA), the key defining feature of cognitive radio networks, is being widely studied as a solution to the problem of spectrum scarcity, in both military and consumer spheres: cognitive radios intelligently access portions of the spectrum that are left unused by the sparse and infrequent transmissions of licensed users in the network, in order to deliver their own network flows, while adhering to non-interference compliance requirements laid down by bureaucratic agencies such as the Federal Communications Commission (FCC) [**?**], [**?**].

[NM: this is quite contorted..please rephrase.. Also, I think it should be moved at the end of this paragraph, after you talk about the open challenges. Otherwise, it is not clear why you are using POMDP in the first place] From an independent cognitive radio perspective, our solution to the spectrum sensing and access problem in the Medium Access Control (MAC) layer of a cognitive radio node, referred to as a Secondary User (SU), sharing a discretized multi-channel AWGN radio environment with several licensed users, involves a Partially Observable Markov Decision Processes (POMDP) formulation. As alluded to earlier,**[NM: no need to repeat the same thing that you just repeated in the previous paragraph..]** a cognitive radio facilitates efficient spectrum utilization by intelligently accessing unused spectrum holes across both time and frequency known as "spectrum white spaces," in order to deliver its network flows while limiting interference to the priority or licensed users (incumbents), referred to as Primary Users (PUs)**[NM: repetition]** [**?**]. In order to intelligently access these white spaces, the SU needs to solve for a channel sensing and subsequent access policy based on the noisy observations of the occupancy behavior of the PUs in the network. However, critical design limitations prevent the SU from sensing all the channels in the discretized spectrum of interest. These sensing limitations are primarily driven by energy efficiency requirements, with some additional restrictions imposed by the need for minimal sensing times [**?**]. So, in view of these sensing limitations, the logical next

step would be to develop algorithms that try to maximize the accuracy of incumbent occupancy behavior estimation, subject to upper bounds on the number of channels that can be sensed by the SU at the beginning of each time-slot: several works in the state-of-the-art [**?**], [**?**], [**?**], [**?**] propose algorithms to solve this limited information spectrum sensing and access problem, however, almost all of these works [**?**], [**?**], [**?**], [**?**], [**?**], [**?**] fail to leverage the correlations exhibited in the occupancy behavior of the incumbents across both frequency and time [**?**], **[NM: too many which... which.. please break it down]** which as we will illustrate later in this work, lead to significant improvements in the estimation accuracy, which in turn leads to a greater number of white spaces accessed by the SU for delivering its network flows, thereby resulting in a higher SU network throughput with lower levels of interference caused to the PUs in the network.**[NM: you tend to use very long phrases...these may be difficult to follow, try to break them down]** In the sections that follow, **[NM: be more specific, "To overcome this challenges, in this paper we propose..."]** we detail solutions to learn these frequency-time correlations in PU occupancy behavior, and to concurrently utilize these learned statistics to solve for an optimal sensing and access policy using approximate POMDP value iteration methods. We also extend our single-agent system model to distributed and centralized multi-agent deployment settings, with neighbor discovery and channel access rank allocation schemes over the control channel, to not only study the implementation feasibility of our POMDP framework, but to also illustrate the performance disparities between collaborative and opportunistic access. **[NM: summarize the numerical results here..]**

Related Work: ~~As mentioned earlier, algorithms in the state-of-the-art do tackle the spectrum sensing and access problem, albeit with some underlying assumptions: many of these assumptions when broken down or generalized will lead to a better solution, as discussed in this work. Firstly,~~ The work [**?**] details a solution for spectrum sensing and access employing TD-SARSA with linear value function approximation. However, this work fails to capitalize on the correlated occupancy behavior of the PUs across frequencies. Additionally, the authors fail to provide a mechanism to manage the trade-off between secondary network throughput and incumbent interference, which we do. Frequency correlation is studied in [**?**], but the assumed observation model is noise-free, which is not realistic. ~~Unlike citeWCL:5, although citeWCL:7 considers frequency correlation, the assumed observation model is noise-free, which is not realistic.~~ In our work, we employ ~~On the other hand, we in this work, present~~ a Hidden Markov Model (HMM) system-level framework in which the true occupancy states of the incumbents in the channels

are hidden behind noisy observations at the SU's spectrum sensor. ~~In addition to the noise-free observation model assumptions in $citeWCL:7$, our solution outperforms both the Minimum Entropy Merging algorithms detailed in it, i.e., Markov Process Estimation coupled with Greedy Clustering and Markov Process Estimation coupled with Minimum Entropy Increment Clustering.~~ **[NM: dont talk about the numerical results here. In this section, you are just outlining the most relevant work and the key differences with your work]** Under the HMM framework, the work [**?**] develops a Viterbi algorithm for occupancy behavior estimation. However, this scheme relies on prior knowledge of the transition model, and fails to dictate *which* spectrum bands should be sensed under spectrum sensing limitations. In contrast, in our proposed solution, SUs learn the transition model with the Baum-Welch algorithm, and leverages this knowledge to concurrently optimize spectrum sensing and access under sensing limitations. ~~Additionally, among works that tackle this problem as an HMM framework $citeWCL:6$ like we do, the Viterbi algorithm is featured as a potential solution for occupancy behavior estimation. As illustrated in the subsequent sections of this work, not only does our solution outperform the Viterbi algorithm (with the same channel sensing limitations), our solution also provides for an online transition model estimation algorithm that operates concurrently with the approximate POMDP value iteration algorithm.~~ **[NM: you need to be more precise as to WHY our solution is better than Viterbi. I.e what are the key differences?]** In contrast, the proposals outlined in [**?**] and [**?**] determine the time-frequency incumbent occupancy correlation structure offline using pre-loaded databases, which is inefficient in non-stationary settings.

Furthermore, [**?**], [**?**], [**?**], [**?**], [**?**] develop spectrum sensing and access algorithms under the assumption that the occupancy behavior of incumbents is independent across both time and/or frequencies, which is not only impractical but also imprudent because critical information aiding the accurate detection of white spaces can be gleaned by exploiting the correlations in their time-frequency occupancy behavior. Prudently, in this paper, we exploit both frequency and temporal correlations. ~~In $citeWCL:9$, a compressed spectrum sensing scheme is devised that exploits sparse temporal dynamics in PU occupancies, and in $citeWCL:10$, an efficient spectrum sensing strategy is proposed for dense multi-cell cognitive networks, that also exploits the spatial structure of interference; however, both works assume independence across frequencies.~~ Not dissimilar to the system model adhered to in our work, the adaptive DQN framework in [**?**] considers both frequency and time correlation in incumbent occupancies, governed by an unknown Markov model, with channel sensing restrictions: however, evaluating cognitive radio throughput with

respect to incumbent interference, our framework provides superior performance, as illustrated in Sec. IV.**[NM: again, what are the key differences? i believe they do not allow to tune the trade off? What else?]**

Additionally, analyzing the state-of-the-art in the distributed cognitive radio networks domain, we find both collaborative as well as opportunistic schemes for channel access: namely, [**?**] describes a TD-SARSA framework with Linear Function Approximation applied to distributed multi-agent settings with the MDP transition model learned via stochastic approximation, and [**?**] details a collaborative scheme (greedy learning under pre-allocation) as well as an opportunistic scheme (g-statistics with ACKs) in distributed multi-agent deployment settings. As discussed earlier,**[NM: no need to repeat]** [**?**] fails to leverage correlations in incumbent occupancy behavior across frequencies, and bypassing the estimation of state-transition probabilities during the heuristic belief update rule which is instead based directly on the observations,**[NM: do you mean that they have prior knowledge?]** results in avoidable errors while devising the optimal access policy. The frameworks in [**?**] are based on assumptions of independence in incumbent occupancy behavior across both time and frequency, which as mentioned earlier, is impractical and imprudent.**[NM: ..then you should have cited this paper in the other paragraph, and don't repeat again]** In this work, we also evaluate the implementation feasibility of our distributed POMDP optimal policy by testing it on an ad-hoc wireless platform of ESP32 radios [**?**], [**?**]. **[NM: What are the key differences of your work wrt [?] and [?] (except for the correlation thing, which should only be discussed in the previous paragraph)?]**

[NM: are there other works doing implementation as you do? IF not, than you should explicitly mention that to the best of our knowlefge, spectrum access and sensing have not been implemented in real-world settings. in contrast, we implement our solution on the DARPA and EPS32 network.]

[NM: this is part of the contributions, not of related work] Finally, in order to evaluate the performance of our POMDP policy in centralized multi-agent settings, we retrofit it into our BAM! Wireless radio [**?**], designed specifically for the DARPA Spectrum Collaboration Challenge (SC2) [**?**], [**?**], emulate its operations during the Active Incumbent scenario (TDWR-UNII WLAN) [**?**], and prove superior performance over heuristics that perform channel and bandwidth allocation via weighted PSD + CIL heuristics [**?**], [**?**], [**?**], [**?**], [**?**].

Contributions: In a nutshell, the contributions of this paper are as follows: **[NM: the list is too long.. should have no more than 4-5 points]**

- We first develop a POMDP framework for spectrum sensing and access in a radio environment with a single cognitive radio node and multiple licensed users exhibiting Markovian correlations in their occupancy behavior across both time and frequency, assuming an AWGN observation model with sensing limitations, and a Raleigh channel fading model;
- We develop an online parameter estimation algorithm to learn the incumbents' occupancy correlation model via an HMM EM formulation, i.e., the Baum-Welch algorithm;
- Concurrently, we leverage these learned statistics in a randomized point-based value iteration algorithm known as PERSEUS, to devise the optimal spectrum sensing and access policy;
- Additionally, we alleviate the computational complexity associated with PERSEUS by introducing fragmentation heuristics and belief update simplification tactics (via Hamming distance state filters); **[NM: this should be combined with the previous item]**
- We compare numerically, in single-agent settings, the proposed framework with state-of-the-art algorithms, and demonstrate superior performance; **[NM: not really a contribution, remove]**
- Next, we extend this single-agent formulation to distributed multi-agent deployment settings, with neighbor discovery (RSSI-based thresholding) and channel access rank allocation (quorum-based preferential ballot voting), and demonstrate enhanced performance over both collaborative and opportunistic distributed multi-agent state-of-the-art;
- Furthermore, retrofitting this POMDP framework into the MAC layer of our DARPA SC2 radio (BAM! Wireless [**?**]), we evaluate its performance against centralized state-of-the-art channel allocation schemes, during the emulation of a real-world TDWR-UNII WLAN scenario, and prove better performance; and
- Finally, we evaluate the implementation feasibility of our POMDP optimal channel sensing and access policy, by testing it on an ad-hoc distributed wireless platform of ESP32 radios. **[NM: combine these two together since they are both part of "implementation"]**

The rest of this paper is organized as follows: Sec. II details the system model; Sec. III describes the individual algorithms that constitute our solution; Sec. IV presents numerical evaluations for the single-agent case; Sec. V elucidates an extension of our solution to a distributed multi-agent setup; Sec. VI illustrates the design of our radio for DARPA SC2, in addition to evaluating its performance in centralized multi-agent settings, i.e., the Active Incumbent scenario emulation; Sec. VII outlines the feasibility analysis of our solution by studying its implementation

on an ad-hoc platform of ESP32 radios; and finally in Sec. VIII we state our conclusions.

II. System Model

A. Signal Model

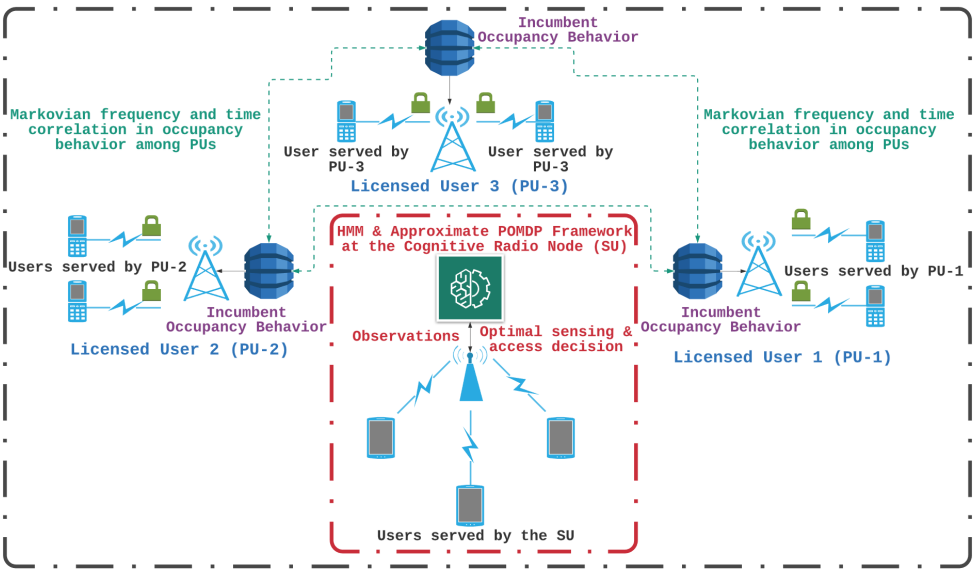

Fig. 1. The radio ecosystem under analysis: An exemplification of the system model detailed in Sec. II-A with $J{=}3$ and $\tilde{J}{=}1$

We consider a primary network of J incumbents/licensed users referred to as Primary Users (PUs) and a secondary network of $\tilde{J}$ cognitive radio nodes referred to as Secondary Users (SUs), trying to exploit portions of the spectrum left unused by these PUs, as illustrated in Fig. 1.~~each equipped with a spectrum-sharing component that is tasked with the objective of maximizing its throughput while limiting interference to the PUs. The spectrum of interest is discretized into K channels of equal bandwidth W, and $\tilde{J}$ SUs trying to exploit portions of the spectrum left unused by these PUs, across time-slots and across frequencies. As illustrated in Fig. $ref fig: A.0$, we first discuss a scenario with a single cognitive radio node $\tilde{J}{=}1$ in a network of $J{=}3$ incumbents, after which we extend our discussion to multiple SUs.~~ In the following, we focus on the single-agent case ($\tilde{J} = 1$); we will discuss the multi-agent scenario in Sec. XXX. The spectrum of interest is discretized into K channels of equal bandwidth W. The discretized wide-band signal received at the SU's spectrum sensor in time-slot i at carrier frequency k can be expressed in the frequency domain as

$$Y_k(i) = \sum_{j=1}^{J} H_{j,k}(i) X_{j,k}(i) + V_k(i), \tag{1}$$

where $X_{j,k}(i)$ represents the frequency domain signal of PU $j \in \{1, 2, \ldots, J\}$ in channel $k \in \{1, 2, \ldots, K\}$, with $X_{j,k}(i)=0$, if PU j is not transmitting over channel k in time-slot i; $H_{j,k}(i)$ denotes the frequency domain channel between the SU and PU j; and $V_k(i) \sim \mathcal{CN}(0, \sigma_V^2)$ constitutes the zero-mean circularly symmetric additive complex Gaussian noise with variance σ_V^2, i.i.d across frequency and time, and independent of the channel H and the PU signal X. Assuming an Orthogonal Frequency Division Multiple Access (OFDMA) strategy among the PUs with respect to the channels in this discretized spectrum, and letting $X_k(i) \triangleq X_{j_{k,i},k}(i)$ and $H_k(i) \triangleq H_{j_{k,i},k}(i)$, where subscript $j_{k,i}$ denotes the index of the PU that occupies channel k in time-slot i, we can rewrite (1) as

$$Y_k(i) = H_k(i)X_k(i) + V_k(i), \tag{2}$$

where $X_k(i)=0$, if channel k is not occupied by any PU in time-slot i. We model the frequency domain channel $H_k(i)$ as Rayleigh fading, so that $H_{j,k}(i) \sim \mathcal{CN}(0, \sigma_H^2)$, with variance σ_H^2, i.i.d across frequency and time.

B. Occupancy Correlation Structure

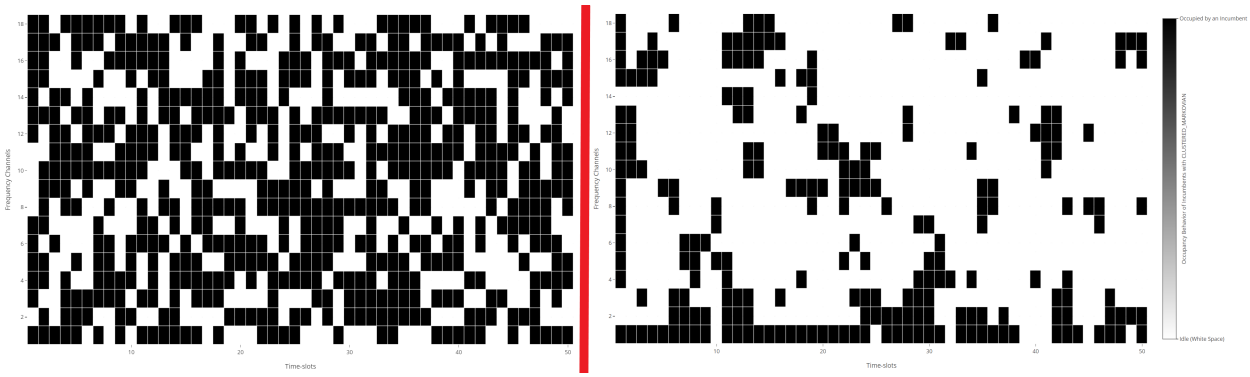

Fig. 2. The incumbent spectrum occupancy heat-map assuming independence across both frequency and time (L), and assuming a time-frequency Markovian correlation structure (R) **[NM: I would remove this figure]**

The frequency domain signal of the PU occupying channel k in time-slot i is modeled as

$$X_k(i) = \sqrt{P_T} B_k(i) S_k(i), \tag{3}$$

where P_T denotes the transmission power of the occupant PU; $B_k(i)$ represents the binary channel occupancy variable, with $B_k(i)=1$ if channel k is occupied by a PU in time-slot i, and $B_k(i)=0$ otherwise; $S_k(i)$ is the transmitted symbol, i.i.d across frequency and time, modeled

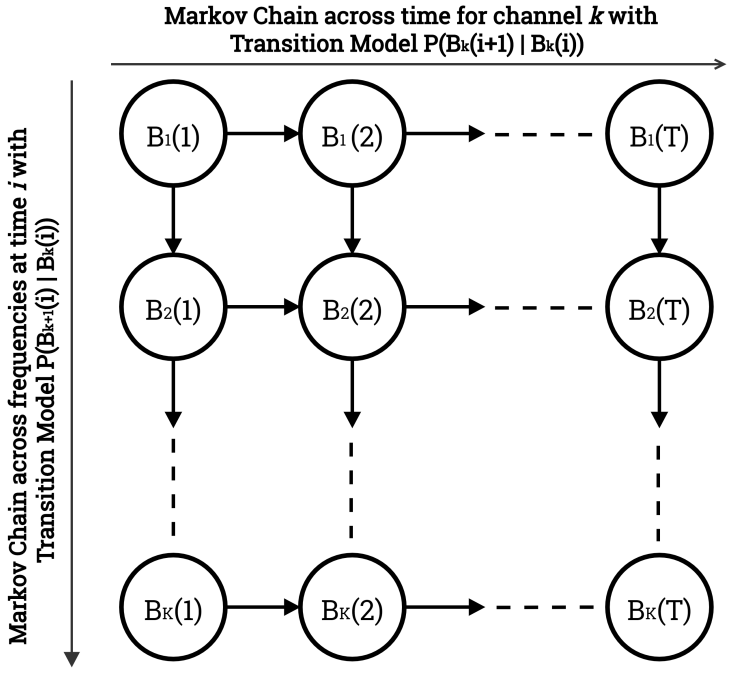

Fig. 3. The visualization of the incumbent occupancy time-frequency correlation structure as two dependent Markov chains: one across time and the other across frequencies

from a certain constellation. Then, $H_k(i)X_k(i)=\sqrt{P_T}B_k(i)H_k(i)S_k(i)$. Herein, we approximate $H_k(i)S_k(i)$ as a zero-mean complex Gaussian random variable with variance $\sigma_H^2\mathbb{E}[|S_k|^2]$. We denote the spectrum occupancy state in time-slot i as

$$\vec{B}(i) = [B_1(i), B_2(i), B_3(i), \ldots, B_K(i)]^\intercal \in \{0,1\}^K. \tag{4}$$

~~where $\vec{B}(i) \in \{0,1\}^K$.~~ **[NM: next phrase is toooo looong]** We assume that spectrum occupancy is correlated in frequency and time because a PU usually occupies adjacent channels for prolonged periods of time, i.e., the incumbents usually restrict their transmissions to certain parts of the spectrum, occupying a set of adjacent bands, and exhibit temporal patterns in their occupancy of these bands, with the temporal patterns governed by the operational periodicity of military users or by the prolonged usage by licensed service providers [**?**], with this correlation in occupancy behavior depicted via connected databases**[NM: unclear]** in Fig. 1. To capture temporal correlation, we model the evolution of $\vec{B}(i)$ over time as a Markov process ~~We decompose this time-frequency correlation structure as follows: we model the temporal correlation in incumbent~~

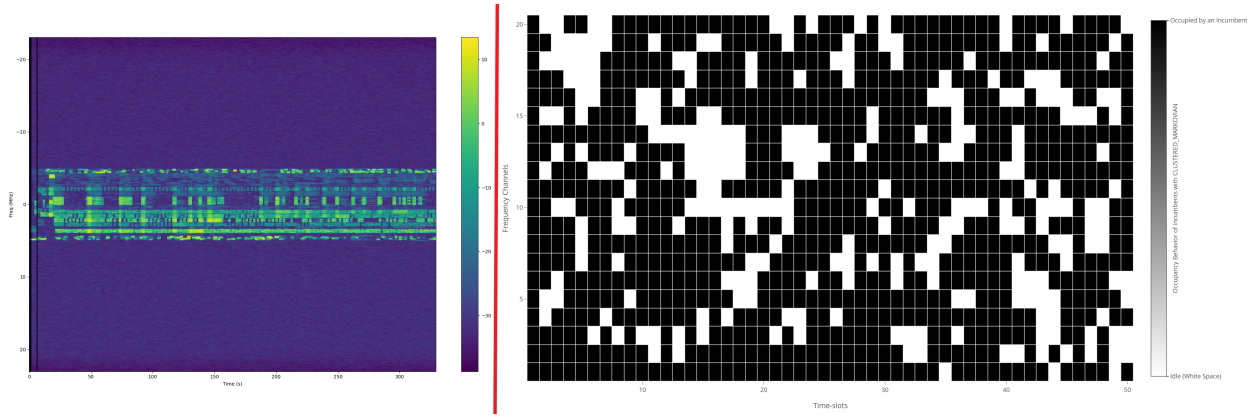

Fig. 4. The combined PSD plot of the occupancy behavior of incumbents and competitors during the DARPA SC2 Active Incumbent scenario emulation (L), and the spectrum occupancy heat-map visualized by fitting an estimation of the proposed time-frequency Markovian correlation map to this data (R)**[NM: this is confusing.. is the R generated independently from the L? OR is the R the estimation of the spectrum occupancy? The R figure should be removed..] [NM: I would remove this figure and have instead only the heatmap of Figure 5]**

Fig. 5. A visualization of the occupancy heat-map of a DARPA SC2 Passive Incumbent in order to illustrate scenarios with well-defined patterns in the occupancy behavior of the incumbent(s) **[NM: Is this the actual heatmap of darpa?] [NM: the text in the figure cannot be read too small font]**

~~occupancy behavior as a Markov process described by~~

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(j), \forall j \leq i) = \mathbb{P}(\vec{B}(i+1)|\vec{B}(i)). \tag{5}$$

In addition, to model frequency correlation, we further decompose $\mathbb{P}(\vec{B}(i+1)|\vec{B}(i))$ as ~~and~~

~~we couple this model with another Markov chain across the frequency bands to capture the frequency correlation in incumbent occupancy behavior, to get the final correlation structure as~~

$$\mathbb{P}(\vec{B}(i+1)|\vec{B}(i)) = \mathbb{P}(B_1(i+1)|B_1(i)) \prod_{k=2}^{K} \mathbb{P}(B_k(i+1)|B_{k-1}(i+1), B_k(i)). \quad (6)$$

In other words, we can describe this Markovian time-frequency correlation as follows: the occupancy of frequency band k in time-slot $i+1$ depends on the occupancy of the adjacent frequency band $k-1$ in the same time-slot $i+1$, and the occupancy of the same frequency band k in the previous time-slot i. **[NM: Please refer to Fig 3]** We parameterize this two-chain Markovian correlation structure by the parameter vector $\vec{\theta} = [\vec{p}\ \vec{q}]^\intercal$, where

$$\begin{aligned} &\vec{\theta} = [\vec{p}\ \vec{q}]^\intercal, \text{ where} \\ &\vec{p} = [p_{uv} = \mathbb{P}(B_k(i+1) = 1 | B_{k-1}(i+1) = u, B_k(i) = v) : u, v \in \{0,1\}]^\intercal, \text{ and} \\ &\vec{q} = [q_w = \mathbb{P}(B_1(i+1) = 1 | B_1(i) = w) : w \in \{0,1\}]^\intercal. \end{aligned} \quad (7)$$

This vector $\vec{\theta}$, parameterizes the transition model of our POMDP formulation described in Sec. II-B, and is estimated by an HMM-specific Expectation Maximization (EM) algorithm, i.e., the Baum-Welch algorithm, as detailed in Sec. III-B.

[NM: Figures should appear in the same order as they are mentioned in the manuscript, and on the SAME page or AFTER when they are first mentioned (never before)] Fig. 2 (L) illustrates the spectrum occupancy heat-map, assuming independence in occupancy behavior across both frequency and time, while Fig. 2 (R) depicts the spectrum occupancy heat-map, assuming a time-frequency Markovian correlation structure in incumbent occupancy behavior, with $\vec{p}$=$[p_{00}$=$0.1, p_{01}$=$0.3, p_{10}$=$0.3, p_{11}$=$0.7]^\intercal$ and $\vec{q} = [q_0$=$0.3, q_1$=$0.8]^\intercal$. **[NM: I would remove this figure and accompanying discussion,.. it is quite trivial]** The time-frequency correlation structure underlying the occupancy behavior of PUs in the network, as described by (6), can be illustrated as two dependent Markov chains: one across time and the other across frequencies, as shown in Fig. 3. If the frequency correlation direction is changed, i.e., as opposed to the depiction in Fig. 3, if occupancy of channel $k+1$ influences the occupancy of channel k, k∈$\{1, 2, ...K-1\}$ (bottom-up correlation), our model and subsequent analyses still hold.

In order to experimentally verify that our time-frequency Markovian correlation structure ($\vec{\theta}$) is indeed the correct**[NM: there is no such thing as "correct" or "wrong".. there is "good" or "better", "worse" or "poor"]** model to analyze incumbent occupancy behavior in the real-world, as opposed to time-frequency independence models (Q) [?], [?], [?], [?], [?], only

temporal correlation models ($\mathbf{R}$) [**?**], and only frequency correlation models ($\mathbf{S}$), in the state-of-the-art, we evaluate the Kullback-Liebler (KL) divergence of an estimation of our model ($\hat{\vec{\theta}}$) against the true occupancy behavioral model ($\mathbf{P}$)**[NM: how do you know the true model given that you only have data??]** of the incumbent and the competitors in the DARPA SC2 Active Incumbent scenario emulated in the Colosseum **[NM: are you doing this measurements based on the heatmap of Fig 5? Please specify]** [**?**], [**?**], [**?**], [**?**] ($D_{KL}(\mathbf{P}||\hat{\vec{\theta}})$), which mimics the operation of a practical cognitive radio network wherein WiFi DFS radios (50 SUs) intelligently access the spectrum white spaces left unused by a Terminal Doppler Weather Radar (TDWR) system (1 Active PU). Fig. 4 illustrates the combined PSD plot of the measurements made at the individual cognitive radios of our BAM! Wireless network [**?**], aggregated at the gateway node, and a visualization of the spectrum occupancy heat-map obtained by fitting an estimation of our model to this dataset. **[NM: this section does not make much sense to me..]** By fitting our time-frequency Markovian correlation structure to this emulation **[NM: using what algorithm described in which section?]**, we find that our model represents the true occupancy behavior of the incumbent and the competitors in this scenario with the least information loss, among other prominent models in the state-of-the-art. Specifically, our KL-divergence analyses yields the following results:

$$\begin{aligned} D_{KL}(\mathbf{P}||\hat{\vec{\theta}}) &= 0.05997, \\ D_{KL}(\mathbf{P}||\mathbf{Q}) &= 0.23071, \\ D_{KL}(\mathbf{P}||\mathbf{R}) &= 0.14349, \\ D_{KL}(\mathbf{P}||\mathbf{S}) &= 0.25665; \end{aligned} \tag{8}$$

where, for our time-frequency Markovian correlation model estimate ($\hat{\vec{\theta}}$), the KL-divergence metric is obtained by

$$D_{KL}(\mathbf{P}||\hat{\vec{\theta}}) = \mathbb{E}_{k,i}\left[\sum_{b\in\{0,1\}} \mathbb{P}(B_k(i){=}b)\ln\left(\frac{\mathbb{P}(B_k(i){=}b)}{\mathbb{P}(B_k(i){=}b|\Gamma,\hat{\vec{\theta}})}\right)\right], \tag{9}$$

[NM: I dont understand what you are doing here...1) you fit the parameters, 2) you compute the expectation?? How do you compute $\mathbb{P}(B_k(i){=}b)$ from data? Why is there an expectation? (I believe what you have inside the E is already and expectation....)] where Γ represents the appropriate**[NM: what do you mean]** history of occupancies for this model, i.e., the occupancies in the adjacent channels and time-slots, and $\hat{\vec{\theta}} = [p_{00} = 0.25, p_{01} = 0.75, p_{10} = 0.71, p_{11} = 0.8, q_0 = 0.67, q_1 = 0.75]^{\intercal}$**[NM: Is this your estimateed parameter?]** with the

steady-state occupancy estimate being $\hat{\Pi}$=0.7. On another note,**[NM: why another note? Im confused..]** the Passive Incumbent scenario [**?**] emulated in the DARPA SC2 Colosseum mimics incumbents with a more predictive occupancy behavior, as illustrated in Fig. 5.

[NM: Remove:]Based on the estimates of spectrum occupancy obtained by our spectrum sensing policy, discussed in Sec. III-C, the SU accesses only those channels deemed idle by this estimation procedure. The reward metric associated with the cognitive radio node captures both the number of truly idle channels (correctly estimated idle) accessed by it, accounting for the throughput maximization aspect of our objective, as well as the number of truly occupied channels (incorrectly estimated idle) accessed by it, accounting for the incumbent interference minimization aspect of our objective. **[NM: why is this paragraph here? IT is related to the reward model it has nothing to do with the transition model]**

C. Channel Sensing Model

Equipped with a spectrum sensor, the SU detects white spaces and accesses them to deliver its network flows. Owing to physical design limitations, specifically, the restriction on the number of channels that can be sensed by the SU's spectrum sensor in any given time-slot, primarily due to concerns about energy-efficiency and sensing/data aggregation times [**?**], the SU can sense a maximum of κ spectrum bands in a time-slot, with $1 \leq \kappa \leq K$. **[NM: compact version:]**However, due to concerns about energy-efficiency and sensing/data aggregation times [**?**], the SU can sense a maximum of κ spectrum bands in a time-slot, with $1 \leq \kappa \leq K$. Let $\mathcal{K}_i$ be the set of channels sensed by the SU at time i, i.e., $\mathcal{K}_i \subseteq \{1, 2, \ldots, K\}$, with ~~the sensing limitation imposed as~~ $|\mathcal{K}_i| \leq \kappa$. **[NM: It is fine to connect the sections together, but here you are still discussing the model, so be concise and orgnanized]** ~~The solution to the spectrum sensing problem is to determine an optimal set of channels sensed by the SU in any time-slot i, and this optimal set is dictated by the optimal policy derived from the POMDP formulation via the PERSEUS algorithm detailed in Sec. $refIII.III$. The solution to the access problem hinges on the optimal sensing policy, i.e., based on the "best-possible" spectrum occupancy picture painted by the optimal sensing action in a time-slot i, the SU accesses all the channels it deems to be idle, more details on this access strategy are discussed in Sec. $refIII.0$.~~ After sensing the channels listed in $\mathcal{K}_i$, governed by the sensing policy, the obtained observation vector is $\vec{Y}(i) = [Y_k(i)]_{k \in \mathcal{K}_i}]$, where $Y_k(i)$ is described in (2) .

In statistics, Hidden Markov Models (HMMs) are used to describe systems modeled by Markov processes, with the actual system states "hidden" behind the observed noisy measurements of these states. Along these lines, constructing an HMM for our problem, the linear, additive, Gaussian noise in the observation model described in Sec. II-A, introduces uncertainty into the sensing process, the true occupancy states of the frequency bands in time-slot i, i.e., $\vec{B}(i)$, represent the actual states of the model, while the observations at the SU's spectrum sensor, i.e., $\vec{Y}(i)$, represent the noisy observations of these true occupancy states. The observation vector in time-slot i, $\vec{Y}(i)$, given the occupancy vector in that time-slot, $\vec{B}(i)$, has a Probability Density Function (PDF) described by

$$f(\vec{Y}(i)|\vec{B}(i),\mathcal{K}_i) = \prod_{k=1}^{K} f(Y_k(i)|B_k(i)), \tag{10}$$

due to the i.i.d assumptions of the noise $V_k(i)$, the transmitted symbols $S_k(i)$, and the Rayleigh fading variables $H_k(i)$, across channels, given the occupancy state vector, as discussed in Sec. II-A. Additionally, we can infer from (2) that

$$Y_k(i)|B_k(i) \sim \mathcal{CN}(0, \sigma_H^2 P_T B_k(i) + \sigma_V^2). \tag{11}$$

D. POMPD formulation

III. Proposed Solution: The Algorithms

[NM: POMDP formulation should still be part of the model section...]

A. POMDP Formulation

Partially Observable Markov Decision Processes (POMDPs) are employed in modeling the repeated, sequential interactions of an agent, tasked with maximizing its reward, subject to the problem at hand, with a stochastic environment, wherein the limited observational capacity of the agent and/or the observation noise, creates partial observability vis-à-vis the underlying states of the environment. Our POMDP formulation, represented by the 5-tuple $(\mathcal{B}, \mathcal{A}, \mathcal{Y}, \mathbf{A}, \mathbf{M})$, features the state space of the underlying MDP, denoted by $\mathcal{B} \equiv \{0,1\}^K$, which is given by all possible realizations of the occupancy vector $\vec{B}$; the action space of the SU**[NM: this is just the action space of the sensing, what about the action space of the access?]**, denoted by $\mathcal{A}$, which is described by all possible combinations in which $1 \leq \kappa \leq K$ channels are chosen to be sensed in a time-slot (discussed in Sec. II-C); the observation space, denoted by $\mathcal{Y}$, which is discussed

in Sec. II-A; the transition model of the underlying MDP, denoted by $\mathbf{A}$, which is discussed in Sec. II-B; and the observation model (also known as the emission model), denoted by $\mathbf{Y}$, which is described by (10) and (11).

Prior to gathering the occupancy information in time-slot i, based on the measurements obtained by the SU's spectrum sensor up to, but not including, time-slot i, the POMDP state is described by the prior belief, denoted by β_i, representing the probability distribution of the underlying MDP state~~, i.e.,~~ $\vec{B}(i)$ given the history. Given this prior belief β_i, the SU chooses a sensing action according to a sensing policy π, i.e., $\pi(\beta_i) = \mathcal{K}_i \in \mathcal{A}$, wherein as detailed in Sec. II-C, the SU senses the frequency bands corresponding to the channel indices in the set $\mathcal{K}_i$, observes $[Y_k(i)]_{k\in\mathcal{K}_i} \in \mathcal{Y}$, and updates its belief of the underlying MDP state $\vec{B}(i)$ to obtain its posterior belief, which is written as

$$\begin{aligned}\hat{\beta}_i(\vec{B}') &= \mathbb{P}(\vec{B}(i) = \vec{B}' | \beta_i, \mathcal{K}_i, [Y_k(i)]_{k\in\mathcal{K}_i}) \\ &= \frac{\mathbb{P}([Y_k(i)]_{k\in\mathcal{K}_i} | \vec{B}', \mathcal{K}_i)\beta(\vec{B}')}{\sum_{\vec{B}''\in\{0,1\}^K} \mathbb{P}([Y_k(i)]_{k\in\mathcal{K}_i} | \vec{B}'', \mathcal{K}_i)\beta_i(\vec{B}'')}.\end{aligned} \tag{12}$$

Given the posterior belief $\hat{\beta}_i$ the SU then performs channel access decisions After channel sensing is performed by the SU's spectrum sensor, according to the sensing policy, using the posterior belief described in (12), channel access decisions have to be made: the underlying MDP state $\vec{B}(i)$ is estimated as follows. After computing the posterior belief $\hat{\beta}_i$, the reward under the access decision $\vec{\phi}(i)$ is

$$R(\vec{\phi}(i), \hat{\beta}_i) = \mathbb{E}\left[\sum_{k=1}^{K}(1 - B_k(i))\phi_k(i) - \lambda B_k(i)\phi_k(i)\right] = \mathbb{E}\left[\sum_{k=1}^{K}(1 - \hat{\beta}_{i,k})\phi_k(i) - \lambda\hat{\beta}_{i,k}\phi_k(i)\right] \sum_{k=1}^{K}(1 - \hat{\beta}_{i,k})\phi_k(\tag{13}$$

[NM: describe first this reward before going into its optimization and your overall optimization problem: you are trying to optimize the cumulative reward with respect to sensing and access actions... also, the expectation is conditional on the posterior belief! Show the conditioning... also, the quantity inside the E in the right hand side is already an expectation] and the optimal access decision is $\vec{\phi}^*(i) = \arg\max R(\vec{\phi}(i), \hat{\beta}_i)$, **[NM: this can be done in closed form, please show the closed form expression]** following which, if $\phi_k(i)$=1, which implies that the SU estimated channel k to be occupied by a PU in this time-slot i, and hence leaves it untouched, while if $\phi_k(i)$=0, the SU accesses this "estimated idle" channel k in time-slot i to deliver its network flows.

Ensuing the determination of the reward for its access decision from the radio environment, the SU computes the prior belief for the next time-slot $i+1$ as

$$\beta_{i+1}(\vec{B}'') = \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}''|\vec{B}(i) = \vec{B}')\hat{\beta}(\vec{B}'). \tag{14}$$

Let

$$\hat{\beta}_i = \hat{\mathbb{B}}(\beta_i, \mathcal{K}_i, \vec{Y}(i)) \tag{15}$$

denote the function that maps the prior belief β_i to the posterior belief $\hat{\beta}_i$ in time-slot i, and let

$$\beta_{i+1} = \mathbb{B}(\hat{\beta}_i) \tag{16}$$

denote the function that maps the posterior belief $\hat{\beta}_i$ in time-slot i to the prior belief β_{i+1} in time-slot $i+1$. The objective of the SU is to determine the optimal spectrum sensing policy (based on which the access decisions are made in the corresponding time-slots) to maximize its infinite-horizon discounted reward, i.e.,

$$\pi^* = \arg\max_{\pi} V^{\pi}(\beta), \tag{17}$$

where

$$V^{\pi}(\beta) = \mathbb{E}_{\pi}\left[\sum_{i=1}^{\infty} \gamma^i R(\vec{B}(i), \hat{\beta}_i)\Big|\beta_0 = \beta\right], \tag{18}$$

where $0<\gamma<1$ is the discount factor, $\beta_0=\beta$ is the initial belief such that the value function $V^{\pi}(\beta)$ is evaluated from this starting belief, and $\hat{\beta}_i$ is the posterior belief induced by the policy $\mathcal{K}_i=\pi(\beta_i)$ and the observation vector $[Y_k(i)]_{k\in\mathcal{K}_i}$ via the formulation $\hat{\beta}_i=\hat{\mathbb{B}}(\beta_i, \mathcal{K}_i=\pi(\beta_i), [Y_k(i)]_{k\in\mathcal{K}_i})$. The Bellman operator, denoted by $\mathcal{H}$, employed in the Bellman optimality equation, $V^*=\mathcal{H}(V^*)$, is defined at iteration $t+1$ as, $\forall\beta$

$$\begin{aligned} V_{t+1} &= \mathcal{H}(V_t) \\ &= \max_{\mathcal{K}\in\mathcal{A}} \sum_{\vec{B}\in\mathcal{B}} \beta(\vec{B}) \mathbb{E}_{[Y_k]_{k\in\mathcal{K}}|\vec{B},\mathcal{K}} \left[R(\vec{B}, \hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k\in\mathcal{K}})) + \gamma V_t(\mathbb{B}(\hat{\mathbb{B}}(\beta, \mathcal{K}, [Y_k]_{k\in\mathcal{K}}))\right]. \end{aligned} \tag{19}$$

By employing value iteration algorithms, (19) can be solved iteratively until convergence to a fixed point that corresponds to the optimal sensing policy. However, this direct approach results in complications associated with the lack of prior knowledge about the incumbent occupancy time-frequency correlation structure that defines the transition model of the underlying MDP, and the computational infeasibility of the approach, as the number of channels in the discretized spectrum of interest increases, the number of states of the underlying MDP scales exponentially,

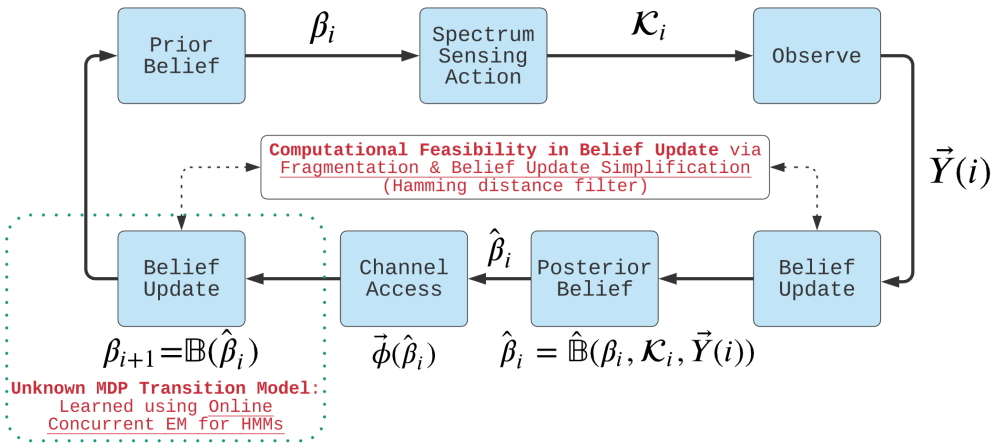

Fig. 6. The POMDP process flow as discussed in Sec. III-A

resulting in a high-dimensional belief space, which makes the approach intractable. An illustration of the POMDP formulation is provided in Fig. 6.

We solve the problem of intractability of the POMDP for large state and action spaces by employing a randomized, point-based, approximate value iteration algorithm known as PERSEUS [**?**] to solve for the optimal sensing policy, while an online parameter estimator embedded into PERSEUS allows us to solve the problem of transition model ignorance. More specifically, there are two challenges that arise when we try to solve (19) optimally,

- The transition model of the MDP underlying the POMDP formulation, defined by the parameter vector $\vec{\theta}$, is unknown. This makes it impossible to perform the belief update procedure detailed in (13). As discussed in Sec. III-B we solve this problem by incorporating an HMM EM estimator, i.e., the Baum-Welch algorithm, to learn the time-frequency occupancy correlation structure while concurrently solving for the optimal sensing and access policy.
- Solving (19) exactly is computationally infeasible because the number of states in the underlying MDP scales exponentially with the number of frequency bands in the spectrum of interest. As discussed in Sec. III-C we solve this problem by incorporating a low-complexity approximate value iteration algorithm known as PERSEUS, with fragmentation (into independent subsets of highly-correlated channels) and belief update simplification heuristics (Hamming distance state filters).

B. Occupancy Correlation Structure Estimation

Practical implementations of the MAC layer of the cognitive radio's network protocol stack involve solving for the optimal sensing and access policy, without having any prior information about the time-frequency correlation structure underlying the occupancy behavior of the incumbents in the network. This correlation structure, as discussed earlier, can be leveraged to improve the occupancy state estimation accuracy, which in turn facilitates higher SU network throughput with lower PU interference. In this regard, in this section, we propose a parameter estimator algorithm that learns this correlation structure over time, with the learned correlation structure iteratively fed into the POMDP optimal policy solver (i.e., PERSEUS), in order to enable a concurrent framework that minimizes the amount of computational resources (time, memory, processing power) required to obtain the optimal policy, which is especially crucial in non-stationary settings.

Let τ refer to the learning period of the parameter estimation algorithm: this can be equal to the entire duration of the SU's interaction with the radio environment while solving for the optimal policy, implying concurrent model learning facilitated by a publisher-subscriber software architecture and multi-threading features: in a time-slot, the diverse, sparse observations made by the SU as a part of the PERSEUS thread's exploration period are concatenated into a complete observation vector over repeated iterations (we assume the dynamics of the PU occupancy over the time-slots are slower than the time needed for these observations and their subsequent concatenation) and injected into the parameter estimation thread, which estimates the transition probabilities in that iteration (which is synchronized with the PERSEUS thread's time-slot dynamics in order to have these two threads operate on the same time-scale) and publishes them, with the PERSEUS thread using the most current published estimates in its operation; or it can be equal to an initial learning period that has been set aside exclusively for the SU to estimate the underlying MDP's transition model, after which the PERSEUS algorithm is initiated, employing these final estimated (converged) transition probabilities. Defining $\mathbf{B}=[\vec{B}(i)]_{i=1}^{\tau}$ as the sequence of states encountered by the SU in time-slots $i=1$ to $i=\tau$, and $\mathbf{Y}=[\vec{Y}(i)]_{i=1}^{\tau}$ as the sequence of observations made at the SU's spectrum sensor from $i=1$ to $i=\tau$, having a one-to-one correspondence with the elements of $[\vec{B}]_{i=1}^{\tau}$, we formulate the Maximum Likelihood Estimation (MLE) problem to estimate the vector $\vec{\theta}$ that parameterizes the PU occupancy time-frequency

correlation structure (detailed in Sec. II-B) as follows:

$$\vec{\theta}^{*} = \arg\max_{\vec{\theta}} \log \left(\sum_{\mathbf{B}} \mathbb{P}(\mathbf{B}, \mathbf{Y}|\vec{\theta}) \right). \tag{20}$$

Solving this MLE formulation using the Expectation-Maximization algorithm [**?**] for HMMs, i.e., the Baum-Welch algorithm, the algorithm boils down to two-steps, the E-step constitutes

$$Q(\vec{\theta}|\vec{\theta}^{(t)}) = \mathbb{E}_{\mathbf{B}|\mathbf{Y},\vec{\theta}^{(t)}} \left[\log \left(\mathbb{P}(\mathbf{B}, \mathbf{Y}|\vec{\theta}^{(t)}) \right] \right], \tag{21}$$

where $Q(\vec{\theta}|\vec{\theta}^{(t)})$ is computed using the Forward-Backward algorithm [**?**]; and the M-step constitutes

$$\vec{\theta}^{(t+1)} = \arg\max_{\vec{\theta}} Q(\vec{\theta}|\vec{\theta}^{(t)}), \tag{22}$$

which involves the re-estimation of $\vec{\theta}$ by employing the statistics $Q(\vec{\theta}|\vec{\theta}^{(t)})$ obtained from the Forward-Backward algorithm.

C. The PERSEUS Algorithm

In our proposed solution, we solve for the optimal spectrum sensing (and access, based on the MAP estimation detailed in Sec. III-A) policy, in parallel with the parameter estimation algorithm, employing its published iterative transition model estimates, until both the EM algorithm and the POMDP policy solver algorithms converge.

As alluded to in Sec. III-A, in order to solve the computational infeasibility precipitated by the exponential increase in the number of states of the underlying MDP, induced by an increase in the number of frequency bands in the discretized spectrum of interest, we employ approximate POMDP value iteration methods to ensure that the formulations and the algorithms scale well to a large number of relevant channels in the radio environment in which the SU operates. Consequently, we choose the PERSEUS algorithm [**?**] to solve for the optimal policy, primarily motivated by the following: the exact value iteration strategies proposed in [**?**], namely the Exhaustive Enumeration algorithm and the Witness algorithm are untenable for large belief spaces, because these techniques involve performing the backup procedure, i.e., determining the optimal action (or hyperplane in a Piece-Wise Linear Convex (PWLC) context) for every belief point in the belief space; and a fellow contemporary approximate value iteration algorithm known as the Point-Based Value Iteration (PBVI) algorithm proposed in [**?**], although involves performing the backup operation over a reduced set of beliefs known as the "reachable beliefs,"

unlike the strategies in [**?**], is computationally expensive due to the task of computing the distances between all the belief points in the set of reachable beliefs in addition to the subsequent backup operation on all these belief points. The PERSEUS algorithm, on the other hand, does not involve performing the backup operation for every point in the belief space, unlike the Exhaustive Enumeration and Witness algorithms detailed in [**?**]; and unlike the PBVI algorithm [**?**] does not involve computing the distances between all the belief points in the set of reachable beliefs, and furthermore, does not involve performing backups on all the reachable belief points, instead, PERSEUS involves "backing-up" only on a subset of this set of reachable beliefs, while ensuring that the computed solution is effective for all the points in the reachable belief set.

PERSEUS, a randomized, point-based, approximate POMDP value iteration algorithm, involves an initial phase of exploration, wherein the set of "reachable-beliefs," denoted by $\tilde{\mathcal{B}}$, is determined by allowing the SU to randomly interact with the radio environment. As referenced earlier, one simplifying (or approximating) feature of PERSEUS is to improve the value of all the belief points in the set $\tilde{\mathcal{B}}$, by computing the value of only a subset of these belief points, which are chosen iteratively at random. For finite-horizon POMDP formulations, the optimal value function V^* described by (19), can be approximated by a Piece-Wise Linear Convex (PWLC) function [**?**], in other words, the value function at iteration t is parameterized by a set of hyperplanes, denoted by $\{\vec{\alpha}_t^u\}, u \in \{1, 2, \ldots, |\tilde{\mathcal{B}}|\}$, wherein each hyperplane represents a region of the belief space for which the action corresponding to this hyperplane, denoted by $\mathcal{K}_t^u$, is the maximizer. Ergo, the value function of belief β in a given iteration t is approximated as

$$V_t(\beta) \approx \beta \cdot \vec{\alpha}_t^{u^*}, \tag{23}$$

where,

$$u^* = \underset{u \in \{1,2,\ldots,|\tilde{\mathcal{B}}|\}}{\arg\max} \; \beta \cdot \vec{\alpha}_t^u, \tag{24}$$

with

$$\beta \cdot \vec{\alpha} = \sum_{\vec{B}} \beta(\vec{B}) \vec{\alpha}(\vec{B}) \tag{25}$$

describing the inner product, and $\mathcal{K}_t^{u^*}$ representing the optimal spectrum sensing action.

We define a set of unimproved belief points, denoted as $\tilde{\mathcal{U}}$, which initially corresponds to the set of reachable beliefs $\tilde{\mathcal{B}}$ obtained by the random exploration procedure detailed earlier. Pick a belief β_u from this set $\tilde{\mathcal{U}}$, and perform the backup operation on this chosen belief point, which as discussed earlier, involves associating a new hyperplane and its corresponding spectrum sensing

action with this belief β_u. In iteration $t+1$, defining $\mathcal{K}^u_{t+1}$ as the action associated with hyperplane $\vec{\alpha_{t+1}}$, corresponding to belief $\beta_u \in \tilde{\mathcal{U}}$, we can describe the backup procedure mathematically as

$$\begin{aligned} \vec{\alpha}_{t+1} &= \xi^u_{\mathcal{K}^u_{t+1}}, \\ \mathcal{K}^u_{t+1} &= \arg\max_{\mathcal{K}\in\mathcal{A}} \beta_u \cdot \xi^u_{\mathcal{K}}, \end{aligned} \tag{26}$$

where $\xi^u_{\mathcal{K}}$ is the hyperplane corresponding to the one-step look-ahead under action $\mathcal{K}\in\mathcal{A}$ and belief β_u, i.e.,

$$\xi^u_{\mathcal{K}}(\vec{B}) = \mathbb{E}_{\vec{Y}|\vec{B},\mathcal{K}} \left[R(\vec{B}, \hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) + \gamma \sum_{\vec{B}'} \mathbb{P}(\vec{B}(i+1) = \vec{B}' | \vec{B}(i) = \vec{B}) \xi^u_{\mathcal{K},\vec{Y}}(\vec{B}') \right], \tag{27}$$

where $\xi^u_{\mathcal{K},\vec{Y}}$ refers to the hyperplane corresponding to the future value function computed from the previous set of hyperplanes from the new belief $\mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y}))$ obtained from β_u by executing action $\mathcal{K}$ and observing $\vec{Y}$, as

$$\xi^u_{\mathcal{K},\vec{Y}} = \arg\max_{\vec{\alpha}^{u'}_t, u' \in \{1,2,\ldots,|\tilde{\mathcal{B}}\}} \mathbb{B}(\hat{\mathbb{B}}(\beta_u, \mathcal{K}, \vec{Y})) \cdot \vec{\alpha}^{u'}_t. \tag{28}$$

After determining the hyperplane α^u_{t+1} associated with this chosen belief point β_u using the backup procedure detailed above, we now know that $V_{t+1}(\beta_u) = \beta_u \cdot \vec{\alpha}^u_{t+1}$ is its approximate value function. The most crucial aspect of PERSEUS that approximates the optimization of a randomly chosen belief point to the entire set $\tilde{\mathcal{U}}$ is as follows: if the approximate value function for belief point $\beta_u \in \tilde{\mathcal{U}}$ is improved by the aforementioned backup iteration, i.e., $V_{t+1}(\beta_u) \geq V_t(\beta_u)$, the belief point β_u is removed from the set $\tilde{\mathcal{U}}$, and now, we check if this hyperplane $\vec{\alpha}^u_{t+1}$ improves the approximate value functions of the other beliefs in $\tilde{\mathcal{U}}$, i.e., $\forall \beta' \in \tilde{\mathcal{U}} - \{\beta_u\}$, if $\beta' \cdot \vec{\alpha}^u_{t+1} \geq V_t(\beta')$, this new hyperplane generates an improved approximate value function, and these respective belief points for which this hyperplane improves their approximate value functions, are removed from the set $\tilde{\mathcal{U}}$. In other words,

$$\begin{aligned} &\tilde{\mathcal{U}} \longleftarrow \tilde{\mathcal{U}} - \{\beta_u\}, \text{ if } \beta_u \cdot \vec{\alpha}^u_{t+1} \geq V_t(\beta_u), \text{ and subsequently} \\ &\tilde{\mathcal{U}} \longleftarrow \tilde{\mathcal{U}} - \{\beta' \in \tilde{\mathcal{U}} : \beta' \cdot \vec{\alpha}^u_{t+1} \geq V_t(\beta')\}. \end{aligned} \tag{29}$$

On the other hand, if this hyperplane $\vec{\alpha}^u_{t+1}$ worsens the approximate value function of β_u, i.e., $\beta_u \cdot \vec{\alpha}^u_{t+1} < V_t(\beta_u)$, the old hyperplane and its associated sensing action persist for β_u, mathematically, $\vec{\alpha}^u_{t+1} := \vec{\alpha}^u_t$ and $\mathcal{K}^u_{t+1} := \mathcal{K}^u_t$; but we still check for improvements with respect to the other belief points in $\tilde{\mathcal{U}}$, and remove all those belief points $\beta' \in \tilde{\mathcal{U}}$ for which $\beta' \cdot \vec{\alpha}^u_{t+1} \geq V_t(\beta')$.

In general, if a hyperplane determined from the backup procedure improves a belief point in the set of unimproved belief points $\tilde{\mathcal{U}}$, this news hyperplane (and its associated sensing action)

becomes the relevant hyperplane (and the relevant sensing action) for this belief point, and the belief point will be removed from the set of unimproved belief points $\tilde{\mathcal{U}}$. These sequence of operations (random choice from $\tilde{\mathcal{U}} \longrightarrow$ backup $\longrightarrow$ check for improvement and removal) are performed iteratively until the set $\tilde{\mathcal{U}}$ is empty: this constitutes a single PERSEUS iteration. These PERSEUS iterations are executed until the specified value iteration termination condition is satisfied, i.e.,

$$|V_{t+1}(\beta) - V_t(\beta)| < \epsilon, \ \forall \beta \in \tilde{\mathcal{B}}, \tag{30}$$

where $\epsilon > 0$ (a very small value), is the value iteration difference threshold.

The PERSEUS algorithm, although is an approximate POMDP method which eliminates the computational overhead associated with the exhaustive belief space and reachable space optimization techniques [**?**], [**?**] by approximating the optimization of a randomly chosen belief point to the entire set of unimproved, reachable belief points, still possesses computational intractability challenges because it involves iterations over all possible combinations of the occupancy state vector, i.e., $\vec{B} \in \{0,1\}^K$, the computational cost scales exponentially with the number of states in the underlying MDP, which is induced by the number of channels K in the discretized spectrum of interest. In order to solve this computational tractability problem, we introduce two simplifying heuristics into the PERSEUS algorithm. Firstly, we avoid iterating over all possible occupancy states by considering only those state transitions that involve a Hamming distance of $d \in \{1, 2, \ldots, K\}$ between two consecutive state vectors, $\vec{B}(i)$ and $\vec{B}(i+1)$,this is practical because the temporal dynamics of the occupancy of the radio environment, dictated by the behavior of the PUs in the network, are typically slower than the processing dynamics of the POMDP agent, i.e., the SU. Secondly, we fragment the discretized spectrum into smaller, independent sets of correlated channels (for example, an 18 channel radio environment with 3 PUs is fragmented into 3 independent fragments, each comprising 6 channels correlated by the occupancy behavior of the corresponding PU); run PERSEUS on these fragments concurrently by employing multi-threading capabilities in software frameworks; and finally, combine the results from each of these fragmented, parallel runs to get a full picture about the performance of our POMDP agent: this is practical because in a radio environment with multiple PUs, each PU is typically restricted to a portion (a set of adjacent frequency bands) of the spectrum, either by design or by bureaucracy.

IV. NUMERICAL EVALUATIONS

Sticking with the single-agent deployment setting, our simulations evaluate the operational capabilities of the proposed POMDP framework and compare it against the state-of-the-art. The simulated radio environment constitutes $J{=}3$ incumbents, i.e., PUs, accessing a 2.88 MHz spectrum, discretized into $K{=}18$ channels, each having a bandwidth of $W{=}160$ kHz, and a cognitive radio node ($\tilde{J}{=}1$ SU) trying to intelligently access spectrum holes to deliver its network flows while limiting incumbent interference, as illustrated in Fig. 1. The 3 PUs access these 18 channels according to a time-frequency Markovian correlation structure parameterized by

$$\vec{\theta} = \begin{bmatrix} \vec{p} & \vec{q} \end{bmatrix}$$

as described in Sec. II-B, where

$$\vec{p} = \begin{bmatrix} p_{00} = 0.1 & p_{01} = 0.3 & p_{10} = 0.3 & p_{11} = 0.7 \end{bmatrix},$$

and

$$\vec{q} = \begin{bmatrix} q_0 = 0.3 & q_1 = 0.8 \end{bmatrix}.$$

Regarding the channel sensing limitations induced by a need to minimize the amount of time and energy spent sensing the spectrum [**?**], we model our simulation framework on $\kappa{=}6$, i.e., in any given time-slot i, the SU can sense a maximum of 6 channels out of the 18 in the discretized spectrum of interest. Regarding the expected Signal to Interference Noise Ratios (SINR) at the PUs and the SU, subject to fading, and conditioned on the PU and SU access decisions, we model our simulation framework based off the following numbers:

$\text{SINR}_{\text{SU}}(k,i){=}0$, if the SU does not access channel k in time-slot i,

$\text{SINR}_{\text{SU}}(k,i){=}11$ dB, if the SU accesses a truly idle channel k in time-slot i,

$\text{SINR}_{\text{SU}}(k,i){=}{-}6$ dB, if SU accesses an incumbent-occupied channel k in slot i,

$\text{SINR}_{\text{PU}_j}(k,i){=}0$, if the PU j does not access channel k in time-slot i,

$\text{SINR}_{\text{PU}_j}(k,i){=}17$ dB, if PU j occupies channel k in slot i without SU interference,

$\text{SINR}_{\text{PU}_j}(k,i){=}6$ dB, if PU j occupies channel k in slot i with SU interference.

As described in Sec. II, the only objective of the SU is to maximize its throughput subject to a constraint on the amount of interference its transmissions can cause to incumbents in the network. To this end, assuming an always back-logged SU, i.e., the SU always has network flows

to deliver, the optimal POMDP sensing policy dictates which channels should be sensed by the SU's spectrum sensor in a given time-slot, based off the learned correlated occupancy dynamics of the PUs, in order to obtain an optimal picture about the occupancy of the channels in this time-slot, and then access all the channels deemed to be idle by the MAP estimation procedure detailed in Sec. III-A. The average throughput attained by the SU over T time-slots is given by

$$C^{\text{SU}} = \frac{1}{T}\sum_{i=1}^{T}\sum_{k=1}^{K} R_{\text{SU}}\mathcal{I}\left\{\text{SINR}_{\text{SU}}(k,i) \geq 2^{\frac{R_{\text{SU}}}{W}} - 1\right\}, \tag{31}$$

where R_{SU}=0.6 Mbps is the transmission rate of the SU on each channel, and $\mathcal{I}$ is an indicator variable; and the throughput attained by the PUs in the network over the same T time-slots, normalized over time and the number of transmissions (normalization is necessary here because of the temporally intermittent transmissions of the PUs, the PUs are not always back-logged, unlike the SU) is given by

$$C^{\text{PUs}} = \frac{\sum_{i=1}^{T}\sum_{k=1}^{K} R_{\text{PU}}B_k(i)\mathcal{J}\left\{\text{SINR}_{\text{PU}}(k,i) \geq 2^{\frac{R_{\text{PU}}}{W}} - 1\right\}}{\sum_{i=1}^{T}\sum_{k=1}^{K} B_k(i)}, \tag{32}$$

where R_{PU}=0.9 Mbps is the transmission rate of the PUs on each channel, $\mathcal{J}$ is an indicator variable, $\text{SINR}_{\text{PU}}(k,i)$=$\text{SINR}_{\text{PU}_j}(k,i)$, $j\in\{1,2,\ldots,J\}$ being the index of the PU occupying channel k in time-slot i, and $B_k(i)$=1 if channel k is occupied by an incumbent in time-slot i (note here that PUs do not interfere with each other because of OFDMA, i.e., due to the clearly laid out administrative guidelines about licensed frequency use for incumbents, so only one PU accesses a frequency band and that band would have been specifically licensed for that PU). In the simulated radio environment described above, we compare our proposed algorithm with the following state-of-the-art solutions proposed in the literature:

- MEM with GC-CCE and MPE [**?**]: Minimum Entropy Merging (MEM) with Greedy Clustering based Channel Correlation Estimation (GC-CCE) and Markov Process Estimation (MPE), Correlation Threshold ρ_{th}=0.7, Number of clusters T=6, i.e., a channel sensing restriction of 6;
- MEM with MEI-CCE and MPE [**?**]: Minimum Entropy Merging (MEM) with Minimum Entropy Increment Clustering based Channel Correlation Estimation (MEI-CCE) and Markov Process Estimation (MPE), Correlation Threshold ρ_{th}=0.7, Number of clusters T=6, i.e., a channel sensing restriction of 6;

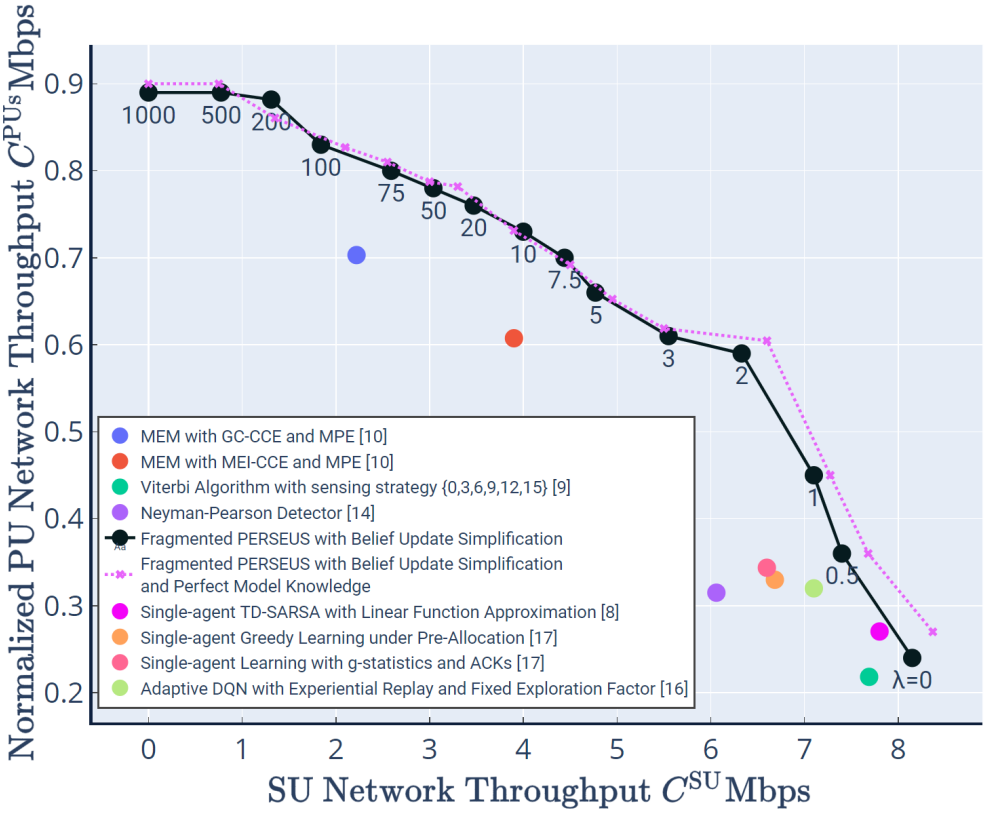

Fig. 7. The evaluation of SU and PU network throughputs for different values of λ, along with comparisons with the state-of-the-art

- Imperfect HMM-MAP State Estimation [**?**]: The Viterbi algorithm, assuming a priori knowledge of the time-frequency Markovian correlation structure in incumbent occupancy behavior, with a channel sensing restriction of 6;
- Neyman-Pearson Detection [**?**]: A Neyman-Pearson Detector, assuming independence across channels and across time, with no channel sensing restrictions, an AND fusion rule across 300 samplings, and threshold determination via a false alarm probability of 30%;
- HMM EM + Fragmented PERSEUS with Belief Update Simplification: The proposed framework in practical settings – A Fragmented PERSEUS algorithm with Belief-Update Simplification (Hamming distance state filters), with no prior occupancy behavior correlation model information, instead, this model is learned over time, concurrently with the optimal channel sensing & access policy solver;
- Prior Perfect Model Knowledge + Fragmented PERSEUS with Belief Update Simplification: The proposed framework in ideal settings – A Fragmented PERSEUS algorithm with

Belief-Update Simplification (Hamming distance state filters), with prior occupancy behavior correlation model information;

- Temporal Difference Learning via SARSA with Linear Function Approximation [**?**]: TD-SARSA with Linear Function Approximation (LFA) in single-agent deployment settings, with a sensing restriction of 6, a belief update heuristic constant λ=0.9, a discount factor of γ=0.9, a fixed exploration factor ϵ=0.01, and a raw false alarm probability of $p_{fa,1}$=5%;
- Greedy Learning under Pre-Allocation [**?**]: Greedy Learning in single-agent deployment settings, with a channel sensing restriction of 6, and a time-varying exploration factor ϵ= $\min(\frac{\beta}{i}, 1)$, where β> $\max(20, \frac{4}{\Delta^2_{\text{min}}})$, with Δ_{min} referring to the smallest Kullback-Liebler distance between a pair of channels;
- g-statistics [**?**]: Learning with g-statistics and ACKs in single-agent deployment settings, with a channel sensing restriction of 6;
- Adaptive Deep Q-Networks [**?**]: An adaptive Deep Q-Network (DQN) with Experiential Replay (Memory Size C=10^6), 2048 input neurons, 4096 neurons with ReLU activation functions in each of the 2 hidden layers of the Neural Network, a Mean-Squared Error cost function with an Adam Optimizer, a Fixed Exploration Factor ϵ=0.1, a Learning Rate of α=10^{-4}, a Batch Size of W=32, and a sensing restriction of 6.

Incorporating a concurrent parameter estimator embedded into the fragmented PERSEUS algorithm (with belief update simplification via the Hamming distance state filter) through an iterative publisher-subscriber routine described in Sec. III-B, we find that our framework outperforms the state-of-the-art algorithms that also tackle the spectrum sensing and access problem in single-agent deployment settings. Specifically, evaluating the performance of our framework against the Minimum Entropy Merging (MEM) algorithm with Greedy Clustering based Channel Correlation Estimation (GC-CCE) and Markov Process Estimation (MPE) [**?**], we find that with a correlation threshold of ρ_{th}=0.77 in the MEM with GC-CCE and MPE solution, our framework achieves a 104% improvement in the throughput attained by the SU, for the same level of interference to the incumbents in the network. Similarly, we find that our solution achieves a 38% improvement in the SU throughput, for the same level of PU interference, over the Minimum Entropy Merging (MEM) algorithm with Minimum Entropy Increment (MEI) Clustering based Channel Correlation Estimation (CCE) and Markov Process Estimation (MPE) with a correlation threshold of ρ_{th}=0.77 [**?**]. Additionally, our solution attains a 25% increase

in SU network throughput, for the same level of interference caused to the PUs in the network, over a Neyman-Pearson Detector that assumes independence among the channels across both frequency and time, has no channel sensing restrictions, involves an AND fusion rule across 300 different samplings, and whose threshold is determined based off of a specified false alarm probability of 0.3 [**?**]. Moreover, comparing the performance of our POMDP framework against well-known HMM state estimators, specifically, the Viterbi algorithm that solves the MAP state estimation problem for the system described in this simulation setup consisting of a two chain Markovian correlation structure (one across time and the other across frequency) [**?**], with the same channel sensing restriction as ours, i.e., 6, but that which knows the exact underlying MDP transition model $\mathbf{A}$ a priori, we note that our solution offers a 6% increase in the attained SU network throughput, for the same amount of incumbent interference. Sticking to the fact that our framework does not know the underlying MDP transition model, which is governed by the correlated PU occupancy behavior, ahead of time, but instead learns this correlation structure as it is interacting with the radio environment and solving for the optimal policy, we evaluated the accuracy of our optimal policy's behavior against a similar PERSEUS agent which knew the transition model beforehand: we find that in the worst-case with respect to the difference in performance between the two, i.e., when $\lambda{=}0$, for the same level of incumbent interference, knowing the correlation model ahead of time only provided a 3.75% increase in the attained SU network throughput, which is a testament to the accuracy of our embedded parameter estimator and the iterative publish-subscribe to-and-fro between the EM thread and the PERSEUS thread.

Evaluating the performance of our framework against Reinforcement Learning strategies in the state-of-the-art such as TD-SARSA with Linear Function Approximation (with a sensing restriction of 6) from [**?**], and an adaptive DQN algorithm (with an experiential replay memory size of 10^6, a fixed exploration factor of 0.1, a learning rate of 10^{-4}, and a batch size of 32) from [**?**], we find that our proposal provides for a 3% boost in SU throughput over the TD-SARSA with LFA framework, and a 9% enhancement in SU throughput over the adaptive DQN framework, for the same level of incumbent interference. Finally, we find that our framework achieves a 10% and a 15% improvement in SU throughput vis-à-vis incumbent interference over greedy learning under pre-allocation and g-statistics with ACKs, respectively from [**?**]. These evaluations are illustrated in Fig. 7.

Analyzing the performance of our POMDP solution from a different perspective, we find that, as illustrated in Fig. 8, our framework obtains an average utility, i.e., $R(\vec{\phi}(i), \hat{\beta}_i)$ described in Sec.

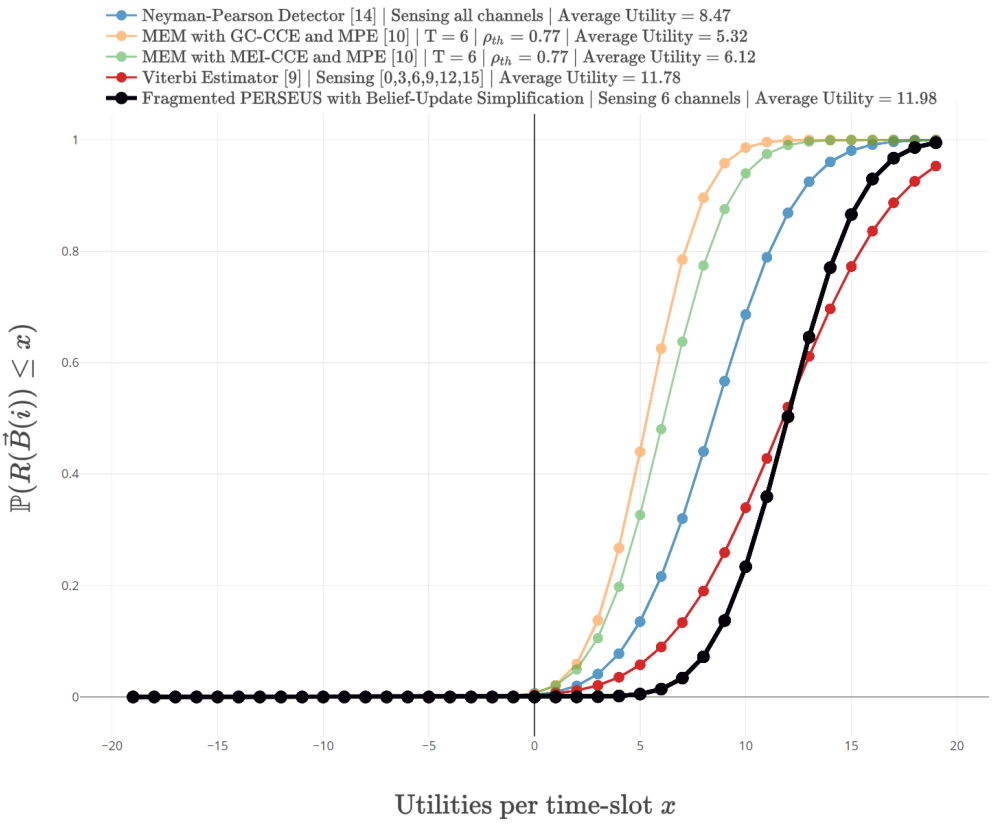

Fig. 8. The evaluation of the proposed solution, from an average utility per time-slot perspective, against a medley of approaches in the state-of-the-art: $\mathbb{P}(R(\vec{B}(i))\leq x)$ versus utility per time-slot x

III-A, of 11.98 per time-step i, 125% higher than that achieved by the MEM with GC-CCE and MPE algorithm from [**?**], 96% higher than that achieved by the MEM with MEI-CCE and MPE algorithm from [**?**], and 42% more than that attained by the Neyman-Pearson Detector detailed above [**?**]. Furthermore, in order to understand how our framework performs against a standard HMM state estimation solution like the Viterbi algorithm described earlier [**?**], especially, one with a priori transition model information and one that senses a maximum of 6 channels per time-slot (channel sensing restriction of 6), we compare our solution with this Viterbi agent, and find that the average utility per time-slot obtained by the Viterbi agent ($=11.78$) is 2% lower than ours ($=11.98$).

As opposed to the double-chain Viterbi algorithm described above, now we separately simulate a single-chain Viterbi algorithm to address the advantage of having more sensing information and to prove a few results about the importance of leveraging the correlations in incumbent occupancy behavior across frequencies, which many works in the state-of-the-art fail to do.

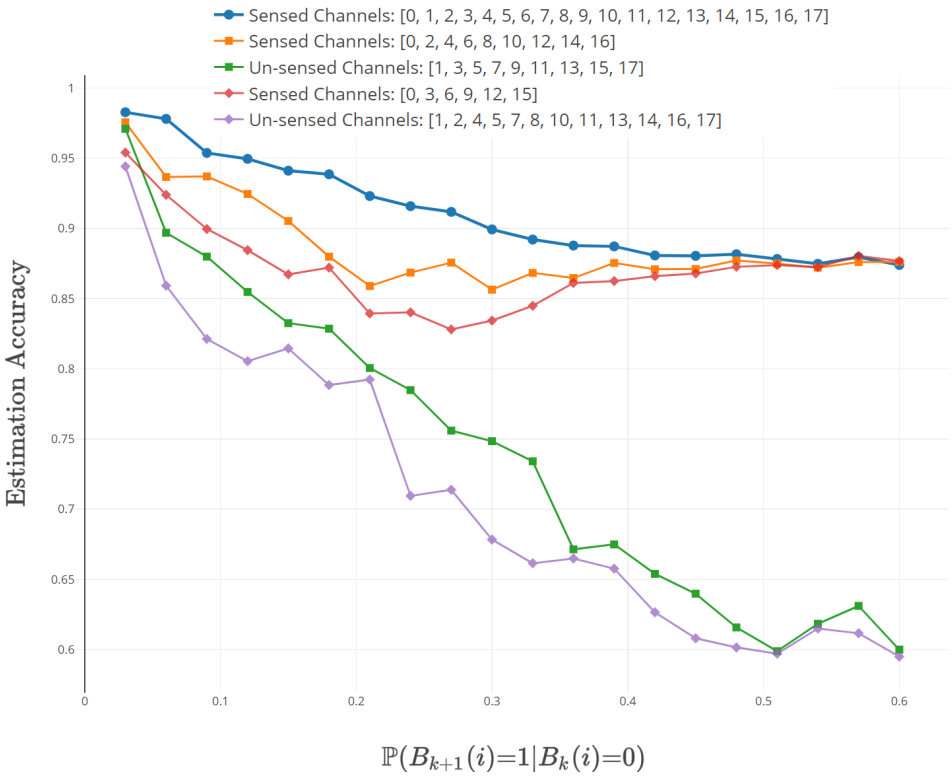

Fig. 9. The evaluation of estimation accuracies for different channel sensing strategies, corresponding to a frequency-correlation Markov chain Viterbi algorithm, parameterized by $\mathbb{P}(1|0)$, in relation to variations in the amount of correlation

Addressing the advantage of having more sensing information, Fig. 9 drives home the point that sensing more channels improves the accuracy of the HMM state estimator, i.e., the Viterbi algorithm discussed here, which in turn gives the SU a better occupancy picture. As we go down the plot in Fig. 9, i.e., as the number of channels sensed per time-step decreases, the estimation accuracy, which refers to the number of channels whose states (0 or 1) were correctly estimated by the Viterbi algorithm, mathematically described as $\sum_{k=1}^{K} \mathcal{L}\left\{B_k(i)=\hat{B}_k(i)\right\}$, where $\mathcal{L}$ is an indicator variable, $B_k(i)\in\{0,1\}$ is the true occupancy state of the channel in time-slot i, and $\hat{B}_k(i)\in\{0,1\}$ is the occupancy state of the channel estimated by the Viterbi algorithm, averaged over 300 sampling rounds, decreases consistently. Also, note that the estimation accuracies of the un-sensed channels are expectedly worse than those of the sensed channels. An additional points about the effects of correlation in incumbent occupancy behavior across frequency on the accuracy of state estimation can be made by analyzing Fig. 9 from a different perspective: as the incumbent occupancy behavior becomes more and more correlated across frequency, as denoted

by the X-axis, i.e., as $\mathbb{P}(B_{k+1}(i){=}1|B_k(i){=}0), \forall 1{\leq}k{\leq}K$ moves from a highly correlated model ($\mathbb{P}(B_{k+1}(i) = 1|B_k(i) = 0) = 0.1 > 0.6 = \mathbb{P}(B_{k+1}(i){=}1)$, $\forall 1{\leq}k{\leq}K$) to an independence model ($\mathbb{P}(B_{k+1}(i){=}1|B_k(i){=}0){=}0.6{=}\mathbb{P}(B_{k+1}(i){=}1), \forall 1{\leq}k{\leq}K$), the estimation accuracy decreases. Therefore, this evaluation of the single-chain Viterbi algorithm in a separate, hypothetical simulation model proves that we can achieve an improved estimation of incumbent occupancy behavior by sensing more channels per time-step and by leveraging the correlations in PU occupancy behavior across frequencies. However, as already noted, the number of channels that can be simultaneously sensed by the SU in a given time-step is restricted by design limitations [**?**], hence, fixing $\kappa{=}6$ in our POMDP solution, we resort to leveraging the correlation in incumbent occupancy behavior across frequency (and time) in order to attain better state estimation performance. Furthermore, we find that adapting the spectrum sensing decision based on the system state (true or perceived) [**?**], in contrast to a fixed sensing strategy [**?**], [**?**], adds to the performance gains attained by exploiting the PU occupancy correlation. As already discussed, our proposed framework can be decomposed into two components: the parameter estimator and PERSEUS. Next, we take up each of these two individually and analyze their performance.

Specifically discussing the performance of the parameter estimation algorithm, i.e., the HMM EM algorithm (Baum-Welch), we find that, with initial estimates of 0.5, i.e., $p_{uv}{=}0.5, \forall u, v{\in}\{0, 1\}$ and $q_w{=}0.5, w{\in}\{0, 1\}$, the estimator converges to the true parameter vector $\vec{\theta}$ with an error/delta of $\eta{=}10^{-8}$ in $45,000$ iterations: this corresponds to an observation and estimation period of 135 s, considering a typical time-slot duration of 3 ms. We illustrate this convergence via the Mean Square Error (MSE) plot depicted in Fig. 10, in which the MSE in iteration t given by,

$$||\vec{\theta} - \hat{\vec{\theta}}^{(t)}||_2^2 = \sum_{\theta \in \vec{\theta}} \mathbb{E}[(\theta - \hat{\theta}^{(t)})^2] \tag{33}$$

is decreased iteratively, as the estimation process goes through the E-step and the M-step in each iteration t until $\mathbb{E}[\theta{-}\hat{\theta}^{(t)}]{\leq}10^{-8}, \forall \theta{\in}\vec{\theta}$.

On the same time-scale as the parameter estimation algorithm, focusing on the loss convergence of the PERSEUS algorithm with a discount factor of $\gamma{=}0.9$ and a termination threshold of $\epsilon{=}10^{-5}$, wherein we define the expected loss as the difference between the utility obtained by the proposed PERSEUS framework, denoted by $R_P(\vec{B}(i))$ (discussed in Sec. III-A), and that obtained by an Oracle, which knows the exact occupancy behavior of the incumbents in the network, denoted by $R_O(\vec{B}(i))$, we find that, as depicted in Fig. 10, the loss convergence of

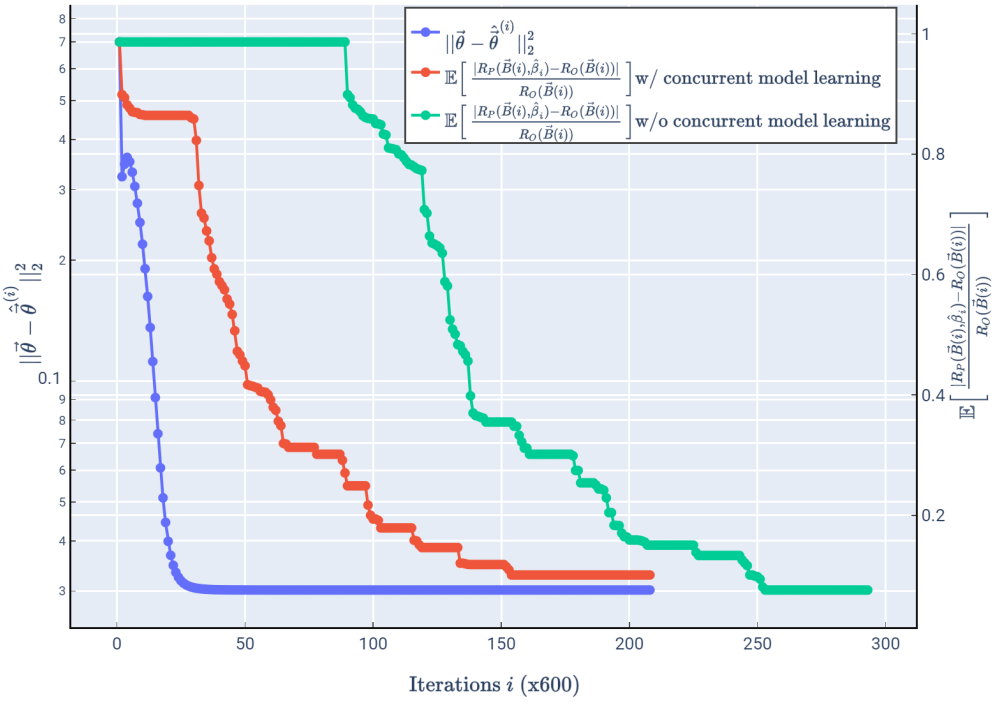

Fig. 10. The convergence of the MSE of the HMM EM algorithm to estimate $\vec{\theta}$, and the convergence of the loss of the fragmented PERSEUS algorithm with belief update simplification

PERSEUS is relatively slower while the parameter estimator is learning the transition model; as opposed to after the convergence of the parameter estimator, when we see a more consistent gradient towards the optimality. Also, note the normalized sub-optimality gap of 0.05, i.e., the average normalized difference between the utility obtained by our optimal POMDP policy (post-convergence) and the utility obtained by the Oracle (which knows the exact incumbent occupancy behavior) is 0.05. Moreover, Fig. 10 depicts the computational time difference between running the parameter estimator and the PERSEUS algorithm concurrently via the iterative publisher-subscriber architecture, as opposed to initiating the PERSEUS run after the convergence of the parameter estimator: we cut down the time to completion of our HMM-POMDP framework by half by employing the former approach as opposed to the latter, without worsening the sub-optimal gap significantly.

Finally, inspecting Fig. 7 in a new light, we see that our POMDP agent limits channel access when the penalty (λ) is high, leading to lower SU throughput and lower PU interference, and conversely, follows a more lenient channel access strategy when the penalty is low, resulting in

higher SU throughput and higher PU interference. Generally speaking, Fig. 7 depicts a trend of increasing SU throughput and increasing incumbent interference, as the penalty for missed detections, i.e., λ is lowered. Therefore, our framework provides a crucial practical tool in cognitive radio MAC design: the ability to tune the trade-off between the throughput obtained by the cognitive radio and the interference caused by it to incumbent transmissions in the network.

V. Multi-Agent Deployment Model: An Extension to the Single-Agent Setting

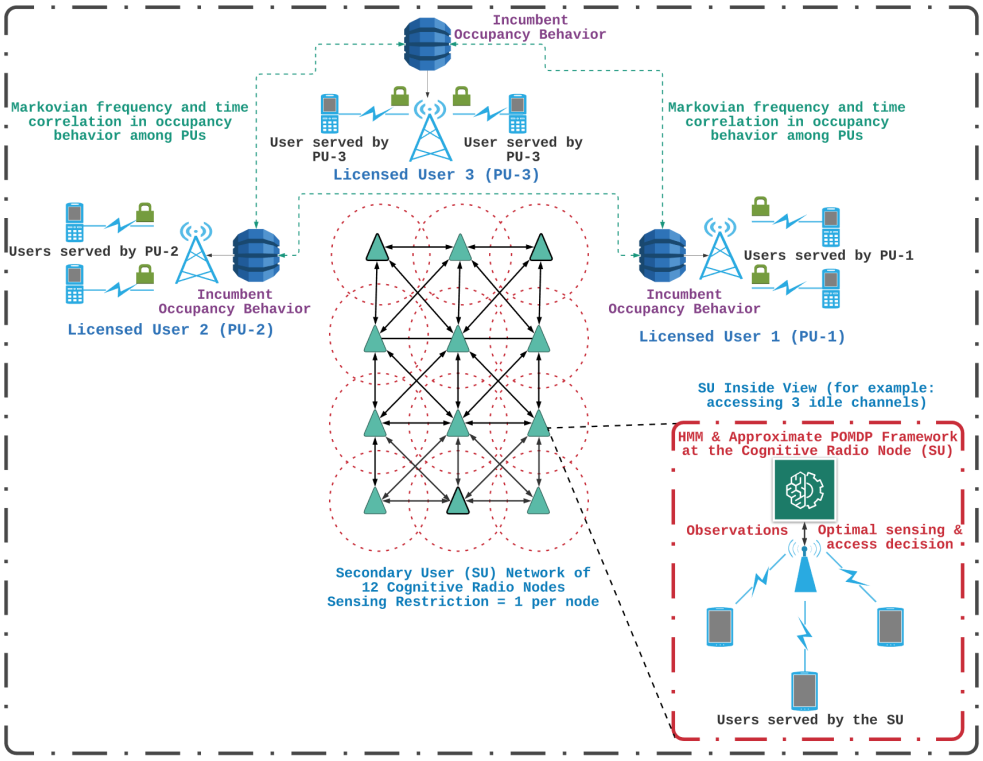

Fig. 11. The deployment setup of a distributed multi-agent cognitive radio network with 3 PUs, 12 SUs, and a channel sensing restriction of 1 per SU per time-slot, in an 18 channel radio environment **[NM: isnt this similar to Fig 1? It seems quite redundant, please remove or merge them together.. no need to have both]**

In this section, we evaluate the performance of the proposed framework: HMM EM + Fragmented PERSEUS with Belief Update Simplification, in distributed multi-agent deployment settings. Operating under the same signal and observation models as in Sec. II, consider a network of 3 PUs operating in an 18-channel radio environment, with their occupancy behaviors in this discretized spectrum of interest governed by Markovian time-frequency correlation structure ($\vec{\theta}$), and 12 SUs intelligently trying to access white-spaces in the spectrum (cooperatively [**?**] or

opportunistically [**?**]), with an added restriction of being able to sense only 1 channel per SU per time-slot, as illustrated in Fig. 11.

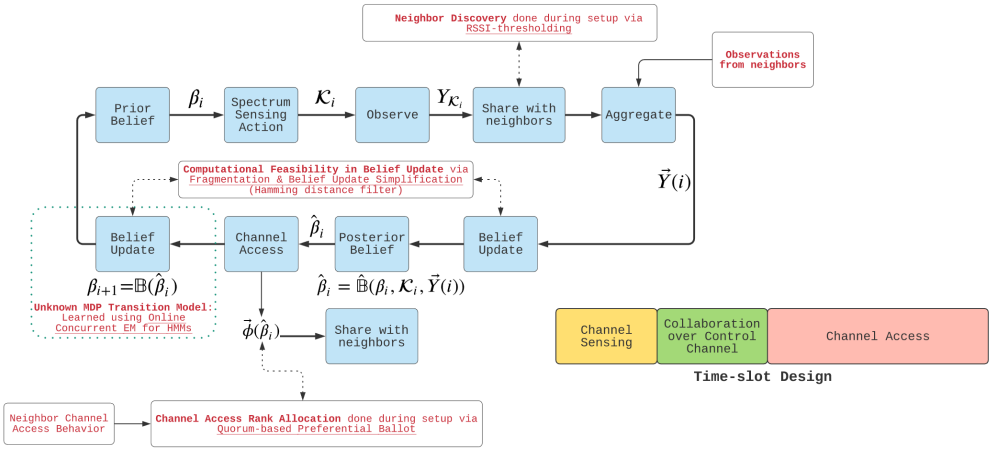

Fig. 12. The POMDP flow at an SU and the associated time-slot design for the multi-agent deployment analysis

The POMDP model described in Sec. III-A has been adapted to this multi-agent deployment setting by incorporating neighbor discovery, channel access rank allocation, and data aggregation algorithms into the original POMDP process flow, as depicted in Fig. 12. Designating the band-edges as the control channel, for neighbor discovery, each cognitive radio node broadcasts its control frames (with a frame header and node identifier) over the control channel, and upon receiving control messages from all its surrounding nodes, each cognitive radio node checks if the expected RSSI of the radio signals corresponding to a certain node is above a threshold RSSI_{th}: if yes, adds that node's identifier to its list of neighbors. With a similar control channel strategy for channel access rank allocation, we employ a quorum-based preferential ballot voting scheme to determine the order in which the "estimated-idle" channels are accessed by the SUs in the network. This procedure kicks in only after a quorum has been achieved, i.e., the number of neighbors identified by an SU should be equal to or exceed a node-specific pre-defined number. Over the control channel, each cognitive radio exchanges a ranked list of its neighbors in the decreasing order of their respective RSSIs, with itself being on the list at position-1 (ties are broken via uniform random choice). Upon receiving an "RSSI-ranked" list from one of its neighbors, each cognitive radio node assigns points to each ranked position, with higher ranks getting larger point values, and re-broadcasts an "aggregated-ranked" list of neighbors (with itself being on the list) with the ranking based on the point-values aggregated across all the ranked

lists received from its neighbors (ties are broken via uniform random choice). If the "aggregated-ranked" lists received from its neighbors matches the one at the SU, and this is true for a pre-specified consecutive period of time, a consensus has been reached, the channel access order is determined by this "harmonized-aggregated-ranked" list. If the "aggregated-ranked" lists received from its neighbors differ from the one at the SU, then the SU repeats the re-ranking of these list members based on their new aggregated point-values and broadcasts the new "aggregated-ranked" list to its neighbors over the control channel. This repetitive process continues until a consensus is reached.

Analyzing the performance of the proposed framework (HMM EM + Fragmented PERSEUS with Belief-Update Simplification) against other distributed multi-agent schemes in the state-of-the-art, as shown in Fig. 13, we find that our framework, in terms of the average utility $R(\vec{\phi}(i), \hat{\beta}(i))$ obtained per time-slot, out-performs the distributed, cooperative, ϵ-greedy TD-SARSA with Linear Function Approximation framework from [**?**] by 43%; out-performs the distributed, cooperative, time-decaying ϵ-greedy algorithm with channel access rank pre-allocations from [**?**] by 84%; and out-performs the distributed, opportunistic, g-statistics algorithm with ACKs (without channel access rank pre-allocations) from [**?**] by 324%.

VI. DARPA SC2: Active Incumbent Retrofit Emulation

In order to evaluate the performance of the proposed framework (HMM EM + Fragmented PERSEUS with Belief Update Simplification) in real-world settings, we retrofit it into the MAC layer (channel & bandwidth allocation) of our BAM! Wireless radio [**?**] (designed for the DARPA SC2, see Fig. 15), and analyze its operational capabilities in the DARPA SC2 Active Incumbent scenario [**?**] emulated on the Colosseum [**?**], [**?**]. The DARPA SC2 Active Incumbent scenario consists of a Terminal Doppler Weather Radar (TDWR) system functioning as the PU, and 5 competitor networks (ours included), each constituting 2 UNII WLANs: 2 Access Points (APs) and 4 STAtions (STAs) per AP, serving as the SUs, in a 10 MHz radio environment (995 MHz to 1005 MHz), for 330 seconds of emulation on the Colosseum [**?**], as illustrated in Fig. 14.

During the Active Incumbent scenario emulation, every competitor network receives network flows from the Colosseum which need to be delivered to the appropriate destination nodes within the network, while satisfying the imposed QoS mandates per flow (for example: max_latency, min_throughput, file_transfer_deadline, etc.). If the QoS mandates imposed on a particular network flow have been satisfied for a pre-specified period of time (referred to as "Measurement

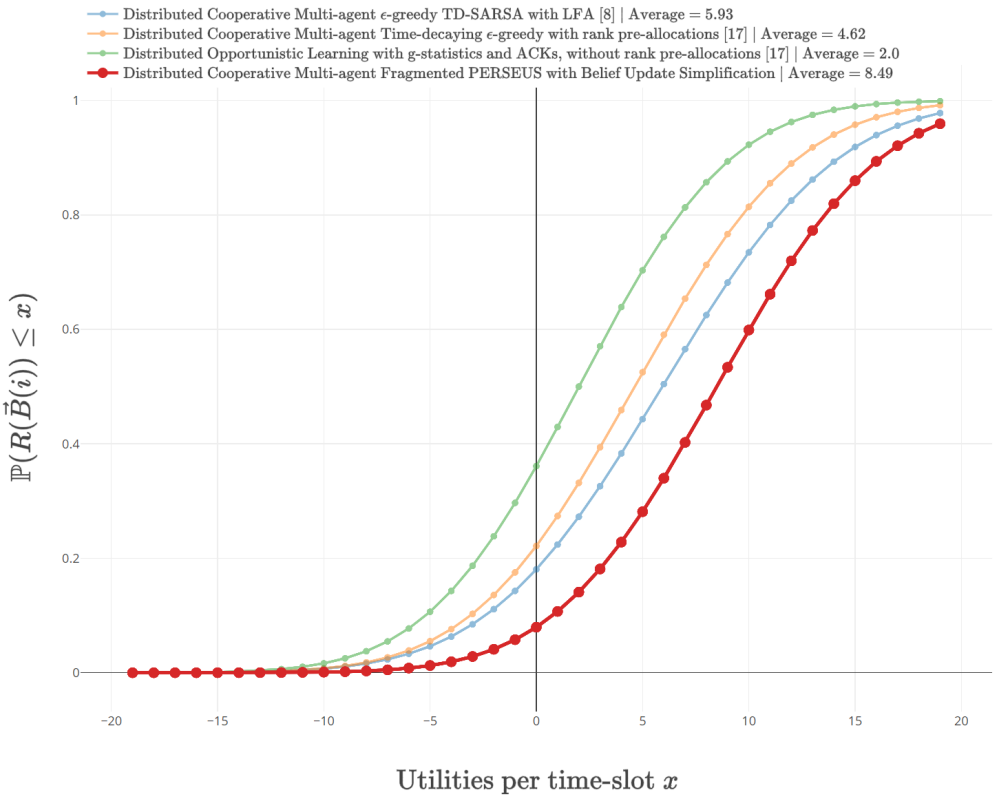

Fig. 13. An evaluation of the performance (average utility per time-slot) of the proposed framework in a distributed multi-agent deployment setting, against other distributed cooperative/opportunistic multi-agent channel sensing & access frameworks in the state-of-the-art: $\mathbb{P}(R(\vec{B}(i))\leq x)$ versus utility per time-slot x

Periods" (MPs)), then the Individual Mandates (IMs) associated with the flow are said to have been met. With this concept of IMs in mind, we can define the points achieved or the "score" of a participant network corresponding to a certain time-slot i as $\sum_{v\in\mathcal{V}_i} \mathrm{p}_v$, where $\mathcal{V}_i$ denotes the set of IMs achieved by a participant network in time-slot i. The scenario also incorporates ensemble performance thresholds, i.e., all the participant networks should meet the scoring threshold of 8 [**?**]: if a participant network fails to meet this threshold, all the participant networks get the lowest score, i.e., the score corresponding to that achieved by this under-performing network, else, if all the participant networks in the emulation achieve scores that exceed the threshold, their scores are incremented beyond this threshold commensurate with the IMs achieved by them in that time-slot.

After having understood the scoring mechanism involved in the DARPA SC2, we can now evaluate the performance of the proposed framework retrofitted into our standard BAM! Wireless

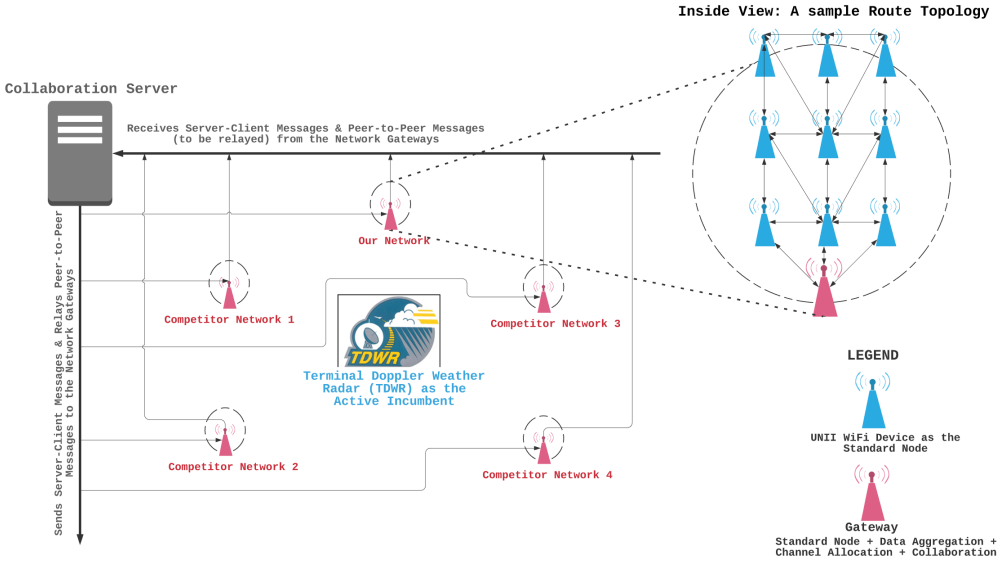

Fig. 14. The deployment setup of the DARPA SC2 Active Incumbent scenario emulated in the Colosseum: the TDWR system serves as the PU, 5 competitor networks with 2 UNII WLANs (2 APs and 4 STAs per AP) with individual nodes in our network retrofitted with our multi-agent POMDP framework, 10 MHz scenario bandwidth, and 330 seconds of emulation [**?**]

radio [**?**] against other radios designed by our peers who also participated in this competition, in addition to a performance comparison with the weighted PSD + CIL [**?**] channel & bandwidth allocation scheme employed, as a standard out-of-the-box protocol, in our traditional BAM! Wireless network. We showed in Sec. II-B that the proposed Markovian time-frequency correlation structure fits the actual occupancy behavior of the incumbent and fellow competitors very well, with the Kullback-Liebler divergence analysis yielding a model fitting loss of 0.05997 nats, which is significantly better than the conventional models (independence, purely temporal correlation, and purely frequency correlation) employed in the state-of-the-art: the reason being that, during our channel & bandwidth allocation analyses post-emulation, we found that the radios in the network (incumbent + competitors) regularly occupy adjacent channels, leading to frequency correlation in their occupancy behaviors, and they occupy these frequency bands for a prolonged period of time unless disturbed by rogue transmissions or transmissions by other radios, exhibiting "inertia" in their occupancy behaviors, which results in the temporal correlation we discuss in this paper. Leveraging the aggregated PSD measurements obtained at the gateway node of our BAM! Wireless network, as shown in Fig. 16, and the estimated link SNRs, as depicted in Fig. 17, we evaluate the scores of the proposed framework retrofitted

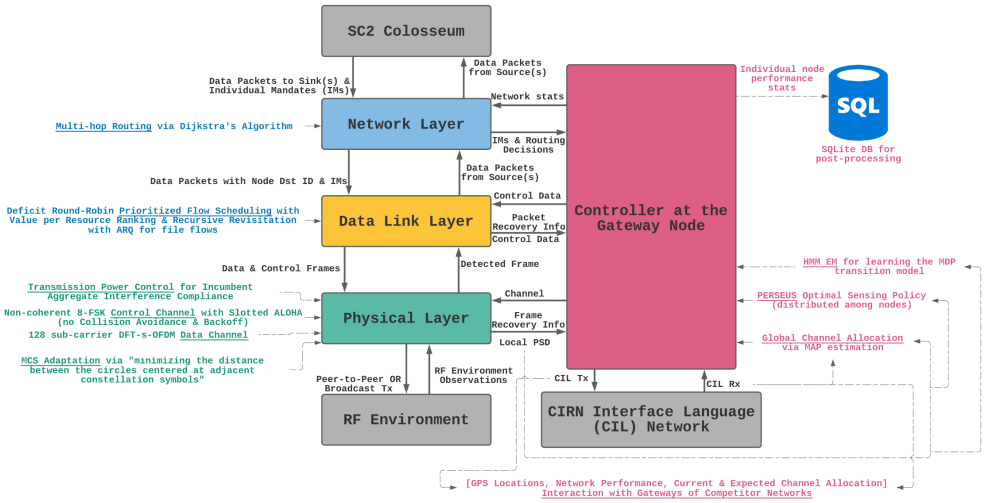

Fig. 15. The design of our BAM! Wireless radio [**?**], with layer-specific protocol/algorithm description: the gateway node (aggregator) in our network performs channel & bandwidth allocation via the proposed POMDP framework **[NM: please remove this figure, there are too many details that we dont have room to explain.. ask yourself: is this adding any useful information to THIS paper? I dont think so]**

into our standard BAM! Wireless radios against our traditional channel & bandwidth allocation scheme (titled "Standard BAM! Wireless Radio [Purdue]), and against the designs of our peers (identified by their collaboration network registered IP address [**?**], "172.30.210.191 [Peer]" and "172.30.210.181 [Peer]"): in terms of the average score achieved per time-slot, we deduce from Fig. 18 that the proposed framework ("BAM! Wireless Radio + HMM EM + Fragmented PERSEUS with Belief Update Simplification") out-performs our traditional channel & bandwidth allocation scheme (a simple weighted PSD + CIL heuristic) by 21%; provides a 56% better performance than one of our peers, identified by "172.30.210.181"; and attains an 81% boost in performance over another one of our peers, identified by "172.30.210.191".

VII. Feasibility Analysis of the POMDP Optimal Policy on ESP32 Radios

We employ 8 ESP32 radios [**?**], with each one embedded in a GCTronic e-puck2 robot [**?**], categorized into a network of 3 PUs (and their 3 corresponding sinks) occupying 6 channels in the discretized spectrum of interest according to a Markovian time-frequency correlation structure (described by (6)), and 2 independent SUs, with each having the capability of sensing only one channel at a time, intelligently trying to exploit the white-spaces in the spectrum. The detailed methodology of this implementation is provided below:

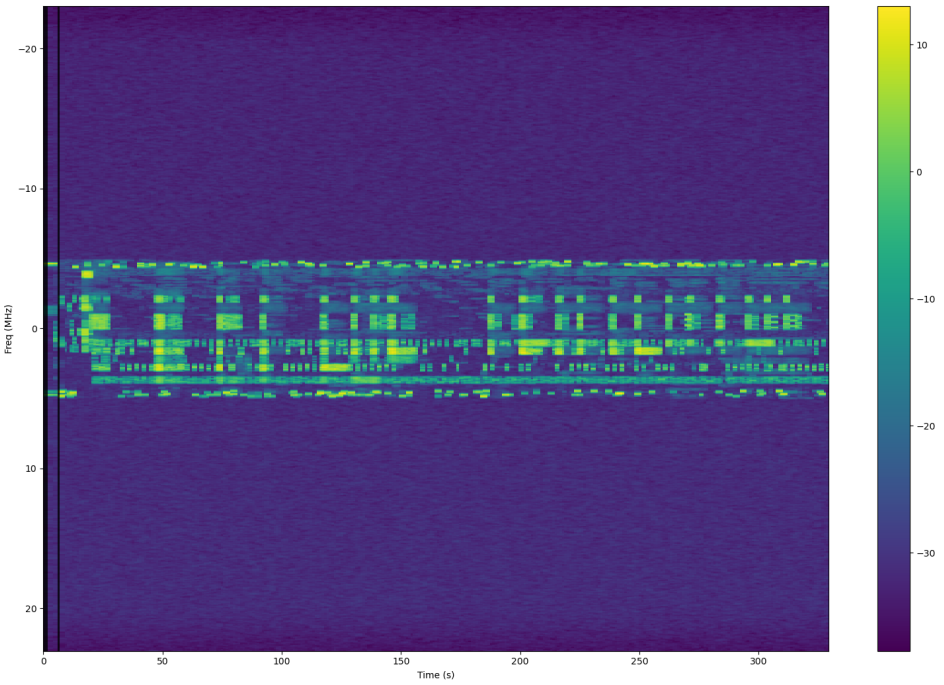

Fig. 16. A plot of the PSD observations made at a cognitive radio node within our network (Radio_ID: 99) during the DARPA SC2 Active Incumbent scenario emulation on the Colosseum. The aggregated PSD measurement map which is used for channel & bandwidth allocation at the gateway is illustrated in Fig. 4 (L)**[NM: this is same as Fig 4, please do not repeat figures, remember we have strict page constraints]**

- Considering a network with $J{=}3$ PUs and one SU (work split over 2 ESP32 radios due to design limitations) with a channel sensing restriction of $\kappa{=}2$ out of $K{=}6$ channels in the discretized spectrum of interest, and assuming a linear AWGN observation model, with a Rayleigh channel fading model (discussed in Sec. II-A), we simulate the occupancy behavior of the PUs according to a Markovian time-frequency correlation structure parameterized by $\vec{\theta}{=}[\vec{p}, \vec{q}]^\intercal$, where $\vec{p}{=}[p_{00}{=}0.1, p_{01}{=}0.3, p_{10}{=}0.3, p_{11}{=}0.7]^\intercal$ and $\vec{q}{=}[q_0{=}0.3, q_1{=}0.8]^\intercal$; and solve for the optimal spectrum sensing and access policy using PERSEUS, embedded with a concurrent parameter estimation algorithm learning the parameter vector $\vec{\theta}$, by mimicking the observational capabilities of the actual ESP32 radios. Note this step is performed on a PC.
- The simulated PU occupancy behavior, Markovian correlated according to (6) and parameterized by $\vec{\theta}$, and the time-slot specific optimal channel access decisions (derived off of the

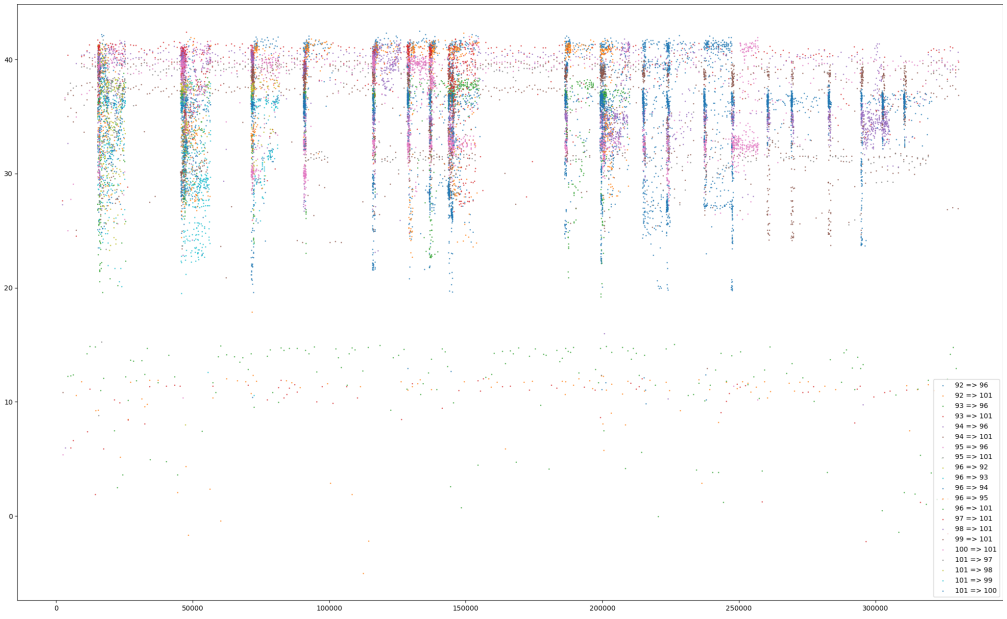

Fig. 17. A plot of the link SNRs observed during the DARPA SC2 Active Incumbent scenario emulation on the Colosseum**[NM: this figure is quite uninformative.. remove? OR replace with something more meaninfgul, such as CDF]**

POMDP optimal sensing policy and the simulated PU occupancy behavior), are stored in databases (for export onto the ESP32 network).

- Peer-to-Peer communication links are established between a PU ESP32 radio and its sink, using the 3 ESP32 radios designated as PUs. In other words, 3 wireless communication links are established: one for each ESP32 PU pair (a source and a sink), over WiFi (2.4 GHz) and using a channel according to the occupancy information detailed in the exported PU occupancy database, in time-slot i.
- Note here that in this ESP32 PU network implementation, in time-slot i, while establishing a wireless communication link between a ESP32 PU $j \in \{1, 2, 3\}$ and its respective sink $i \in \{1, 2, 3\}$ s.t. i is the designated sink for PU j, i.e., while forming link l_{ij} over channel $k_{l_{ij}} = k \in \{1, 2, \ldots, 6\}$ (as determined by the exported PU occupancy database which contains simulated PU occupancy behavior according to the Markovian time-frequency correlation structure described above) such that $k_{l_{ij}} \neq k_{l_{i',j'}}, \forall i, i' \in \{1, 2, 3\},\ j, j' \in \{1, 2, 3\}$, PU j serves as an Access Point (AP) accepting transmission requests from PU i, which is designated as a STAtion (STA). In the next synchronized time-slot $i + 1$, this link l_{ij} moves to channel $k' \in \{1, 2, \ldots, 6\}$, as detailed in the exported PU occupancy database. This same procedure

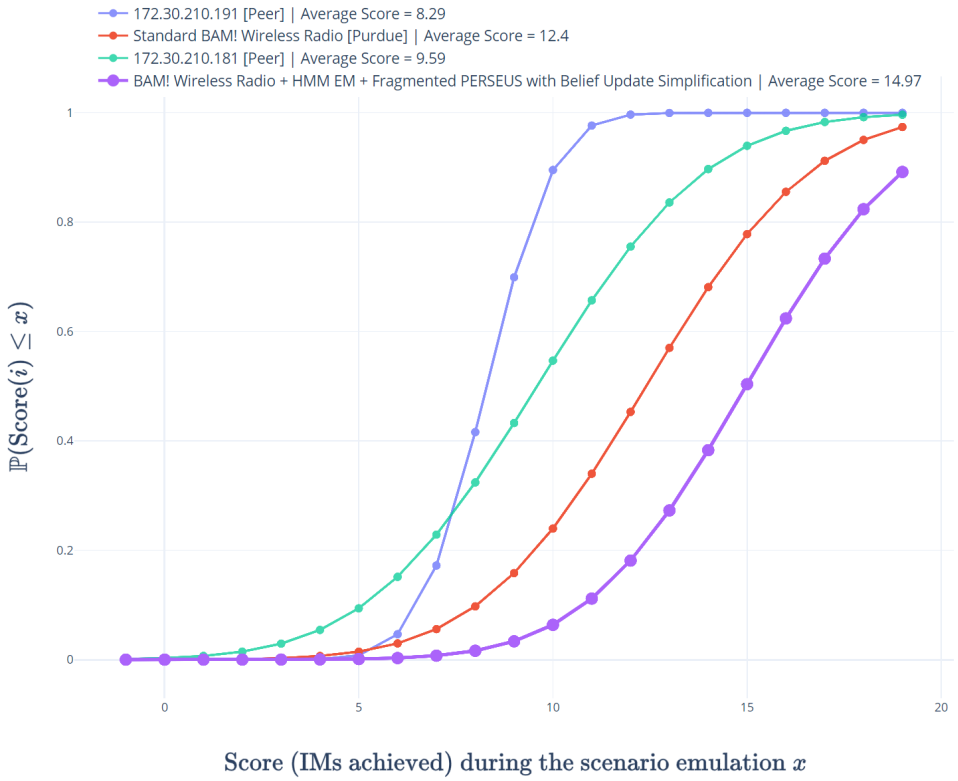

Fig. 18. An evaluation of the performance (scores/Individual Mandates (IMs) achieved) of our solution by retrofitting the proposed POMDP framework into our BAM! Wireless cognitive radio network design, with respect to an emulation of the DARPA SC2 Active Incumbent scenario, against other competitor network radio designs: $\mathbb{P}(\text{Score} \leq x)$ versus the scores achieved during the course of this emulation x

takes place for the other two incumbent communication links in every time-slot until the end of the implementation evaluation period.

- Although the PC-based POMDP solver employs an SU which can access 2 channels at a time in order to deliver its flows (see the access part of the POMDP formulation in Sec. III-A), we employ 2 ESP32 SU radios in the network (serving as one), with the channel access work synchronously and evenly split between the two, due to the actual physical design limitations of the ESP32 radio that it can only access one channel at a time, forcing us to be creative: split the optimal 2 channel access decision in time-slot i, as determined by the time-slot specific optimal POMDP channel access database, into a 1 channel access action at each ESP32 SU radio. Next, based on whether the channel access at the 2 ESP32 SU radios was successful, we compute the success rate.

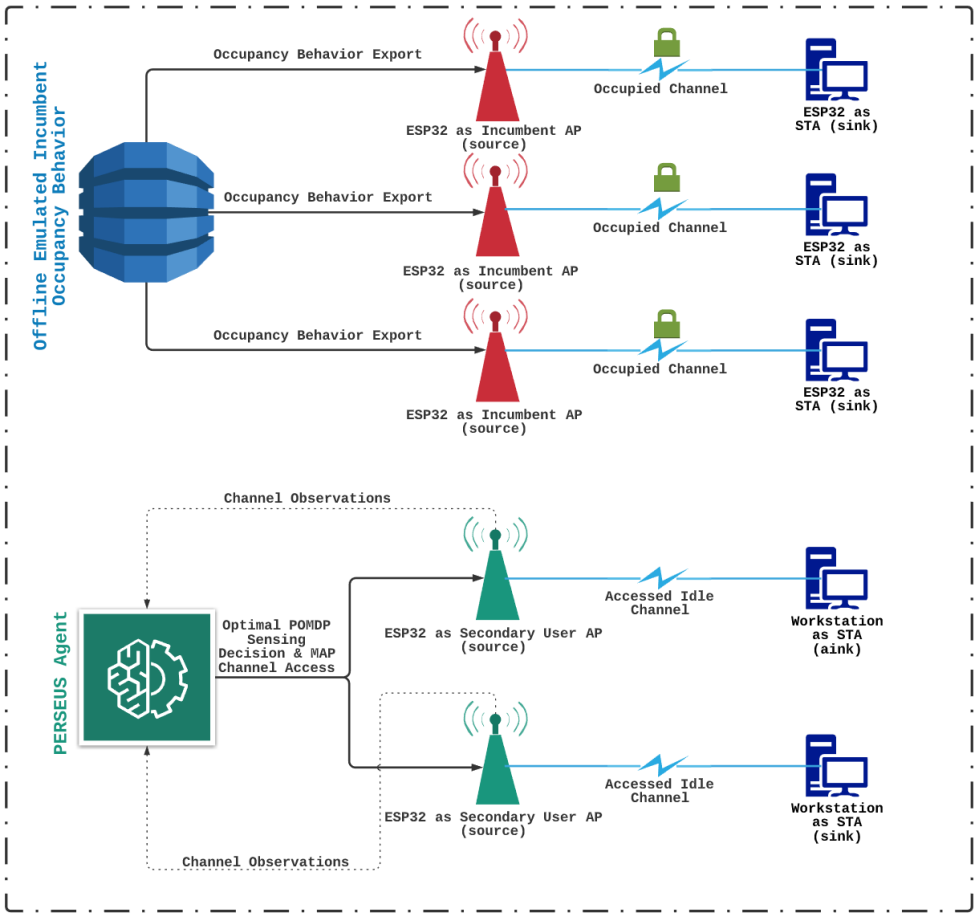

Fig. 19. The deployment setup of a distributed ad-hoc WLAN network for feasibility analysis of the PERSEUS optimal policy **[NM: do we need this figure orcan you reuse a previous one?]**

The deployment setup of this distributed ad-hoc WLAN network for evaluating the implementation feasibility of the POMDP optimal policy is illustrated in Fig. 19. The channel access success rate metric defined as

$$\text{Channel Access Success Probability} = \frac{\sum_{j=1}^{2} \mathcal{I}\left\{B_{k_{SU_j}}(i) = 0\right\}}{2}, \tag{34}$$

where $\mathcal{I}$ corresponding to $\mathcal{I}\left\{B_{k_{SU_j}}(i) = 0\right\}$ is an indicator variable whose value is 1 if the channel accessed by the ESP32 SU $j \in \{1, 2\}$ in time-slot i is not occupied by an incumbent PU ESP32 radio, and $B_{k_{SU_j}} \in \{0, 1\}$ is the occupancy variable of the channel accessed by the ESP32 SU j in time-slot i, is evaluated per time-slot i, and the resultant metrics are plotted against time, which is illustrated in Fig. 20.

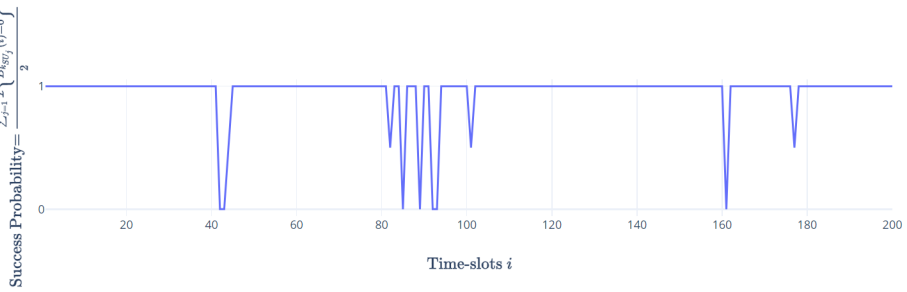

Fig. 20. The channel access success probability of the ESP32 SU radios per time-slot**[NM: no need to probvide a figure when yu can describe this figure by a few numbers, i.e. numerical values of misdetection and false alarm probabilities, or detection error probability]**

VIII. Conclusion

[NM: too long! please cut in half] Firstly, in single-agent deployment settings, we formulate the optimal spectrum sensing and access problem in cognitive radio networks via approximate POMDPs: in radio environments with a single cognitive radio node restricted in the number of channels it can sense per time-slot, and multiple licensed users wherein the occupancy behavior of these incumbents is correlated across both time and frequency, we present a framework that employs the Baum-Welch algorithm (HMM EM) to estimate the transition model of this occupancy behavior, and leverage these learned statistics in a fragmented PERSEUS algorithm with belief update simplification (via Hamming distance state filters) to concurrently solve for the optimal spectrum sensing and access policy. In addition to its superior performance compared to the single-agent state-of-the-art, our framework facilitates regulation of the trade-off between SU throughput and PU interference. Secondly, extending our single-agent model to distributed multi-agent settings, with neighbor discovery (via RSSI thresholding) and channel access rank allocations (via quorum-based preferential ballot voting), we demonstrate superior performance over both collaborative and opportunistic distributed multi-agent state-of-the-art. Thirdly, evaluating the performance of our POMDP framework (concurrent HMM EM and fragmented PERSEUS with Hamming distance state filters) in centralized multi-agent settings by retrofitting it into our DARPA SC2 radio (BAM! Wireless) and emulating a real-world TDWR-UNII WLAN scenario, we illustrate that our solution achieves higher scores over our fellow competitors. Finally, in order to study the implementation feasibility of our solution in practical settings, we test it on an distributed ad-hoc wireless platform of ESP32 radios, and prove seamless

execution.