2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN) | 978-1-6654-1339-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/DySPAN53946.2021.9677439

Robust Neural Network-Based Spectrum Occupancy Mapping

Abbas Termos
Department of Electrical Engineering
University of Notre Dame
Notre Dame, USA
atermos@nd.edu

Bertrand Hochwald
Department of Electrical Engineering
University of Notre Dame
Notre Dame, USA
bhochwald@nd.edu

***Abstract*—We present a neural network decision system for determining if spectrum is occupied in a region. Given a threshold, we wish to determine if power at a given frequency exceeds the threshold, thus determining if that frequency is "occupied". The emitting sources are unknown in number, locations, and powers. The sensors, which measure the signal power, are random in number and location. The measurements are aggregated as log-likelihood ratios into a fixed-resolution image suitable as input to a neural network. The network is trained to produce an occupancy map over a wide area, even where there are no sensors, and achieves excellent accuracy at determining occupancy. The system is robust to the number of sensors, and occupancy threshold in a variety of environments.**

***Index Terms*—Spectrum occupancy map, binary decisions, deep learning, robust spatial modeling**

I. Introduction

We present a neural network framework for computing spectrum decision maps over a region from a limited number of sensor measurements. The sensors measure signal power at a frequency and bandwidth of interest, and are random in number and location. The number, location, and power of the emitters are entirely unknown. The decisions to be made are whether the signal power at *any* location is above a prescribed power threshold.

Spectrum occupancy maps are enabled by low-cost sensors such as RadioHound [1] developed at the University of Notre Dame. The sensor is in its third generation, featuring an 8-bit 48 MSps A/D, on-board GPS, 45 dB tunable gain, high sensitivity (−120 dBm noise floor), high linearity ($P_{1dB} = -20$ dBm), and sub-$50 cost.

We show that even in a severely sub-sampled region with very few such sensors, the performance of the decision system across the whole region is excellent and robust to the number of sensors and occupancy threshold in a variety of environments.

The authors gratefully acknowledge joint funding by Intel and the National Science Foundation under Grant ECCS-2002921.

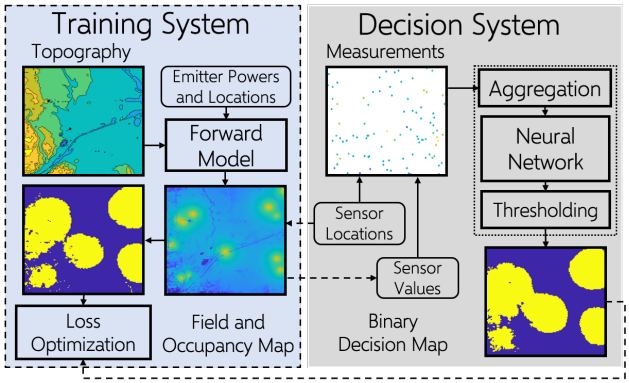

Fig. 1. Neural network-based decision maps from sensor measurements. The sensors measure power at a limited number of locations. The training system (left) supervises the training with a collection of occupancy maps and corresponding sensor measurements. The decision system (right) aggregates sensor measurements into fixed-resolution images that are sent to a neural network. The neural network then produces a decision map also in the form of an image. The dashed traces indicate information flow during training.

A. Setup

The block diagram in Fig. 1 illustrates the training and decision systems for learning occupancy maps over a region $\mathcal{R}$. The training system (outlined with dashed lines) stores examples of the field and occupancy maps.

Denote the set of emitters (coordinate) locations by $\mathcal{E} = \{\mathbf{e}_l\}_{l=1}^{n_\mathcal{E}}$, where $n_\mathcal{E}$ is the number of emitters; these quantities are generally unknown during deployment. Similarly, denote the set of $n_\mathcal{S}$ sensors locations by $\mathcal{S}$; $\mathcal{S}$ and $n_\mathcal{S}$ are known during deployment. A sensor at $\mathbf{r} \in \mathcal{S} \subset \mathcal{R}$ then measures the power from the (uncorrelated) emitters as

$$f(\mathbf{r}) = \sum_{l=1}^{n_\mathcal{E}} p_l f_{\mathbf{e}_l}(\mathbf{r}), \tag{1}$$

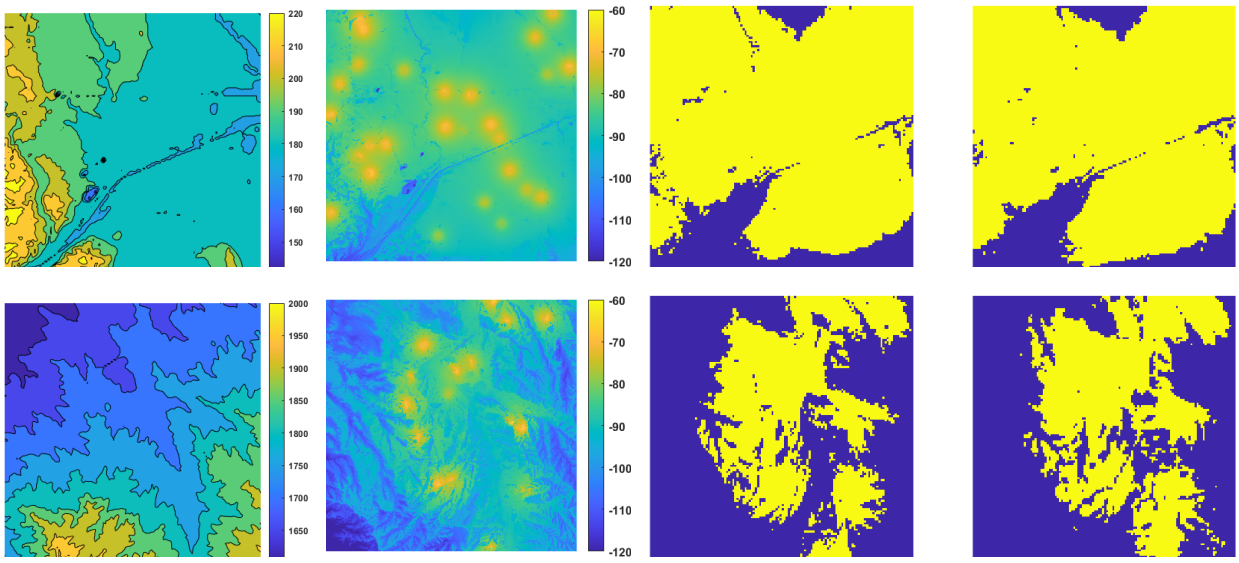

Fig. 2. Fields, occupancy and decision maps in a two regions with different topographies – Top: Chicago, Bottom: Denver. Left: contours of the altitude profiles. Color bars indicate altitudes in meters; Middle Left: fields (1) with $n_{\mathcal{E}} = 26$ and $n_{\mathcal{E}} = 16$ for Chicago and Denver respectively. Color bars indicate power in dBm; Middle Right: occupancy maps $t(\mathcal{G})$ with $\tau = -90\,\text{dBm}$. The color yellow (blue) is shown at locations where the field is larger (smaller) than τ. Right: Decision maps $\hat{t}(\mathcal{G})$. In these figures, $n_{\mathcal{G}} = 128 \times 128$, and the regions are $655.36\,\text{km}^2$. Note that occupancy has very complex and irregular shapes in rough and elevated terrain.

where p_l is the (unknown) power of the lth emitter (in Watts), and $f_{\mathbf{e}_l}(\mathbf{r}) > 0$ is the power measured by a sensor from a unit-power emitter located at $\mathbf{e}_l \in \mathcal{E}$. The forward model $f_{\mathbf{e}_l}(\mathbf{r})$ from any $\mathbf{e}_l$ to $\mathbf{r}$ is used to train the neural network about the propagation characteristics of the region where it is desired to determine occupancy. The model is used only during training; once trained, the network can determine the occupancy map for the trained region using only the sensor measurements, without further information about the forward model or emitter number or locations.

We define the sub-regions $\mathcal{G} = \{\mathcal{G}_k\}_{k=1}^{n_{\mathcal{G}}}$ where $\mathcal{G}_k \subset \mathcal{R}$ as a collection of non-overlapping contiguous units that cover $\mathcal{R}$, where $n_{\mathcal{G}}$ is the number of sub-regions. In this work, $n_{\mathcal{G}}$ is chosen such that the variation of the field within a sub-region $\mathcal{G}_k$ is less than 1 dB on average, and we consider an aggressively small number of sensors, i.e. $n_{\mathcal{S}} \ll n_{\mathcal{G}}$.

We define the occupancy in a sub-region $t(\mathcal{G}_k)$ as the indicator function for whether the field, averaged over $\mathcal{G}_k$, exceeds a predetermined occupancy threshold τ, or

$$t(\mathcal{G}_k) = \mathbb{1}\{\langle f \rangle_{\mathcal{G}_k} \geq \tau\}, \tag{2}$$

where $\langle f \rangle_{\mathcal{G}_k}$ is the mean field over $\mathcal{G}_k$, computed over all $\mathbf{r} \in \mathcal{G}_k$, and $\mathbb{1}\{\cdot\}$ is an indicator function that is one if its argument is true and zero otherwise. Thus, if the average $f(\mathbf{r})$ exceeds τ in $\mathcal{G}_k$, then $t(\mathcal{G}_k) = 1$. Otherwise, it is zero. Typically, $\mathcal{G}$ rasterizes $\mathcal{R}$ along a regular grid and the occupancy maps are images of fixed resolution. The occupancy map is then defined as $t(\mathcal{G}) \equiv \{t(\mathcal{G}_k)\}_{k=1}^{n_{\mathcal{G}}}$.

The system is trained by minimizing a loss function between the decision maps $\hat{t}(\mathcal{G})$ and the occupancy maps $t(\mathcal{G})$. Central to this process is the ability of a system to accommodate an arbitrary number of sensors. We assume that sensors measure power over multiple transmissions such that the effects of fading are averaged out, and fading is therefore not explicitly modeled. We also assume that the sensors are all tuned to the same center frequency and bandwidth of interest.

We utilize Forsk's Atoll [2] to obtain ground truth occupancy maps in an outdoor setting. Atoll accounts for the region topography, antenna patterns, heights, and tilts, emitter powers and locations, and allows field simulations over a large region. The process of generating and storing field and occupancy examples is described in Section II. Topography, occupancy map examples and their corresponding decision maps in the vicinity of two considered cities are shown in Fig. 2.

B. Related Work

ML-based methods make decisions about occupancy without explicitly solving for emitter locations and pow-

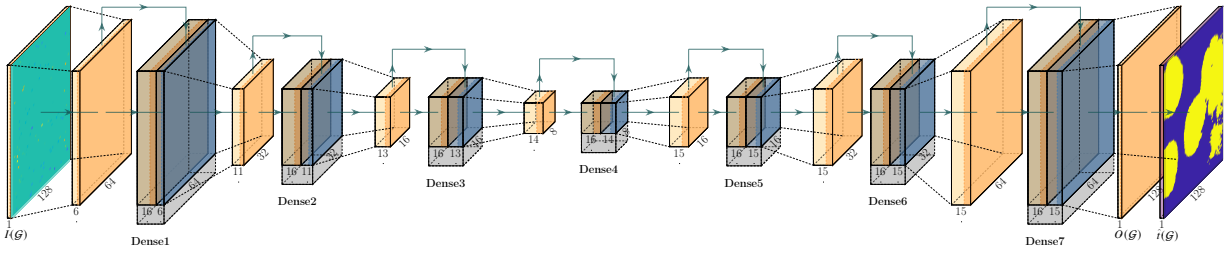

Fig. 3. Encoder-Decoder architecture. The blocks represent feature maps and their dimensions are indicated. The output of a dense block is the concatenation of two feature maps: the output of a convolution block (yellow), and the output of an identity map (blue). The output $O(\mathcal{G})$ passes through a Sigmoid layer and a threshold detector to produce the decisions $\hat{t}(\mathcal{G})$.

ers. An early example [3] uses energy measurements of a single sensor from a single emitter, to analyze the accuracy of several machine learning algorithms; similarly [4] uses multiple sensors and emitters with known $n_{\mathcal{E}}$ in a distance-based pathloss forward model. Both works conclude that support vector machines (SVMs) can be useful to provide decisions at the sensor locations.

Recent works have turned toward neural networks for spectrum occupancy. A measurement campaign for detecting a single 3.5GHz radar band emitter with a single sensor [5] concludes that deep learning methods, in particular convolutional neural networks (CNNs), show an ROC performance superior to classical methods such as energy detection and ML-based methods such as SVMs. Neural networks have also been used to extract features for compressive sensing methods [6] and SVMs [7]. The above methods do not provide decision maps, even when multiple sensors are considered.

More recently, neural networks paired with matrix factorization have been used in spectrum cartography to estimate $f_{\mathbf{e}_l}(\mathbf{s})$ and p_l for every emitter $\mathbf{e}_l$ and sensor location $\mathbf{s}$ [8]. An autoencoder is then used to complete $f_{\mathbf{e}}(\mathbf{s})$ over the whole region. The field is then constructed as in (1), but an estimate of $n_{\mathcal{E}}$ is required by the algorithm. Deep completion autoencoders [9] and generative adversarial networks [10] are also considered to estimate the field $f(\mathbf{r})$, but the experimental data is confined to a small number of emitters $n_{\mathcal{E}} \leq 2$, and in some cases, to idealized forward models. Similar constraints are seen in Bayesian approaches for estimating the field [11].

Our set-up differs from these prior efforts because, rather than making a cartographic map from which decisions can be derived, we make a decision map directly. We allow an unlimited number emitters during training and testing, and do not attempt to explicitly estimate the number of emitters or their locations or powers. We also allow for a variable number of sensors, and do not assume any idealized forward model.

We use a convolutional neural network (CNN) with an encoder-decoder structure, and show that it achieves excellent performance and robustness. The network allows for an arbitrary number of emitters, and sensors, which can vary between training and testing.

C. Problem Statement and Contributions

Given a set of sensor locations $\mathcal{S} \subset \mathcal{R}$, and measurements $m(\mathbf{s})$, $\mathbf{s} \in \mathcal{S}$, we seek occupancy decisions $\hat{t}(\mathcal{G}_k)$ over all $\mathcal{G}_k \in \mathcal{G}$ covering $\mathcal{R}$.

We present a neural network-based decision system with the following properties: (1) It permits any number or distribution of sensors and emitters; (2) It produces a decision map over $\mathcal{G}$, whether or not a sensor occupies every $\mathcal{G}_k$; (3) It is robust to the number of sensors and occupancy threshold; (4) It makes no assumptions on the field parameters, region topography, or emitter properties. During deployment, all sensors report their measurements and locations, and the neural network produces a decision map. A sequence of measurements produces a sequence of decision maps, but temporal prediction and modeling are not considered.

II. Decision System and Performance

Occupancy maps over a region are well-represented by images as shown in Fig. 2. Such images present patterns that encapsulate information about the propagation environment, and various emitter properties such as location, power and number. In this section, we present a CNN with an encoder-decoder structure, and show how it is used as part of a decision system to produce decision maps. In particular, we describe the aggregation of sensor measurements as soft decisions into images of fixed size suitable as inputs to a neural network.

A. Inputs Aggregation and Neural Network

The sensors measure power $m(\mathbf{s})$ at locations $\mathbf{s} \in \mathcal{S}$. Since sensors cannot be everywhere in $\mathcal{R}$, a systematic way to represent the power level at locations with no sensors, with respect to an occupancy threshold τ, is required. A measurement made at $\mathbf{s} \in \mathcal{G}_k$ can be modeled locally, within a sub-region $\mathcal{G}_k$, as

$$m(\mathbf{s}) - \langle f \rangle_{\mathcal{G}_k} = z(\mathbf{s}) \tag{3}$$

where $z(\mathbf{s}) \sim \mathcal{N}(0, \zeta^2)$ is the error in representing the mean field $\langle f \rangle_{\mathcal{G}_k}$ by a single measurement $m(\mathbf{s})$ in a sub-region $\mathcal{G}_k$. The model (3) is only local and the error is generally smaller the smaller the sub-regions. The variance ζ^2 does not need to be known. The measurements $m(\mathbf{s})$ in (3) allow the sensors at to produce log-likelihood ratios (LLRs) for the hypotheses $\mathrm{H}_0 : t(\mathcal{G}_k) = 0$ versus $\mathrm{H}_1 : t(\mathcal{G}_k) = 1$. Since $\langle f \rangle_{\mathcal{G}_k}$ is unknown, this is achieved using a generalized likelihood-ratio test resulting in

$$LLR(\mathbf{s}) = \frac{(m(\mathbf{s}) - \tau)|m(\mathbf{s}) - \tau|}{\zeta^2} \tag{4}$$

Based on (4), the LLR depends on the measurements only through the difference $m(\mathbf{s}) - \tau$, and therefore sensors make the soft decisions according to

$$i(\mathbf{s}) = m(\mathbf{s}) - \tau. \tag{5}$$

This is an intuitively pleasing metric representing the difference between the measured sensor value and the threshold. The larger the metric, the more sure we are that the measurement exceeds the threshold, and conversely. Where there are no sensors, we set $i(\mathbf{r}) = 0$ for $\mathbf{r} \in \mathcal{R} \backslash \mathcal{S}$, representing no information of whether the field is above or below the threshold.

We then form

$$I(\mathcal{G}_k) = \frac{1}{Z} \frac{1}{|\mathcal{G}_k \cap \mathcal{S}|} \sum_{\mathbf{r} \in \mathcal{G}_k} i(\mathbf{r}), \tag{6}$$

which pools the measurement of all the sensors in $\mathcal{G}_k$, where $|\mathcal{G}_k \cap \mathcal{S}|$ is the number of sensors contained in $\mathcal{G}_k$. Since the values of $i(\mathbf{s})$ are very small (on the order of 10^{-11}), we introduce a normalization constant Z that scales the values to be in a range typical for the input to a neural network. For example, choosing Z such that the image $\{I(\mathcal{G}_k)\}_{k=1}^{n_\mathcal{G}}$ has unit variance works well in practice. Thus, for $\mathcal{G}_k$ with no sensors, $I(\mathcal{G}_k) = 0$.

Because we typically consider a square region, this aggregation process can be thought of as forming an image of size $\sqrt{n_\mathcal{G}} \times \sqrt{n_\mathcal{G}}$. We are interested in constructing a binary decision map over all the sub-regions, including those for which $I(\mathcal{G}_k) = 0$. This resembles the problem of image inpainting [12], where incomplete images are recovered from available pixels. CNNs commonly used in these problems employ an encoder-decoder structure, which allows the translation of one type of (input) image, to another type of (output) image.

We use a CNN with an encoder-decoder structure to produce decision maps $\hat{t}(\mathcal{G})$ from $I(\mathcal{G})$. The network structure is shown in Fig. 3. The number of channels and feature map dimensions are indicated in the figure. The input convolution layer kernels are 21×21, and the output layer kernels are 5×5. The network incorporates single layer dense blocks with a growth rate of 16, which concatenate feature maps from previous layers [13]. There are seven dense blocks: three for the encoder, and four for the decoder. Transition layers succeeding every dense block use 3×3 kernels with a stride of 2, followed by 1×1 kernels to pool the channels. Transition layers halve the number of channels. Transition layers perform down-sampling (convolution) in the encoder, and up-sampling (convolution transpose) in the decoder. Batch normalization layers are used before convolutions and ReLUs are chosen as activations. All other kernels are 3×3. The total number of parameters is 29908.

B. Dataset

The emitters number, locations and powers are unknown, and the training system has to generate examples that realize many values of these unknowns. To generate a single occupancy map, the number of emitters $n_\mathcal{E}$ is drawn according to a uniform distribution in the interval $[1, 40]$. Afterwards, a set of emitters locations $\mathcal{E}$ is sampled randomly over the region $\mathcal{R}$. The emitter powers are then uniformly sampled from the interval $[0, 2]$ Watts. The field, over $\mathcal{R}$ is then obtained by (1), and the occupancy map with respect to the occupancy threshold τ is generated according to (2) over $\mathcal{G}$. The datasets are generated per region $\mathcal{R}$, with fixed occupancy threshold τ and number of sensors $n_\mathcal{S}$, with 20480 training examples and 1024 testing examples. The distribution of the sensor locations $\mathcal{S}$ is uniform over $\mathcal{R}$.

The range on $n_\mathcal{E}$ used in the dataset is limited due to diminishing returns of arbitrary large $n_\mathcal{E}$ on the occupancy maps. In our simulations, we find that 40 emitters are enough to cover more than more than 92% of the region in Chicago at $\tau = -90\,\mathrm{dBm}$. Any larger values of $n_\mathcal{E}$ would not generate any richness in the occupancy maps, and therefore during deployment, the larger values of $n_\mathcal{E}$ are automatically accounted for. Moreover, the locations of the emitters used for training are generated independently from those used for testing. This results in a decision system that can be used to produce decision

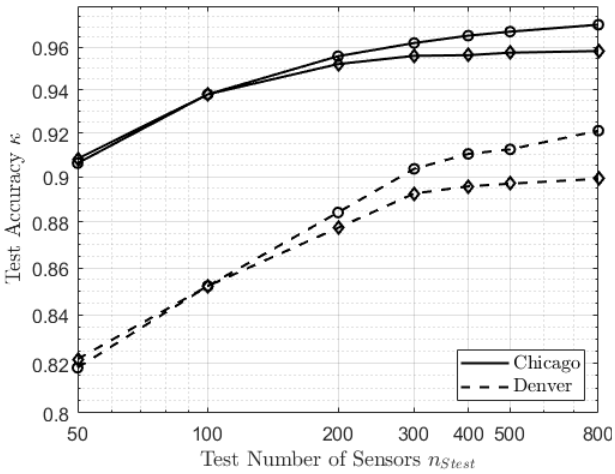

Fig. 4. Robustness to $n_{\mathcal{S}}$. (○) Training and testing are matched: $n_{\mathcal{S}train} = n_{\mathcal{S}test}$; (⋄) Training and testing are mismatched: $n_{\mathcal{S}train} = 100$, $n_{\mathcal{S}test} \in [50, 800]$. Curves (⋄) closely track those of (○) showing robustness to a wide range of $n_{\mathcal{S}}$.

maps for any $n_{\mathcal{E}}$, located anywhere, not just the values it was trained for. All emitters use 11dBi omni-directional 0 Tilt antennas at 2100 MHz mounted 20m above ground level. The bandwidth considered is 10 MHz.

C. Training

The region $\mathcal{R}$, occupancy threshold τ, and $n_{\mathcal{S}}$ are fixed during training, and the number of emitters $n_{\mathcal{E}}$ is allowed to vary. In Section III, we show that the robustness to $n_{\mathcal{E}}$ extends to $n_{\mathcal{S}}$ and τ. The neural network minimizes the pixel-wise cross-entropy loss defined as

$$\mathcal{L} = \frac{1}{M n_{\mathcal{G}}} \sum_{j=1}^{M} \sum_{k=1}^{n_{\mathcal{G}}} \Big(- \; t_j(\mathcal{G}_k) \log \big[\sigma(O_j(\mathcal{G}_k))\big] - (1 - t_j(\mathcal{G}_k)) \log \big[1 - \sigma(O_j(\mathcal{G}_k))\big] \Big). \quad (7)$$

where M is the number of training or testing examples, depending on whether the loss is computed during training or testing, and $O_j(\cdot)$ is the neural network output shown in Fig. 3. The training mini-batch size is set to 32, and the learning rate is 5×10^{-5}; reduced by a factor of 10 if the loss plateaus over a period of 10 epochs. We train the neural network for 500 epochs to make sure that the training is stable. The performance of the decision system is determined by the accuracy, denoted by κ, of classifying the sub-regions $\mathcal{G}_k \in \mathcal{G}$ averaged over sub-regions and emitter locations, powers, and numbers, and sensor locations drawn randomly.

As expected from the region topography, Chicago occupancy maps are easier to learn resulting in 93.8% test accuracy. On the other hand, the occupancy maps in the Denver area are highly irregular, resulting in 85.2% test accuracy. Training and testing are performed on a single NVIDIA GeForce GTX Titan X GPU. During deployment, the time to produce a decision map $\hat{t}(\mathcal{G})$ from an input $I(\mathcal{G})$ is approximately 8 ms. This value is smaller than typical delays of wireless networks and thus decisions can be generated in real-time as soon as the measurements are available.

III. Robustness: Number of sensors $n_{\mathcal{S}}$ and Occupancy Threshold τ

The usefulness of any neural network-based decision system is determined in part by its robustness to parameter values for which it has not been trained. Until now, we have matched the training and test environments as we averaged over emitter and sensor realizations. In this section, we examine the test accuracy of the decision system when (1) the number of sensors $n_{\mathcal{S}}$ and (2) the occupancy threshold τ have values that differ during testing and training. We show that the decision system is robust to a wide range of $n_{\mathcal{S}}$ and τ; we can train with a single $n_{\mathcal{S}}$ or τ, and still do well when testing with various $n_{\mathcal{S}}$ and τ.

A. Number of sensors $n_{\mathcal{S}}$

The decision system may have limited control over the number of sensors $n_{\mathcal{S}}$ and their locations, and the number of measurements aggregated into $I(\mathcal{G})$ during training and testing could differ. As described in Section II, the sensor locations are varied during the training so that there are no location biases. However, we postulate that, the test accuracy of a system trained for $n_{\mathcal{S}train}$ will increase with $n_{\mathcal{S}test} > n_{\mathcal{S}train}$ because more soft decisions are aggregated per sub-region. Following the same process described in Sections II, we generate a test set for values of $n_{\mathcal{S}test} \in [50, 800]$. Due to the aggregation step (6), the size of the input $I(\mathcal{G})$ remains fixed at 128×128, even as $n_{\mathcal{S}test}$ is changed.

The test accuracy as $n_{\mathcal{S}test}$ varies, with $n_{\mathcal{S}train} = n_{\mathcal{S}test}$ are shown in Fig. 4. The monotonic increase in accuracy is expected as more sensors are aggregated in a sub-region. To verify the robustness, we plot the test accuracy as $n_{\mathcal{S}test}$ varies, but $n_{\mathcal{S}train} = 100$. One immediately observes that the decision system's test accuracy is not maximized for $n_{\mathcal{S}test} = 100$. Quite remarkably, the test accuracy improves with $n_{\mathcal{S}test}$ without retraining, and robustness is observed by comparing the curves with ⋄ and ○ markers: the test accuracy of a decision system with fixed $n_{\mathcal{S}train} = 100$ incurs a maximum

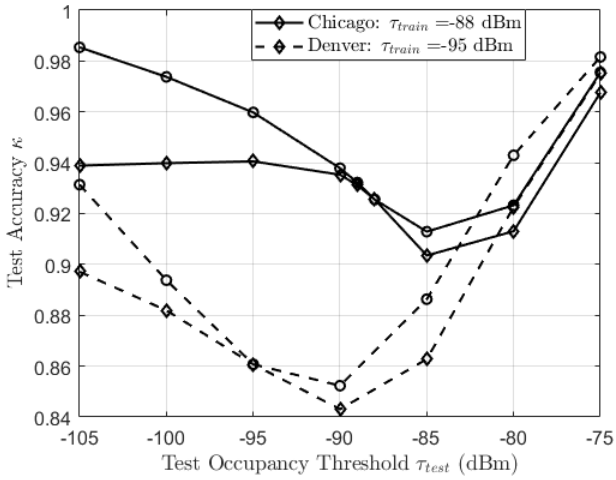

Fig. 5. Robustness to τ. ($\circ$) Training and testing are matched: $\tau_{train} = \tau_{test} \in [-105, -75]$ dBm; ($\diamond$) Training and testing are mismatched: fixed τ_{train}, but $\tau_{test} \in [-105, -75]$ dBm. Curves ($\diamond$) track those of ($\circ$) showing robustness to a wide range of τ_{test}.

difference of approximately 2% relative to that with matched $n_{\mathcal{S}train} = n_{\mathcal{S}test}$.

B. Occupancy Threshold τ

The decision maps $\hat{t}(\mathcal{G})$ produced by the decision system are always referenced to an occupancy threshold τ. We show that a system trained for a threshold τ_{train} can be used effectively for testing with $\tau_{test} \neq \tau_{train}$. In this case, the sensor soft decisions are $i(\mathbf{s}) = m(\mathbf{s}) - \tau_{train}$ during training, and $i(\mathbf{s}) = m(\mathbf{s}) - \tau_{test}$ during testing.

The test accuracy of the decision systems as τ_{test} varies, with $\tau_{train} = \tau_{test}$ are shown in Fig. 5 with ($\circ$) markers, whereas ($\diamond$) markers show the accuracy as τ_{test} varies, but where $\tau_{train} = -88$ dBm in Chicago, $\tau_{train} = -95$ dBm in Denver. The decrease in accuracy is limited to four percent relative to that of $\tau_{train} = \tau_{test}$, showing robustness to the τ_{test}.

The robustness to τ allows the decision system to predict occupancy with respect to an arbitrary τ_{test} from a single trained system. Since $m(\mathbf{s}) - \tau_{test} = m(\mathbf{s}) - \tau_{train} + (\tau_{train} - \tau_{test})$, the difference $\tau_{train} - \tau_{test}$ represents a positive (negative) offset to the measurement, adjusting for the mismatch in τ. This robustness allows us to contemplate generating contours of constant $f(\mathbf{r})$ for various τ, but this is left as future work.

IV. Concluding Remarks and Future Work

We presented a CNN-based decision system for producing accurate decision maps of spectrum occupancy. Minimal assumptions were made on the number, location, or power of the emitters. We demonstrated a simple LLR-based sensor aggregation step to transform the variable number of measurements into images of fixed size. We showed that the system is robust to the number and locations of emitters and sensors, and the threshold τ. Future work includes studying the effects of sensor impairments such as noise, and the effects of fading. Finally, we note that the presented approach is not limited to spatial occupancy, but can be used as a general framework to make binary decisions maps on any field quantity.

References

[1] N. Kleber, A. Termos, G. Martinez, J. Merritt, B. Hochwald, J. Chisum, A. Striegel, and J. N. Laneman, "RadioHound: A pervasive sensing platform for sub-6 GHz dynamic spectrum monitoring," in *Proc. IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2017, pp. 1–2.

[2] Atoll Radio Planning Software. [Online]. Available: http://www.forsk.com/atoll/

[3] F. Azmat, Y. Chen, and N. Stocks, "Analysis of spectrum occupancy using machine learning algorithms," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 6853–6860, 2015.

[4] K. M. Thilina, K. W. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209–2221, 2013.

[5] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, "Deep learning classification of 3.5-GHz band spectrograms with applications to spectrum sensing," *IEEE transactions on cognitive communications and networking*, vol. 5, no. 2, pp. 224–236, 2019.

[6] M. Nazzal, A. R. Ektí, A. Görçin, and H. Arslan, "Exploiting sparsity recovery for compressive spectrum sensing: A machine learning approach," *IEEE Access*, vol. 7, pp. 126 098–126 110, 2019.

[7] J. Kim and J. P. Choi, "Sensing coverage-based cooperative spectrum detection in cognitive radio networks," *IEEE Sensors Journal*, vol. 19, no. 13, pp. 5325–5332, 2019.

[8] S. Shrestha, X. Fu, and M. Hong, "Deep generative model learning for blind spectrum cartography with nmf-based radio map disaggregation," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4920–4924.

[9] Y. Teganya and D. Romero, "Data-driven spectrum cartography via deep completion autoencoders," in *Proc. IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[10] X. Han, L. Xue, F. Shao, and Y. Xu, "A power spectrum maps estimation algorithm based on generative adversarial networks for underlay cognitive radio networks," *Sensors*, vol. 20, no. 1, p. 311, 2020.

[11] Y.-Q. Xu, B. Zhang, G. Ding, B. Zhao, S. Li, and D. Guo, "Radio environment map construction based on spatial statistics and Bayesian hierarchical model," *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[12] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4700–4708.