Jointly Optimizing Placement and Inference for Beacon-based Localization

Charles Schaff David Yunis Ayan Chakrabarti Matthew R. Walter

***Abstract*— The ability of robots to estimate their location is crucial for a wide variety of autonomous operations. In settings where GPS is unavailable, range- or bearing-only observations relative to a set of fixed beacons provide an effective means of estimating a robot's location as it navigates. The accuracy of such a beacon-based localization system depends both on how beacons are spatially distributed in the environment, and how the robot's location is inferred based on noisy measurements of range or bearing. However, it is computationally challenging to search for a placement and an inference strategy that, together, are optimal. Existing methods decouple these decisions, forgoing optimality for tractability. We propose a new optimization approach to jointly determine the beacon placement and inference algorithm. We model inference as a neural network and incorporate beacon placement as a differentiable neural layer. This formulation allows us to optimize placement and inference by jointly training the inference network and beacon layer. We evaluate our method on different localization problems and demonstrate performance that exceeds hand-crafted baselines.**

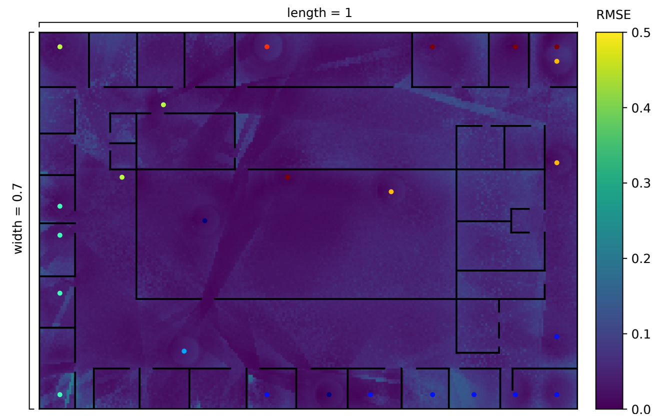

Fig. 1. Our algorithm optimizes beacon allocation jointly with receiver localization (inference). The image depicts the optimized placement and channel assignments (colored circles) along with a map of the RMSE for the environment shown.

I. Introduction

Placing sensors and beacons in a physical environment can assist robots and autonomous systems in understanding their surroundings. However, measurements from these devices provide only indirect or noisy information towards physical properties of interest, and require additional computational processing for inference. Such inference must take into account sources of degradation in the measurement process, as well as any other prior statistical knowledge of the environment. At the same time, the degree to which this inference succeeds is limited by how the beacons and sensors are distributed physically in the first place.

Consider location-awareness, which is critical to enabling navigation by robots and humans, resource discovery, asset tracking, logistical operations, and resource allocation [1]. In situations for which GPS is unavailable (indoors, underground, or underwater) or impractical, measurements (e.g., range and bearing) relative to fixed beacons [2–11] provide an attractive alternative. Designing a system for beacon-based location-awareness requires simultaneously making the following design decisions: (a) how the beacons should be distributed (e.g., spatially and possibly across transmission channels); and (b) how location should be determined based on measurements from these beacons.

Note that these decisions are inherently coupled. The placement of beacons and their channel allocation influence the nature of the ambiguity in measurements at different locations, and therefore which inference strategy is optimal. One must therefore search over the space of both beacon placements (and possibly transmission channels) and inference strategies to find a pair that is jointly optimal. Unfortunately, due to phenomena such as noise, interference, and attenuation due to obstructions (e.g., walls), this search rarely has a closed-form solution in all but the most simplistic of settings.

Consequently, most existing techniques decouple these decisions and rely upon approximations to make the search tractable. Beacon placement is typically performed either by relying on expert system designers to identify "good" locations, by using heuristics (e.g., coverage), or according to the functionality of a particular inference method. For example, a simple approach is to assume that transmitters have a fixed sensing radius and to formulate placement as a solution to the art-gallery problem [12]. This formulation relies upon simplistic assumptions regarding the sensing geometry, and does not account for the effects of the environment (e.g., signal attenuation due to walls), noise, or interference.

Recently, Chakrabarti [13] introduced a method that successfully uses stochastic gradient descent (SGD) to jointly learn sensor multiplexing patterns and reconstruction methods in the context of imaging. Motivated by this, we propose a new learning-based approach to designing the beacon distribution and inference algorithm jointly for the task of localization. We instantiate the inference method as a neural network, and encode the beacon distribution as a differ-

C. Schaff, A. Chakrabarti, and M.R. Walter are with the Toyota Technological Institute at Chicago, Chicago, IL USA {cbschaff, ayanc, mwalter}@ttic.edu.

D. Yunis is with the University of Chicago, Chicago, IL USA dyunis@uchicago.edu.

entiable neural layer. We show that the inference network and beacon layer can be jointly trained in combination with a general signal propagation model for an arbitrary environment, to automatically discover an optimized design for a location-awareness system in that environment.

We show that these designs yield localization accuracies that exceed hand-crafted approaches to beacon placement, and can be adjusted to trade-off location accuracy and the number of beacons. Our approach is general, and can be applied to produce localization systems without expert intervention for arbitrary environments and signal types.

II. Related Work

A great deal of attention has been paid within the robotics community and beyond to the problem of estimating location and other spatial phenomena using sensor networks. Existing work typically focuses either on the problem of inference for a given network or the task of choosing an optimal network allocation for a given inference strategy. While the following discussion focuses on localization within a network of beacons, many of the concepts apply to the more general problem of estimating spatial phenomena using sensor networks.

A common approach to localization within existing radio frequency (RF)-based beacon networks is to use the Received Signal Strength (RSS) as a fingerprint that is matched against a database of RSS-location pairs in order to determine the receiver's location [14]. This database is typically generated via a manually performed site-survey, though there has been work to generate this map automatically or "organically" during operation [15, 16]. In similar fashion to our inference model, Sala et al. [17] use a neural network (a multilayer perceptron) to predict a receiver's location within an existing network based upon Received Signal Strength. Altini et al. [18] take a similar approach for networks that employ Bluetooth for communication.

Many beacon networks provide direct, albeit noisy, measurements of inter-node range or bearing. For example, acoustic long baseline (LBL) networks are frequently used to localize underwater vehicles [3, 4], while a number of low-cost systems exist that use RF and ultrasound to measure range [19, 20]. Moore et al. [21] propose an algorithm for estimating location based upon noise-corrupted range measurements, formulating the problem as one of realizing a two-dimensional graph whose structure is consistent with the observed ranges. Detweiler et al. [7] propose a geometric technique that estimates a robot's location as it navigates a network of fixed beacons using either range or bearing observations. Kennedy et al. [10] employ spectral methods to localize camera networks using angular measurements, without the need for a global coordinate frame. Alternatively, Shareef et al. [22] evaluate the use of feedforward and recurrent neural network architectures to localize a receiver based upon noisy range measurements.

Meanwhile, beacon allocation traditionally relies on coverage as a heuristic to guide the placement of beacons in a particular environment. When beacons are assumed to have a fixed transmission radius, the art-gallery problem [12] provides a reasonable formulation of the coverage problem. However, a beacons's field-of-view is typically not fixed and is instead a function of the layout and composition of the environment, beacon noise, and signal interference, which itself varies according to the placement of other beacons. Agarwal et al. [23] propose a greedy landmark-based method that solves for the placement of minimum number of beacons (within a $\log$ factor) necessary to cover all but a fraction of a given polygonal environment.

When the inferred phenomena is modeled using a Gaussian process (GP), a common technique is to greedily introduce sensors at the location with the highest entropy under the GP [24, 25]. However, this approach does not model the accuracy of the predictions at the selected locations. Alternatively, Krause et al. [26] consider the problem of choosing the location for a fixed sensor network that maximizes mutual information using a Gaussian process model for phenomena over which inference is performed (e.g., temperature). They show that this problem is NP-complete and propose a polynomial-time approximation algorithm that exploits the submodularity of mutual information to provide placements that are within a constant-factor of the optimal locations. Similarly, Cameron and Durrant-Whyte [27] use mutual information in order to determine sensor placement for localization and recognition tasks. Meanwhile, Fang and Lin [28] consider localization accuracy when placing wireless access points and choose locations that maximize signal-to-noise ratio. In similar fashion to our proposal to learn sensor placement via backpropagation, Kang et al. [29] propose using a neural network to determine the placement of WiFi access points.

In many scenarios, beacon allocation includes both choosing where to place beacon in a given environment as well as which channel to assign to each beacon. These problems are typically decoupled—the location of each beacon is first determined and then their channels are chosen in order to minimize interference [30] This can result in a sub-optimal allocation. Ling and Yeung [31] jointly solve access point placement and channel selection using local search.

Meanwhile, the sensor selection problem [32, 33] considers related scenarios in which there is a cost associated to querying sensors. The problem is to choose the subset of sensors to utilize at each point in time so as to balance inference accuracy with the corresponding query cost. Also related is the task of choosing where to place cellular towers, which is driven by multiple objectives including coverage, signal-to-noise ratio, and cost and often involves extensive manual site surveys, high-fidelity simulations, and hand-designed placement strategies [34].

III. Approach

We formalize the design problem for a location-awareness system as that of determining an optimal distribution of beacons $\mathcal{D}$ and an inference function $f(\cdot)$, given an environment $\mathcal{E}$. For a given set of L possible locations for beacons, we parameterize the distribution $\mathcal{D}$ as an assignment I_l to each

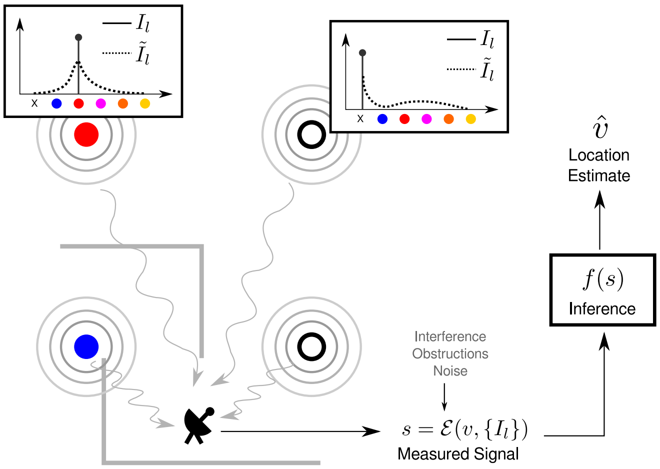

Fig. 2. Proposed approach. We seek to find the distribution of beacons $\{I_l\}$ and inference function $f(\cdot)$ that together yield reliable location estimates for a specific environment $\mathcal{E}$. For a candidate set of locations l, the assignment vectors I_l determine whether a beacon is to be placed at that location, and if so, in what configuration. The environment mapping $\mathcal{E}$ then determines what the measured signal will be at any given location, taking into account non-idealities such as noise, interference, etc. We optimize these assignment vectors jointly with parameters of the inference function $f(\cdot)$, to minimize error in the estimated locations $\hat{v}$.

location $l \in \{1, \ldots L\}$, where $I_l \in \{0,1\}^{C+1}, |I_l| = 1$ is a $(C+1)$-dimension 0-1 vector with all but one entry equal to 0. This vector denotes whether to place a beacon at location l (otherwise $I_l^1 = 1$), and if so, in which one of C possible configurations (i.e., $I_l^{c+1} = 1$ for configuration c).

We parameterize the environment in terms of a function $s = \mathcal{E}(v, \{I_l\})$, which given a location $v \in \mathbb{R}^2$ and a distribution of beacons $\mathcal{D} = \{I_l\}$, produces a vector $s \in \mathbb{R}^m$ of measurements that an agent is likely to make at that location. Note that the environment $\mathcal{E}$ need not be a deterministic function. In case of probabilistic phenomena such as noise and interference, $\mathcal{E}$ will produce a sample from the distribution of possible measurements. The inference function $f(\cdot)$ is then tasked with computing a reliable estimate of the location, given these measurements.

Our goal is to jointly optimize $\mathcal{D} = \{I_l\}$ and $f(\cdot)$ to minimize the average value of the Euclidean distance between the estimated and true locations, over a distribution of possible locations where the agent may visit. Additionally, we add a regularizer to our objective, e.g., to minimize the total number of individual beacons. Figure 2 provides a schematic overview of our problem formulation.

A. Optimization with Gradient Descent

Unfortunately, the problem as stated above involves a combinatorial search, since the space of possible beacon distributions is discrete with $(C+1)^L$ elements. We make the optimization tractable by adopting an approach similar to that of Chakrabarti [13]. We relax the assignment vectors I_l to be real-valued and positive as $\tilde{I}_l \in \mathbb{R}_+^{C+1}, |\tilde{I}_l| = 1$. $\tilde{I}_l$ can be interpreted as a probability distribution over the possible assignments at location l.

Instead of optimizing over distribution vectors $\tilde{I}_l$ directly, we learn a weight vector $w_l \in \mathbb{R}^{C+1}$ with

$$\tilde{I}_l = \text{SoftMax}(\alpha w_l), \quad \tilde{I}_l^c = \frac{\exp(\alpha w_l^c)}{\sum_{c'} \exp(\alpha w_l^{c'})}, \tag{1}$$

where α is a positive scalar parameter. Since our goal is to arrive at values of $\tilde{I}_l$ that correspond to hard assignments, we begin with a small value of α and increase it during the course of optimization according to an annealing schedule. Small values of α in initial iterations allow gradients to be back-propagated across Eqn. 1 to update $\{w_l\}$. As optimization progresses, increasing α causes the distributions $\{I_l\}$ to get "peakier", until they converge to hard assignments.

We also define a distributional version of the environment mapping $\tilde{\mathcal{E}}(v, \{\tilde{I}_l\})$ that operates on these distributions instead of hard-assignments. This mapping can be interpreted as producing the *expectation* of the signal vector s at location v, where the expectation is taken over the distributions $\{\tilde{I}_l\}$. We require that this mapping be differentiable with respect to the distribution vectors $\{\tilde{I}_l\}$. In the next sub-section, we will describe an example of an environment mapping, and its distributional version that satisfies this requirement.

Next, we simply choose the inference function $f(\cdot)$ having some parametric form (e.g., a neural network), and learn its parameters jointly with the weights $\{w_l\}$ of the beacon distribution as the minimizers of a loss function:

$$L(\{w_l\}, \Theta) = R(\{\tilde{I}_l\}) + \frac{1}{\|\mathcal{V}\|} \sum_{v \in \mathcal{V}} \mathop{\mathbb{E}}_{s} \left\| v - f\left(\tilde{\mathcal{E}}\left(v, \tilde{I}_l\}_l\right); \Theta\right) \right\|^2, \tag{2}$$

where $\mathcal{V}$ is the set of possible agent locations, Θ are the parameters of the inference function f, $\tilde{I}_l =$SoftMax(αw_l), as $\alpha \to \infty$, and R is a regularizer. Note that the inner expectation in the second term of Eqn. 2 is with respect to the distribution of possible signal vectors for a fixed location and beacon distribution, and captures the variance in measurements due to noise, interference, etc.

We minimize Eqn. 2 with stochastic gradient descent (SGD) computing gradients over a small batch of locations $v \in \mathcal{V}$, with a single sample of s per location. We find that the quadratic schedule for α used by Chakrabarti [13] works well, i.e., we set $\alpha = \alpha_0(1 + \gamma t^2)$ at iteration t.

B. Application to RF-based Localization

To give a concrete example of an application of this framework, we consider the following candidate setting of localization using RF beacons. We assume that each beacon transmits a sinusoidal signal at one of C frequencies/channels. The amplitude of this signal for every beacon is assumed to be fixed, but we allow different beacons to have arbitrary phase variations amongst them.

We assume an agent at a location has a receiver with multiple band-pass filters, and is able to measure the power in each channel separately (i.e., $m = C$). We assume that the power of each beacon's signal drops as a function of distance, and of the number of obstructions (e.g., walls) in the line-of-sight between the agent and the beacon. The

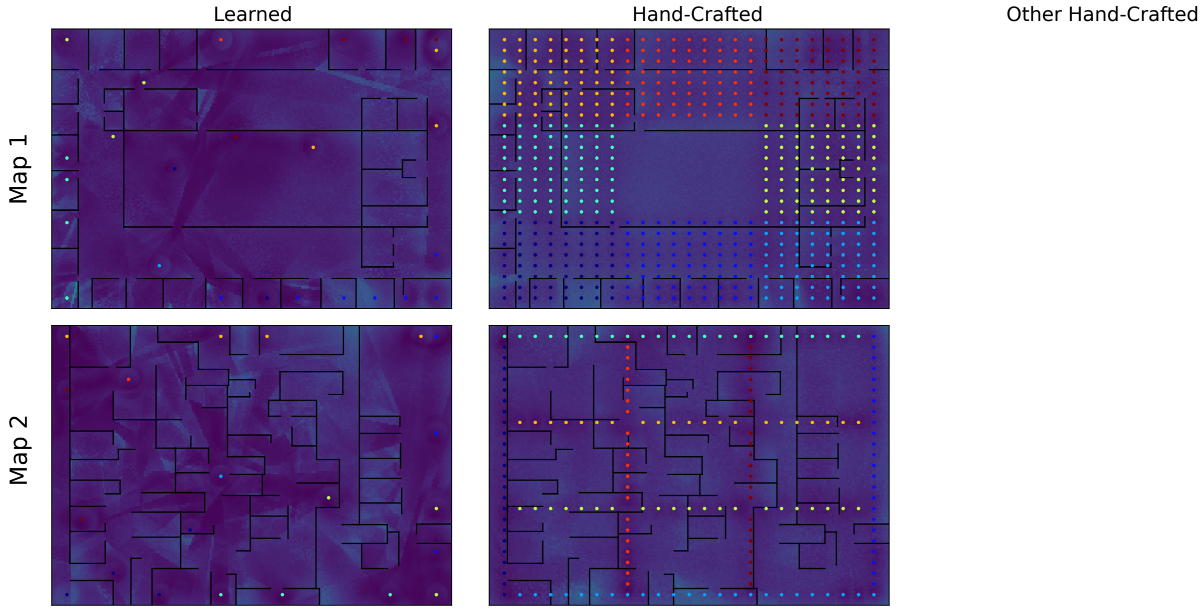

Fig. 3. This figure shows the beacon placement and RMSE heatmap for the best-performing 8 channel learned and hand-crafted beacon placements on Map 1 and Map 2. The figures on the far right show other 8 channel hand-crafted placements that performed worse. Every hand-crafted placement was tried on both maps.

measured power at the receiver in each channel is then based on the amplitude of the super-position of signals from all beacons transmitting in that channel. This super-position is a source of interference, since individual beacons have arbitrary phase. We also assume that there is some measurement noise at the receiver.

We assume all beacons transmit at power P_0, and model the power of the attenuated signal received from beacon l at location v as

$$P_l(v) = P_0 \; r_{l:v}^{-\zeta} \; \beta^{o_{l:v}}, \tag{3}$$

where ζ and β are scalar parameters, $r_{l:v}$ is the distance between v and the beacon location l, and $o_{l:v}$ is the number of walls/obstructions intersecting the line between them. The measured power $s = \mathcal{E}(v, \{I_l\})$ in each channel at the receiver is then modeled as:

$$s^c = \left[\epsilon_1 + \sum_l I_l^{c+1} \sqrt{P_l(v)} \cos\phi_l\right]^2 + \left[\epsilon_2 + \sum_l I_l^{c+1} \sqrt{P_l(v)} \sin\phi_l\right]^2, \tag{4}$$

where ϕ_l is the phase of beacon l, and ϵ_1 and ϵ_2 correspond to sensor noise. We also model sensor saturation by clipping s^c at some threshold τ. At each invocation of the environment function, we randomly sample the phases $\{\phi_l\}$ from a uniform distribution between $[0, 2\pi)$, and noise terms ϵ_1 and ϵ_2 from a zero-mean Gaussian distribution with variance σ_z^2.

For training, the distributional version of the environment function $\tilde{\mathcal{E}}$ is constructed simply by replacing I_l with $\tilde{I}_l$ in Eqn. 4. For regularization, we use a term that penalizes the total number of beacons with a weight λ:

$$R(\{\tilde{I}_l\}) = \lambda \sum_l \tilde{I}_l^1. \tag{5}$$

This setting simulates an environment that is complex enough to not admit closed-form solutions for the inference function or the beacon distribution. Of course, there may be other phenomena in certain applications, such as leakage across channels, multi-path interference, etc., that are not modeled here. However, these too can be incorporated in our framework as long as they can be modeled with an appropriate environment function $\mathcal{E}$.

IV. Results

In this section, we evaluate our method through a series of simulation-based experiments on two different environment maps, learning a neural network for the inference function.

For each map, we consider several manual beacon distributions, and show that the network architecture we choose is able to localize effectively given these distributions, by comparing its results to a simple nearest-neighbors baseline. Then, we show results from learning the beacon distribution and inference network jointly, and show that for both environments, our method learns a more optimal distribution strategy that leads to better localization accuracy.

We end the section by providing the reader with an analysis of the effects of different degrees of regularization, as well as the variation in the learned distributions based on parameters of the environment function $\mathcal{E}$.

A. Experimental Setup

We conduct our experiments on two manually drawn environment maps, which correspond to floor plans (of size

1×0.7 distance units) with walls serving as obstructions. For each map, we arrange $L = 625$ possible beacon locations in a 25×25 evenly spaced grid. We consider configurations with values of $C = 4, 8$, and 16 RF-channels.

Our experiments use the environment model defined in Eqn. 3 with $P_0 = 6.25 \times 10^{-4}$, $\zeta = 2.0$ (where locations are in map units), and $\beta = e^{-1.0}$, with sensor noise variance $\sigma_z^2 = 10^{-4}$. The sensor measurements are saturated at a threshold $\tau = 1.0$

While training the beacons, we use parameters $\alpha_0 = 1$ and for the quadratic temperature scheme $\gamma = 1.25 \times 10^{-9}$. These values were chosen empirically, so that the beacon selection vectors $\tilde{I}_l$ converge at the same pace as it takes the inference network to learn (as observed while training on a fixed beacon distribution). After 9×10^5 iterations, we switch the softmax to an "arg-max", effectively setting α to infinity and fixing the beacon placement, and continue training the inference network.

As mentioned above, the inference function $f(\cdot)$ is parameterized as a 13-layer feed-forward neural network. Our architecture consists of 6 blocks of 2 fully-connected layers. All hidden layers contain 1024 units, and are followed by ReLU activation. Each block is followed by a max pooling operation applied on disjoint sets of four units. After the final block, there is a final output layer with 2 units that predicts the x, y location coordinates.

During training, locations are randomly sampled and fed through our environment model to the inference network in batches of 1000. All networks were trained by minimizing the loss defined in Eqn. 2 for 10^6 iterations using SGD with a learning rate of 0.01 and momentum 0.9. We also use batch-normalization in all hidden layers.

B. Experiments

We evaluate the effectiveness of our inference network, as well as of the joint optimization of beacon placement and localization. To that end, we compare our method with two baselines. First, we compare our inference network with a k-Nearest Neighbor (kNN) baseline using hand-crafted beacon placements. The training data for the kNN experiments was compiled by sampling from our environment model on a dense grid for each map. We ran experiments with $k = 1$, 5, 10, and 20, comparing our model to the best k for each beacon placement. We measure the performance of each model with both root mean squared error (RMSE) and the mean over max errors at each location, which we refer to as *Max Error*. The effect of interference on performance can vary greatly between different beacon placements. Therefore it is useful to measure both average-case performance as well as the worst-case performance, as some placements may be more susceptible to interference.

We compare these results to jointly learned models on both maps. Tables I and II present the results for Map 1 and Map 2, respectively, where we include the best-performing baselines. Figure 3 displays the beacon placement and channel allocation along with the RMSE for both learned and hand-crafted beacon allocations. Figure 5 shows the evolution of beacon placement throughout training for our best performing models. Note that the images depict a hard assignment, however the network is reasoning over high entropy placements early in training, which explains the initial sparsity. With both maps, the network quickly clusters a large number of beacons by channel, and gradually learns to reduce the number of beacons and increase channel diversity, converging to a stable configuration. The results reveal that our inference network can effectively learn to localize the agent given a fixed beacon placement. They also demonstrate that learning this placement yields accuracies that exceed those of hand-crafted placements while using significantly fewer beacons. Note that hand-crafted placements with fewer beacons performed significantly worse.

TABLE I
PERFORMANCE ON MAP 1

Placement	Inference	Channels	Beacons	Mean Error	Max Error
Human	kNN	8	544	0.0801	0.1756
Human	Learned	8	544	0.0731	0.1550
Learned	Learned	8	25	**0.0510**	**0.1180**

TABLE II
PERFORMANCE ON MAP 2

Placement	Inference	Channels	Beacons	Mean Error	Max Error
Human	kNN	8	180	0.0778	0.1883
Human	Learned	8	180	0.0659	0.1568
Learned	Learned	8	21	**0.0512**	**0.1118**

TABLE III
STABILITY WITH REGULARIZATION

	Mean	Std. Dev.	Min	Max
Mean Error	0.0588	0.0021	0.0551	0.0618
Max Error	0.1382	0.0040	0.1312	0.1452
Num. Beacons	93.50	10.60	79	119

In practice, placing beacons will often have a monetary cost. To allow for a trade-off between the number of beacons placed and accuracy, we consider the regularization scheme defined in Eqn. 5. In our experiments, we vary λ between 0.0 and 0.2. As Figure 4 shows, we find that increased regularization leads to solutions with fewer beacons. On Map 2, we find that decreased regularization always leads to solutions with lower error. On Map 1, however, we find that unregularized beacon placements result in increased localization error. This regularization may also allow our model to escape bad local minima during training.

Additionally, we also experiment with an annealing scheme for λ and find that it empirically improves performance. We use a simple annealing schedule that decays λ by a constant factor η every 100,000 iterations. We experiment with initial $\lambda = 0.2$ and $\eta \in \{0.25, 0.5\}$. The best results for each map are shown in Figure 4.

Next, we consider a series of experiments that highlight the robustness of our method to random initializations, different environment models, and varying the number of available

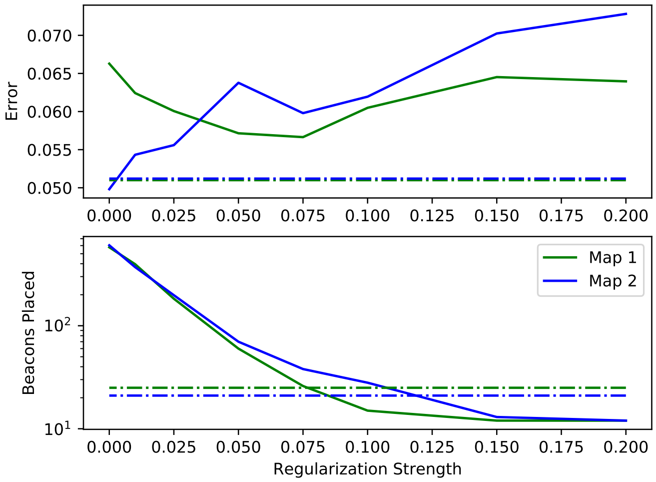

Fig. 4. A plot showing the effects of regularization on mean error and the number of beacons placed. The dashed lines represent the mean error and beacons placed for the best performing annealed regularizer on each map.

TABLE IV
ENVIRONMENT MODEL EXPERIMENTS

Map	Environment Model	Beacons	Mean Error	Max Error
1	Original	12	0.0640	0.1519
1	Low Attenuation	8	0.0429	0.0977
1	High Noise	27	0.1403	0.2910
2	Original	12	0.0728	0.1595
2	Low Attenuation	8	0.0459	0.0975
2	High Noise	28	0.1211	0.2628

TABLE V
CHANNEL EXPERIMENTS

Placement	Inference	Channels	Beacons	Mean Error	Max Error
Human	Learned	4	92	**0.1318**	0.2840
Learned	Learned	4	11	0.1321	**0.2594**
Human	Learned	8	544	0.0731	0.1550
Learned	Learned	8	12	**0.0640**	**0.1519**
Human	Learned	16	134	0.0489	0.1574
Learned	Learned	16	12	**0.0401**	**0.0916**

channels. To test the robustness to random initializations, we repeat an experiment 10 times and report the mean and variance of the mean error in Table III. We then test our method on 2 different environment models, one with decreased attenuation at walls ($\beta = e^{-0.2}$) and one with increased noise ($\sigma_z^2 = 2.5 \times 10^{-4}$). Table IV presents the results. We find that our method adapts to these changes intuitively. Our method places fewer beacons when the signal passes largely unattenuated through walls and places more beacons when combating increased noise. Lastly, we experiment with 4 and 16 channels. Unsurprisingly, increasing the number of channels increases the performance of both handcrafted and learned beacon placements. Results are shown in Table V.

V. CONCLUSION

We described a novel learning-based method capable of jointly optimizing beacon allocation (placement and channel assignment) and inference for localization tasks. Underlying our method is a neural network formulation of inference with an additional differentiable neural layer that encodes the beacon distribution. By jointly training the inference network and beacon layer, we automatically learn the optimal design of a location-awareness system for arbitrary environments. We evaluated our method for the task of RF-based localization and demonstrated localization accuracies that exceed those that can be achieved with hand-crafted allocation strategies. Additionally, we analyzed the performance of our method under different propagation models and presented a strategy for trading off the number of beacons and the achievable accuracy. While we describe our method in the context of range-based localization, the approach generalizes to problems that involve estimating a broader class of spatial phenomena using sensor networks. A reference implementation of our optimization algorithm is available on the project page at `http://www.ttic.edu/chakrabarti/nbp/`.

VI. ACKNOWLEDGEMENTS

We thank the NVIDIA corporation for the donation of Titan X GPUs used in this research.

REFERENCES

[1] S. Teller, J. Chen, and H. Balakrishnan, "Pervasive pose-aware applications and infrastructure," *IEEE Computer Graphics and Applications*, vol. 23, no. 4, pp. 14–18, July–August 2003.

[2] D. Kurth, G. Kantor, and S. Singh, "Experimental results in range-only localization with radio," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, Las Vegas NV, October 2003, pp. 974–979.

[3] P. Newman and J. Leonard, "Pure range-only sub-sea SLAM," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Taipei, Taiwan, September 2003, pp. 1921–1926.

[4] E. Olson, J. Leonard, and S. Teller, "Robust range-only beacon localization," in *Proc. IEEE/OES Autonomous Underwater Vehicles (AUV) Conf.*, June 2004.

[5] J. Djugash, S. Singh, G. Kantor, and W. Zhang, "Range-only SLAM for robots operating cooperatively with sensor networks," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Orlando, FL, May 2006, pp. 2078–2084.

[6] V. Isler, "Placement and distributed deployment of sensor teams for triangulation-based localization," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Orlando, FL, May 2006, pp. 3095–3100.

[7] C. Detweiler, J. Leonard, D. Rus, and S. Teller, "Passive mobile robot localization within a fixed beacon field," in *Proc. Int'l Workshop on Algorithmic Foundations of Robotics (WAFR)*, Guanajuato, Mexico, December 2008, pp. 425–440.

[8] I. Amundson and X. D. Koutsoukos, "A survey on localization for mobile wireless sensor networks," in *Moblile Entity Localization and Tracking in GPS-less Environments*, 2009, pp. 235–254.

[9] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Effiient, generalized indoor WiFi GraphSLAM," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 1038–1043.

[10] R. Kennedy, K. Daniilidis, O. Naroditsky, and C. J. Taylor, "Identifying maximal rigid components in bearing-based localization," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, Vilamoura-Algarve, Portugal, October 2012, pp. 194–201.

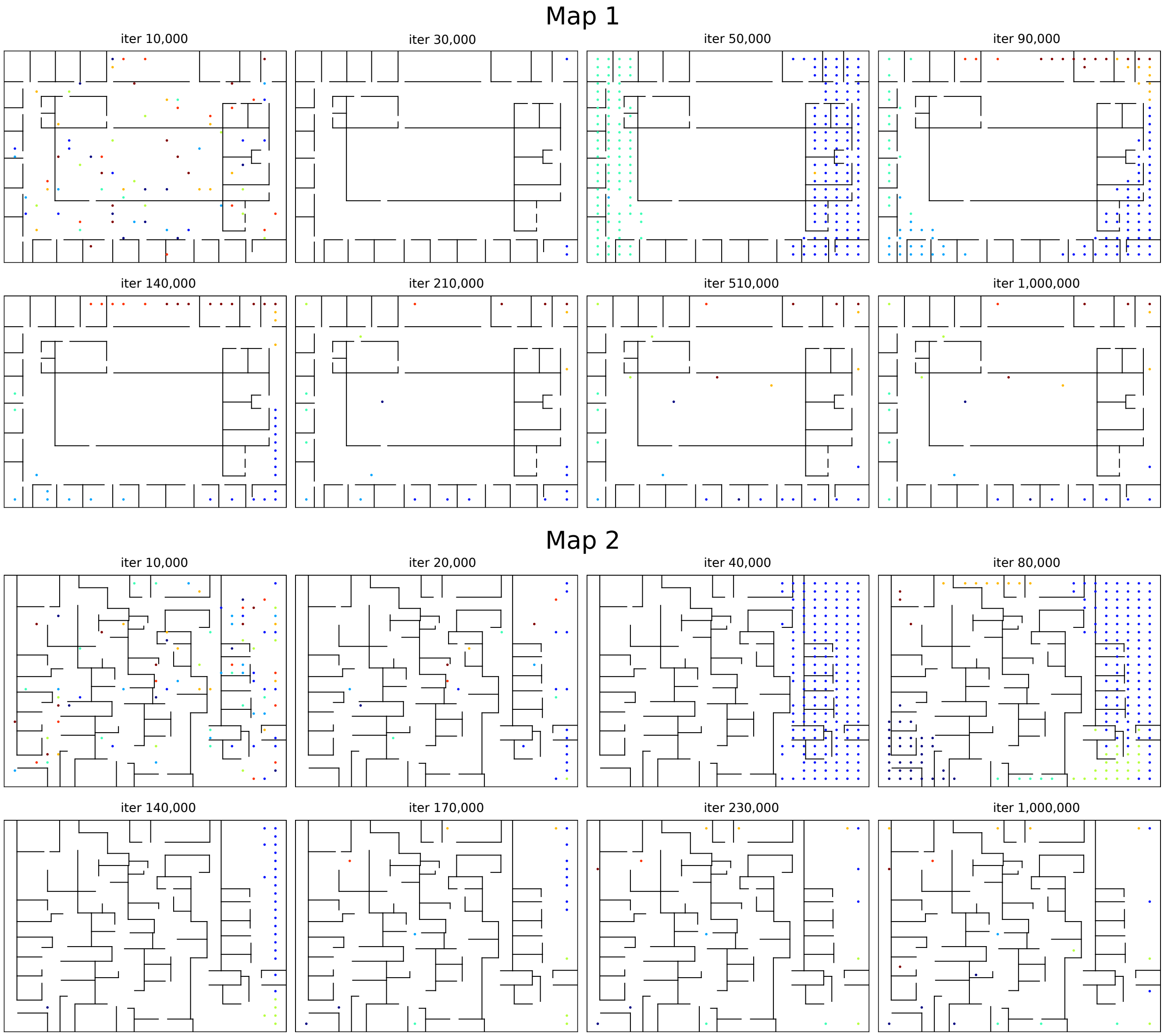

Fig. 5. The evolution of beacon placement throughout training on Map 1 and Map 2. The images depict a hard assignment, but the network is uncertain early in training, which explains the scarcity of beacons early on. The network quickly groups a large number of beacons and channel assignments along the edge of the map, but then gradually learns a sparse, diverse allocation, converging to a stable configuration around 200,000 iterations.

[11] J. Derenick, A. Speranzon, and R. Ghrist, "Homological sensing for mobile robot localization," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 572–579.

[12] H. González-Banos and J.-C. Latombe, "A randomized art-gallery algorithm for sensor placement," in *Proc. Symp. on Computational Geometry (SOCG)*, Medford, MA, June 2001, pp. 232–240.

[13] A. Chakrabarti, "Learning sensor multiplexing design through backpropagation," in *Advances in Neural Information Processing Systems (NIPS)*, Barcelona, Spain, December 2016.

[14] P. Prasithsangaree, P. Krishnamurthy, and P. Chrysanthis, "On indoor position location with wireless LANs," in *Proc. IEEE Int'l Symp. on Personal, Indoor and Mobile Radio Communications*, September 2002, pp. 720–724.

[15] A. Haeberlen, E. Flannery, A. M. Ladd, A. Rudys, D. S. Wallach, and L. Kavraki, "Practical robust localization over large-scale 802.11 wireless networks," in *Proc. Int'l Conf. on Mobile Computing and Networking (MobiCom)*, Philadelphia, PA, September 2004, pp. 70–84.

[16] J. geun Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, J. hicks, S. Teller, and J. Ledlie, "Growing an organic indoor localization system," in *Proc. Int'l Conf. on Mobile Systems, Applications, and Services (MobiSys)*, San Francisco, June 2010, pp. 271–284.

[17] A. S. M. Sala, R. G. Quiros, and E. E. Lopez, "Using neural networks and active RFID for indoor location services," in *Proc. European Workshop on Smart Objects: Systems, Technologies and Applications (RFID Sys Tech)*, Ciudad, Spain, June 2010.

[18] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," in *Proc. IEEE Int'l Symp. on Wireless Pervasive Computing (ISWPC)*, Modena, Italy, May 2010, pp. 295–300.

[19] N. B. Priyantha, “The Cricket indoor localization system,” Ph.D. dissertation, Massachusetts Institute of Technology, June 2005.

[20] Y. Gu, A. Lo, and I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[21] D. Moore, J. Leonard, D. Rus, and S. Teller, “Robust distributed network localization with noisy range measurements,” in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, Baltimore, MD, November 2004, pp. 50–61.

[22] A. Shareef, Y. Zhu, and M. Musavi, “Localization using neural networks in wireless sensor networks,” in *Proc. Int’l Conf. on Mobile Wireless Middleware, Operating Systems, and Applications (MOBILEWARE)*, Innsbruck, Austria, February 2008.

[23] P. K. Agarwal, E. Era, and S. K. Ganjugunte, “Efficient sensor placement for surveillance problems,” in *Proc. Int’l Conf. on Distributed Computing in Sensor Networks (DCOSS)*, Marina Del Rey, CA, June 2009, pp. 301–304.

[24] M. C. Shewry and H. P. Wynn, “Maximum entropy sampling,” *J. Applied Statistics*, vol. 14, no. 2, pp. 165–170, 1987.

[25] N. A. Cressie, *Statistics for Spatial Data*. John Wiley & Sons, 1993.

[26] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms, and empirical studies,” *J. Machine Learning Research*, vol. 9, pp. 235–284, February 2008.

[27] A. Cameron and H. Durrant-Whyte, “A Bayesian approach to optimal sensor placement,” *Int’l J. of Robotics Research*, vol. 9, no. 5, pp. 70–88, October 1990.

[28] S.-H. Fang and T.-N. Lin, “A novel access point placement approach for WLAN-based location systems,” in *Proc. IEEE Wireless Cmmunications and Networking Conf. (WCNC)*, Sydney, Australia, April 2010.

[29] S.-L. Kang, G. Y.-H. Chen, and J. Rogers, “Wireless LAN access point location planning,” in *Proc. Institute of Industrial Engineers Asian Conf.*, Tapei, Taiwan, July 2013, pp. 907–914.

[30] A. Hills, “Large-scale wireless LAN design,” *IEEE Communications Magazine*, vol. 39, no. 11, pp. 98–107, August 2002.

[31] X. Ling and K. L. Yeung, “Joint access point placement and channel assignment for 802.11 wireless LANs,” *IEEE Trans. on Wireless Communications*, vol. 5, no. 10, pp. 2705–2711, October 2006.

[32] V. Isler and R. Bajcsy, “The sensor selection problem for bounded uncertainty sensing models,” in *Proc. Int’l Symp. on Information Processing in Sensor Networks (IPSN)*, Los Angeles, CA, April 2005.

[33] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Trans. on Signal Processing*, vol. 57, no. 2, pp. 451–462, February 2009.

[34] T. S. Rappaport, *Wireless Communications: Princples and Practice*. Prentice Hall, 2002.