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, measurements of transmissions from fixed beacons provide an effective means of estimating a robot's location as it navigates. The accuracy of such a beaconbased localization system depends both on how beacons are distributed in the environment, and how the robot's location is inferred based on noisy and potentially ambiguous measurements. We propose an approach for making these design decisions automatically and without expert supervision, by explicitly searching for the placement and inference strategies that, together, are optimal for a given environment. Since this search is computationally expensive, our approach encodes beacon placement as a differential neural layer that interfaces with a neural network for inference. This formulation allows us to employ standard techniques for training neural networks to carry out the joint optimization. We evaluate this approach on a variety of environments and settings, and find that it is able to discover designs that enable high localization accuracy.

I. INTRODUCTION

Measurements obtained through a distributed network of sensors or beacons can be an effective means of monitoring location, or the spatial distribution of other phenomena. The measurements themselves only provide indirect or noisy information towards the physical properties of interest, and so additional computational processing is required for inference. Such inference must be designed to take into account possible degradations in the measurements, and exploit prior statistical knowledge of the environment. However, the success of inference, in the end, is limited by how the beacons and sensors were physically distributed in the first place.

Consider location-awareness, which is critical to human and robot navigation, resource discovery, asset tracking, logistical operations, and resource allocation [1]. In situations for which GPS is unavailable (indoors, underground, or underwater) or impractical, measurements of transmissions from fixed beacons [2–11] provide an attractive alternative. Designing a system for beacon-based location-awareness requires simultaneously deciding (a) how the beacons should be distributed (e.g., spatially and across transmission channels); and (b) how location should be determined based on measurements of signals received from these beacons.

Note that these decisions are inherently coupled. The placement of beacons and their channel allocation influence



Fig. 1. Our algorithm optimizes beacon placement and channel assignments jointly with an algorithm for location inference, based on measurements of transmissions from these beacons. receiver localization. Here, we show an example beacon distribution inferred by our method (colored circles denoting beacons and channel assignments), and the corresponding error (RMSE) in locations estimated by the inference method.

the nature of the ambiguity in measurements at different locations, and therefore which inference strategy is optimal. Therefore, one should ideally search over the space of both beacon allocation—which includes the number of beacons, and their placement and channel assignment—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, these decisions are decoupled in practice, and beacons are placed with some specific inference strategy in mind. This placement is most often performed manually by expert designers. In some cases, automatic placement methods are employed that apply heuristics (e.g., coverage or field-of-view [12]). However, such heuristics often rely on simplistic assumptions regarding the sensor and environment geometry, and do not adequately account for noise, interference, or other forms of degradation—e.g., ignoring interference or attenuation due to walls to directly map signal strength to observations of range or bearing. These heuristics are thus unsuitable for use in many real-world settings.

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 (across space and channels) and inference al-

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.

gorithm *jointly* for the task of localization based on raw signal transmissions. We instantiate the inference method as a neural network, and encode beacon allocation as a differentiable neural layer. We then describe an approach to jointly training the beacon layer and inference network, with respect to a given signal propagation model and environment, to enable accurate location-awareness in that environment.

We carry out evaluations under a variety of conditions with different environment layouts, different signal propagation parameters, different numbers of transmission channels, and different desired trade-offs against the number of beacons. In all cases, we find that our approach automatically discovers designs—each different and adapted to its environment and settings—that enable high localization accuracy. Therefore, our method provides a way to consistently create optimized location-awareness systems for arbitrary environments and signal types, without expert supervision.

II. RELATED WORK

Networks of sensors and beacons have been widely used for localization, tracking, and measuring other spatial phenomena. Many of the design challenges in sensor and beacon are related, since they involve problems that are duals of one another—based on whether the localization target is transmitting or receiving. In our work, we focus on localization with beacons, i.e., where an agent estimates its location based on transmissions received from fixed, known landmarks.

Most of the effort in localization is typically devoted to finding accurate inference methods, assuming the distribution and location of beacons in the environment are given. One setting for these methods is where sensor measurements can be assumed to provide direct, albeit possibly noisy, measurements of relative range or bearing from beacons an assumption that is typically based on a simple model for signal propagation. Location estimation then proceeds by using these relative range and/or bearing estimates and knowledge of beacon locations. 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 [14, 15].

Moore et al. [16] 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] describe 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 relative angular measurements, and Shareef et al. [17] use feed-forward and recurrent neural networks for localization based on noisy range measurements.

Another approach, common for radio frequency (RF) beacon and WiFi-based networks, is to infer location directly from received signal strength (RSS) signatures. One way to do this is by matching against a database of RSS-location pairs [18]. This database is typically generated manually via a site-survey, or "organically" during operation [19, 20]. Sala

et al. [21] and [22] adopt a different approach, training neural networks to predict a receiver's location within an existing beacon network based upon received signal strength.

The above methods deal with optimal ways of inferring location given an existing network of beacons. The decision of how to distribute these beacons, however, is often made manually based on expert intuition. Automated placement methods are used rarely, and for very specific settings, such as RSS fingerprint-based localization [23]. The most common of these is to ensure full coverage—i.e., to ensure that all locations are within an "acceptable range" of at least one beacon, assuming this condition is sufficient to guarantee accurate localization.

One common instance of optimizing placement for coverage is the standard art-gallery visibility problem [12] that seeks placements that ensure that all locations have line-ofsight to at least one beacon. This problem assumes a polygonal environment and that the beacons have an unlimited field-of-view, subject to occlusions by walls (e.g., cameras). Related, Agarwal et al. [24] propose a greedy landmarkbased method that solves for the placement of the minimum number of beacons (within a log factor) necessary to cover all but a fraction of a given polygonal environment. Note that these methods treat occlusions as absolute, while in practice, obstructions often only cause partial attenuationwith points that are close but obstructed observing similar signal strengths as those that are farther away. Kang et al. [25] provide an interesting alternative, and like us, use backpropagation to place WiFi access points-but again, only for the objective of maximizing coverage. Fang and Lin [23] consider localization accuracy for placing wireless access points to maximize receiver signal-to-noise ratios.

The above methods address spatial placement but not transmission channel assignments, and associated issues with interference. Automatic channel assignment methods have been considered previously, but only for optimizing communication throughput [26, 27]—i.e., to minimize interference from two beacons in the same channel at any location. Note that this is a very different and simpler objective than one of enabling accurate localization, where the goal is to ensure that there is a unique mapping from every RSS signature (with or without interference) to location.

Our approach provides a way to trade-off localization accuracy with the number of beacons, similar to the performance-cost trade-offs considered by the more general problem of sensor selection [28–30]. Some selection strategies are designed with specific inference strategies in mind. Shewry and Wynn [31] and Cressie [32] use a greedy entropy-based approach to place sensors, tied to a Gaussian process (GP) model that is used for inference. However, this approach does not model the accuracy of the predictions at the selected locations. Krause et al. [33] choose locations for a fixed sensor network that maximize mutual information using a GP model for phenomena over which inference is performed (e.g., temperature). However, these formulations require that the phenomena be modeled as a GP, and are thus not suitable for the task of beacon-based localization.



Fig. 2. A schematic of our 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 noise, interference, obstructions, etc. We optimize these assignment vectors jointly with parameters of the inference function $f(\cdot)$, to minimize error in the estimated locations \hat{v} .

III. APPROACH

We formalize the problem of designing a locationawareness 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 $\mathcal{D} = \{I_l\}$ as an assignment I_l to each 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 in which one of C possible configurations (i.e., $I_l^{c+1} = 1$). In our experiments, a beacon's configuration corresponds to the channel on which it broadcasts.

We parameterize the environment in terms of a function $s = \mathcal{E}(v, \{I_l\})$ that takes as input a location $v \in \mathbb{R}^2$ and a distribution of beacons $\mathcal{D} = \{I_l\}$, and produces a vector $s \in \mathbb{R}^m$ of measurements that an agent is likely to make at that location (Fig. 2). Note that the environment \mathcal{E} need not be a deterministic function. In the 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 $f(\cdot)$ such that $f(\mathcal{E}(v, \{I_l\})) \approx v$ for 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.

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 $\tilde{I}_l \in \mathbb{R}^{C+1}_+, |\tilde{I}_l| = 1$. The vector \tilde{I}_l can be interpreted as expressing a probability distribution over the possible assignments at location l.

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

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

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 backpropagated 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 describe an example of an environment mapping and its distributional version that satisfies this requirement.

Next, we simply choose the inference function $f(\cdot)$ to be differentiable and have 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 the loss:

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

where \mathcal{V} is the set of possible agent locations, Θ are the parameters of the inference function f, $\tilde{I}_l = \text{SoftMax}(\alpha w_l)$, 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.

Since we require f and $\hat{\mathcal{E}}$ to be differentiable, we can optimize both Θ and $\{w_l\}$ by minimizing Eqn. 2 with stochastic gradient descent (SGD), computing gradients over a small batch of locations $v \in \mathcal{V}$, with a single sample of sper 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 is assumed to be fixed

for every beacon, 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., the signal vector s is C-dimensional). We assume that the power of each beacon's signal drops as a function of distance and the number of obstructions (e.g., walls) in the line-of-sight between the agent and the beacon. The measured power in each channel at the receiver is then based on the amplitude of the superposition of signals from all beacons transmitting on 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 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

$$s^{c} = \left[\epsilon_{1} + \sum_{l} I_{l}^{c+1} \sqrt{P_{l}(v)} \cos \phi_{l}\right]^{2} + \left[\epsilon_{2} + \sum_{l} I_{l}^{c+1} \sqrt{P_{l}(v)} \sin \phi_{l}\right]^{2}, \quad (4)$$

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 .

During 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

$$R({\tilde{I}_l}) = \lambda \sum_l \tilde{I}_l^1.$$
 (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 three different environment maps. As discussed in Section II, existing methods are not wellsuited to beacon placement for localization. Consequently, we compare our method to to several hand-designed beacon allocation strategies. We first show that our inference network is effective at localization given the baseline placements by comparing to a standard nearest-neighbors method. We then analyze the performance when learning the beacon allocation and inference network jointly, demonstrating that our method learns a distribution and inference strategy that enable high localization accuracy for all three environments. We end by analyzing the effects of different degrees of regularization and different numbers of available channels, as well as the variation in the learned beacon distributions based on parameters of the environment function \mathcal{E} .

A. Setup

We conduct our experiments on three manually drawn environment maps, which correspond to floor plans (of size 1×0.7 map units) with walls that serve 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 $\gamma = 1.25 \times 10^{-9}$ for the quadratic temperature scheme. 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 900k iterations, we switch the softmax to an "arg-max", effectively setting α to infinity and fixing the beacon placement, and then continue training the inference network.

The inference function $f(\cdot)$ is parameterized as a 13layer 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 4 units. After the last 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 are trained by minimizing the loss defined in Eqn. 2 for 1000k iterations using SGD with a learning rate of 0.01 and momentum 0.9, followed by an additional 100k iterations with a learning rate 0.001. We also use batch-normalization in all hidden layers.

B. Experiments

We evaluate the effectiveness of our approach for joint optimization of beacon placement and localization through a series of experiments. Localization performance can vary significantly within an environment. Consequently, for each



Fig. 3. Localization error maps for learned and handcrafted beacon distributions (with 8 channels) for three environments. We show three learned distributions for each map—learned with annealed regularization, and two settings (high, low) of fixed regularization—as well as the best performing handcrafted placement. The overall RMSE is indicated below each error map (see Fig. 1 for color mapping for errors).

beacon allocation and inference strategy, we report both average as well as worst-case performance over a dense set of locations, with multiple samples (corresponding to different random noise and interference phases) per location. We measure performance in terms of the root mean squared error (RMSE)—between estimated and true coordinates over all samples at all locations, as well as over the worst sample at each location, which we refer to as *worst-case RMSE*. We also measure the frequency with which large errors occur—defined with respect to different thresholds (0.1, 0.2, 0.5)—and report these as *failure rates*.

Table I reports these metrics on the three environment maps for different versions our approach that vary the regularization settings. We have also experimented with a number of manually handcrafted distribution strategies for these maps and report the performance of the two strategies that worked best in Table I. For the handcrafted settings, we report the result of training a neural network for inference, as well as of k-nearest neighbors (kNN)-based inference (we try $k \in \{1, 5, 10, 20\}$ and pick the best). These latter experiments show that the network-based inference performs well (better than the kNN baseline) and, therefore, that our architecture is reasonable for the task. Moreover, the results reveal that jointly optimizing beacon allocation and inference provides accuracies that exceed the handcrafted baselines, yielding different distribution strategies with different numbers of beacons. Figure 3 visualizes the beacon placement and channel allocation along with the RMSE for both learned and handcrafted beacon allocations.

Next, we report a more detailed evaluation of the regularization scheme defined in Eqn. 5, and therefore the ability of our method to allow for a trade-off between the number of beacons placed and accuracy. First, we use a constant value of λ , trying various values between 0.0 and 0.2. As Figure 4 shows, increased regularization leads to solutions with fewer beacons. On Map 2, we find that decreased regularization always leads to solutions with lower error. On Maps 1 and 3, however, unregularized beacon placement results in increased localization error. This suggests that regularization may also allow our model to escape bad local minima during training. Then, we experiment with an annealing scheme for λ and find that it leads to a better performance-cost trade-off. We use a simple annealing schedule that decays $\lambda = 0.2$ by a constant factor $\eta = 0.25$ every 100k iterations.

Figure 5 shows the evolution of beacon distributions throughout training (when using an annealed regularizer). Note that the images depict a hard assignment, however the network reasons over high entropy placements early in training, which explains the initial sparsity. With each map, the network quickly clusters a large number of beacons by channel, and gradually learns to reduce the number of beacons while increasing channel diversity.

Next, we evaluate the ability of our method to automatically discover successful placement and inference strategies

TABLE I							
PERFORMANCE COMPARISON FOR DIFFERENT INFERENCE AND PLACEMENT STRATEGIES							

Map	Allocation	Inference	Beacons	RMSE	RMSE (Worst-case)	Failure Rate (0.10)	Failure Rate (0.20)	Failure Rate (0.50)
1	Handcrafted A	kNN	544	0.0817	0.1793	19.1946~%	1.4662~%	0.0096~%
	Handcrafted B	kNN	180	0.0998	0.2384	26.1523~%	4.7400~%	0.0600~%
	Handcrafted A	Network	544	0.0716	0.1537	$13.5691 \ \%$	0.6706~%	0.0045~%
	Handcrafted B	Network	180	0.0811	0.1940	17.1529~%	2.4632~%	0.0192~%
	Learned (low fixed reg.)	Network	183	0.0567	0.1336	$6.4917 \ \%$	0.4866~%	0.0053~%
	Learned (high fixed reg.)	Network	12	0.0633	0.1511	$8.0217 \ \%$	1.3686~%	0.0827~%
	Learned (annealed reg.)	Network	25	0.0497	0.1169	4.7705 ~%	$0.3706\ \%$	0.0125~%
2	Handcrafted A	kNN	544	0.0806	0.1708	18.7127~%	1.3098~%	0.0023~%
	Handcrafted B	kNN	180	0.0839	0.2040	18.3838~%	2.4336~%	0.0306~%
	Handcrafted A	Network	544	0.0653	0.1331	$10.4884 \ \%$	0.2718~%	0.0001 %
	Handcrafted B	Network	180	0.0624	0.1479	9.1473~%	0.6843~%	0.0013~%
	Learned (low fixed reg.)	Network	371	0.0491	0.1154	4.3015~%	0.2931~%	0.0016~%
	Learned (high fixed reg.)	Network	13	0.0688	0.1450	$11.9567 \ \%$	1.4807~%	0.0040~%
	Learned (annealed reg.)	Network	35	0.0473	0.1142	4.6226~%	0.4182~%	0.0091~%
3	Handcrafted A	kNN	544	0.0814	0.1717	18.7939~%	1.6503~%	0.0027~%
	Handcrafted B	kNN	180	0.0840	0.2127	16.6865~%	2.6309~%	0.0978~%
	Handcrafted A	Network	544	0.0670	0.1432	11.1586~%	0.6904~%	0.0009 %
	Handcrafted B	Network	180	0.0649	0.1581	9.7563~%	1.0512~%	0.0079~%
	Learned (low fixed reg.)	Network	325	0.0506	0.1200	4.2496~%	0.3482~%	0.0056~%
	Learned (high fixed reg.)	Network	14	0.0680	0.1495	11.6651~%	1.2166~%	0.0454~%
	Learned (annealed reg.)	Network	43	0.0493	0.1158	4.7211~%	0.6529~%	0.0045~%



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 with annealed regularization on each map.

for different environmental conditions and constraints in Table II. All results are on Map 1, with high fixed regularization (0.2). We report results for a propagation model with decreased attenuation at walls ($\beta = e^{-0.2}$), and one with increased noise ($\sigma_z^2 = 2.5 \times 10^{-4}$). 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. We also experiment with fewer (4) and more (16) available RF channels. As expected, the availability of more channels allows our method to learn a more accurate localization system. More broadly, these experiments show that our approach can enable the automated design of location awareness systems in diverse settings.

Finally, we evaluate the robustness of the joint placement

TABLE II Modified Environment Settings

Scenario	Beacons	RMSE	RMSE (Worst-case)
Original	12	0.0633	0.1511
Low Attenuation	8	0.0426	0.0910
High Noise	27	0.1400	0.2907
Fewer Channels (4)	11	0.1290	0.2581
More Channels (16)	12	0.0397	0.0910

TABLE III Training Stability over Multiple Runs

	Mean	Std. Dev.	Min	Max
RMSE	0.057	0.0019	0.0541	0.0596
Worst-case RMSE	0.1365	0.0039	0.1289	0.1413
Num. Beacons	93.50	11.17	79	119

and inference optimization with different random initializations. We repeat training on Map 1 (with fixed regularization of 0.04) ten times, and report the deviations in the error metrics and numbers of beacons placed in Table III.

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 an optimal design of a location-awareness system for arbitrary environments. We evaluated our method for the task of RF-based localization and demonstrated its ability to consistently discover high-quality localization systems for a variety of environment layouts and propagation models, without expert supervision.





Fig. 5. The evolution of beacon distribution throughout training (with annealed regularization). 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 200k iterations.

Additionally, we presented a strategy that trades off the number of beacons placed and the achievable accuracy. While we describe our method in the context of localization, the approach generalizes to problems that involve estimating a broader class of spatial phenomena using sensor networks. A reference implementation of our algorithm is available on the project page at http://ripl.ttic.edu/nbp.

VI. ACKNOWLEDGEMENTS

We thank NVIDIA for the donation of Titan X GPUs used in this research. This work was supported in part by the National Science Foundation under Grant IIS-1638072.

REFERENCES

- 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 *Mobile 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 Graph-SLAM," 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.
- [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 artgallery 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, pp. 3081–3089.
- [14] N. B. Priyantha, "The Cricket indoor localization system," Ph.D. dissertation, Massachusetts Institute of Technology, June 2005.
- [15] 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.
- [16] 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.
- [17] 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.

- [18] 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.
- [19] 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.
- [20] 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.
- [21] 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, Tech*nologies and Applications (*RFID Sys Tech*), Ciudad, Spain, June 2010.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] A. Hills, "Large-scale wireless LAN design," *IEEE Communications Magazine*, vol. 39, no. 11, pp. 98–107, August 2002.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. on Signal Processing*, vol. 57, no. 2, pp. 451–462, February 2009.
- [31] M. C. Shewry and H. P. Wynn, "Maximum entropy sampling," J. Applied Statistics, vol. 14, no. 2, pp. 165–170, 1987.
- [32] N. A. Cressie, Statistics for Spatial Data. John Wiley & Sons, 1993.
- [33] 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.