Learning Deep Generative Spatial Models for Mobile Robots

Andrzej Pronobis, Rajesh P. N. Rao

***Abstract*— We propose a new probabilistic framework that allows mobile robots to autonomously learn deep generative models of their environments that span multiple levels of abstraction. Unlike traditional approaches that integrate engineered models for low-level features, geometry, and semantic information, our approach leverages recent advances in Sum-Product Networks (SPNs) and deep learning to learn a generative model of a robot's spatial environment, from low-level input to semantic interpretations. Our model is capable of solving a wide range of tasks from semantic classification of places, uncertainty estimation and novelty detection to generation of place appearances based on semantic information and prediction of missing data in partial observations. Experiments on laser range data from a mobile robot show that the proposed single universal model obtains accuracy or efficiency superior to models fine-tuned to specific sub-problems, such as Generative Adversarial Networks (GANs) or SVMs.**

I. INTRODUCTION

The ability to represent knowledge about the environment is fundamental for a mobile robot. Spatial knowledge exists at multiple levels of abstraction, from robot's sensory data, through geometry and appearance, up to high level semantic descriptions. Experiments have demonstrated that robotic systems can leverage knowledge at all levels of abstraction to better perform real-world tasks in human environments [1].

Traditionally, robotic systems utilize an assembly of independent spatial models [2], which exchange information in a limited fashion. This includes engineered feature extractors and combinations of machine learning techniques, making integration with planing and decision making difficult. However, the recent deep learning revolution has demonstrated that replacing multiple representations with a single integrated model can lead to a drastic increase in performance [3][4]. As a result, deep models have also been applied to the tasks of place classification and semantic mapping [5][6]. However, the problem was always framed as one of classification, where sensory data is fed to a convolutional neural network (CNN) to obtain semantic labels.

In contrast, in this work our goal is not only to unify multiple levels of a representation into a single model, but also to demonstrate that the role of a spatial model can go beyond classification. To this end, we propose the Deep Generative Spatial Model (DGSM), a probabilistic model which learns a joint distribution between a low-level representation of the geometry of local environments (places) and their semantic interpretation. Our model leverages Sum-Product Networks (SPNs), a novel probabilistic deep architecture [7][8].

The authors are with Computer Science & Engineering, University of Washington, Seattle, WA, USA. A. Pronobis is also with Robotics, Perception and Learning Lab, KTH Royal Institute of Technology, Stockholm, Sweden. {pronobis,rao}@cs.washington.edu. This work was supported by the Swedish Research Council (VR) project SKAEENet. The help by Kaiyu Zheng and Kousuke Ariga is gratefully acknowledged.

SPNs have been shown to provide state-of-the-art results in several domains [9][10][11]. However, their potential has not yet been exploited in robotics. DGSM consists of an SPN with a unique structure designed to hierarchically represent the geometry and semantics of a place from the perspective of a mobile robot acting in an environment. To this end, the network represents a place as a polar occupancy grid surrounding the robot, where the nearby objects are represented in more detail than objects further apart. On top of the occupancy data, we propose a unique network structure which combines prior knowledge about the problem with random structure generation (as in random forests) for parts of the network modeling complex dependencies.

DGSM is generative, probabilistic, and therefore universal. Once learned, it enables a wide range of inferences. First, it can be used to infer a semantic category of a place from sensory input together with probability representing uncertainty. Probabilistic output provides rich information to a potential planning or decision-making subsystem. However, as shown in this work, it can also be used to detect novel place categories. Furthermore, the model reasons jointly about the geometry of the world and its semantics. We exploit that property for two tasks: to generate prototypical appearances of places based on semantic information, and infer missing geometry information in partial observations. We use laser range data to capture the geometry of places, however the proposed model can be easily extended to include 3D and visual information without changing the general architecture.

Our goal is to present the potential of DGSM, and deep generative models in general, to spatial modeling in robotics. Therefore, we present results of four different experiments addressing each of the inference tasks. In each experiment, we compare our universal model to an alternative approach that is designed for and fine-tuned to a specific task. First, for semantic categorization, we compare to well-established Support Vector Machine (SVM) model learned on widely used geometrical laser range features [12][13]. Second, we benchmark novelty detection against one-class SVM trained on the same features. In both cases, DGSM offers superior accuracy. Finally, we compare the generative properties of our model to Generative Adversarial Networks (GANs) [14][15] on the two remaining inference tasks, reaching state-of-the-art accuracy and superior efficiency beyond real-time. This serves as a benchmark, but also demonstrates the use of GANs for spatial modeling in robotics. Importantly, to open doors for the use of SPNs in a broader range of applications, we contribute *LibSPN*, a new generic library implementing various SPN architectures as well as inference and learning algorithms on GPUs [1].

[1]The library will be open-sourced upon acceptance of the paper.

Preprint submitted to 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2017.

II. RELATED WORK

Representing semantic spatial knowledge is a broadly researched topic, with many solutions employing vision [16][17][2][5]. Images clearly carry rich information about semantics; however, they are also affected by changing environment conditions. At the same time, robotics researchers have seen advantages of using range data that are much more robust in real-world settings and easier to process in real time. In this work, we focus on laser range data, as a way of introducing and evaluating a new spatial model, employing a recently proposed deep architecture.

Laser range data have been extensively used for place classification and semantic mapping, and many traditional, handcrafted representations have been proposed. Buschka et al. [18] contributed a simple method that incrementally divided grid maps of indoor environments into two classes of open spaces (rooms and corridors). Mozos et al. [12] applied AdaBoost to create a classifier based on a set of manually designed geometrical features to classify places in into rooms, corridors and doorways. In [19], omnidirectional vision was combined with laser data to build descriptors, called fingerprints of places. Finally, in [13], SVMs have been applied to the geometrical features of Mozos et al. [12] leading to improved performance over the original AdaBoost. That approach has been further integrated with visual and object cues for semantic mapping in [2].

Deep learning and unsupervised feature learning, after many successes in speech recognition and computer vision [3], entered the field of robotics with superior performance in object recognition [20][21] and robot grasping [22][4]. The latest work in place categorization also employs deep approaches. In [5], deep convolutional network (CNN) complemented with a series of one-vs-all classifiers is used for visual semantic mapping. In [6], CNNs are used to classify grid maps built from laser data into 3 classes: room, corridor, and doorway. In these works, the deep model is used exclusively for classification, and use of generative models has not been explored. In contrast, we propose a universal probabilistic generative model and demonstrate its usefulness in a wide range of robotics tasks, including classification.

Several generative, deep architectures have recently been proposed, notably Variational Autoencoders [23], Generative Adversarial Networks [14], and Sum-Product Networks [8][7][9]. GANs have been shown to produce high-quality generative representations of visual data [15], and have been successfully applied to image completion [24]. SPNs, a probabilistic model, achieved promising results for such varied applications as speech [10] and language modeling [25], human activity recognition [11], and image classification [9] and completion [7], but have not been used in robotics. In this work, we exploit the universality and efficiency of SPNs to propose a single spatial model able to solve a wide range of inference problems relevant to a mobile robot. Furthermore, inspired by their results in other domains, we also evaluate GANs (when applicable). This serves as a comparison and a demonstration of GANs on a new application.

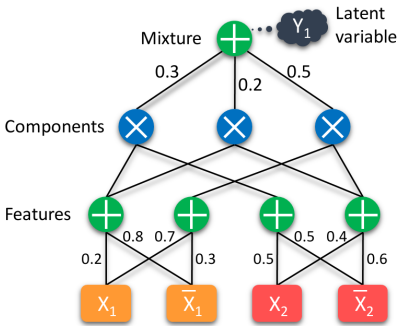


Fig. 1: An SPN for a naive Bayes mixture model $P(X_1, X_2)$, with three components over two binary variables. The bottom layer consists of indicators for each of the two variables. Weights are attached to inputs of sums. Y_1 represents a latent variable marginalized out by the top sum node.

III. SUM-PRODUCT NETWORKS

Sum-product networks are a recently proposed probabilistic deep architecture with several appealing properties and solid theoretical foundations [8][7][9]. One of the primary limitations of probabilistic graphical models is the complexity of their partition function, often requiring complex approximate inference in the presence of non-convex likelihood functions. In contrast, SPNs represent probability distributions with partition functions that are guaranteed to be tractable, involve a polynomial number of sums and product operations, permitting exact inference. While not all probability distributions can be encoded by polynomial-sized SPNs, recent experiments in several domains show that the class of distributions modeled by SPNs is sufficient for many real-world problems, offering real-time efficiency. SPNs can perform fast, tractable inference on high-treewidth models.

SPNs model a joint or conditional probability distribution and can be learned both generatively [7] and discriminatively [9]. They are a deep, hierarchical representation, capable of representing context-specific independence. As shown in Fig. 1 on a simple example of a naive Bayes mixture model, the network is a generalized directed acyclic graph of alternating layers of weighted sum and product nodes. The sum nodes can be seen as mixture models, over components defined using product nodes, with weights of each sum representing mixture priors. The latent variables of such mixtures can be made explicit and their values inferred. This technique is often used for classification models where the root sum is a mixture of sub-SPNs representing multiple classes. The bottom layers effectively define features reacting to certain values of indicators[2] for the input variables.

Formally, following Poon & Domingos [7], we can define an SPN as follows:

Definition 1: An SPN over variables $X_1, \ldots, X_V$ is a rooted directed acyclic graph whose leaves are the indicators $(X_1^1, \ldots, X_1^I), \ldots, (X_V^1, \ldots, X_V^I)$ and whose internal nodes are sums and products. Each edge (i, j) emanating from a sum node i has a non-negative weight w_{ij}. The value of a

[2]Indicator is a binary variable set to 1 when the corresponding categorical variable takes a specific value. For using continuous input variables, see [7].

Preprint submitted to 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2017.

product node is the product of the values of its children. The value of a sum node is $\sum_{j \in Ch(i)} w_{ij} v_j$, where $Ch(i)$ are the children of i and v_j is the value of node j. The value of an SPN $S[X_1, \ldots, X_V]$ is the value of its root.

Not all possible architectures consisting of sums and products will result in a valid probability distribution. However, following simple constraints on the structure of an SPN will guarantee validity (see [7], [8] for details). When the weights of each sum node are normalized to sum to 1, the value of a valid SPN $S[X_1^1, \ldots, X_V^I]$ is equal to the normalized probability $P(X_1, \ldots, X_V)$ of the distribution modeled by the network [8].

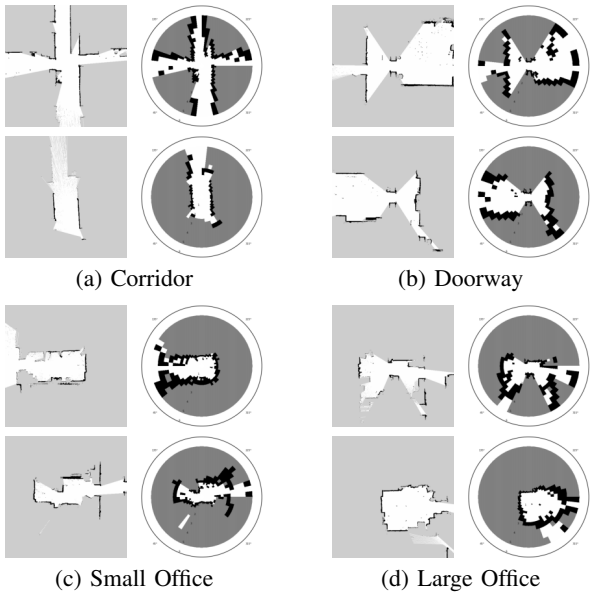
A. Generating SPN structure

The structure of the SPN determines the group of distributions that can be learned. Therefore, most previous works [9][11][25] relied on domain knowledge to design the appropriate structure. Furthermore, several structure learning algorithms were proposed [26][27] to discover independencies between the random variables in the dataset and structure the SPN accordingly. In this work, we experiment with a different approach, originally hinted at in [7], which generates a random structure, as in random forests. Such approach has not been previously evaluated and our experiments demonstrate that it can lead to very good performance and can accommodate a wide range of distributions.

The algorithm recursively generates nodes based on multiple decompositions of a set of random variables into multiple subsets until each subset is a singleton. As illustrated in Fig. 3, at each level the set to be decomposed is modeled by multiple mixtures (green nodes), and each subset in each decomposition is also modeled by multiple mixtures (green nodes one level below). Product nodes (blue) are used as an intermediate layer and act as features detecting particular combinations of mixtures representing each subset in a particular decomposition (e.g., in each of the two decompositions at the top of Fig. 3, the first product node is a feature based on the first mixture for each subset, while the last product node combines the last mixture for each subset). The top mixtures of each level mix outputs of all product nodes at that level with independent weights.

B. Inference and Learning

Inference in SPNs is accomplished by an upward pass through the network. Once the indicators are set to represent the evidence, the upward pass will yield the probability of the evidence as the value of the root node. Partial evidence (or missing data) can easily be expressed by setting all indicators for a variable to 1. Moreover, since SPNs compute a network polynomial [28], derivatives computed over the network can be used to perform inference for modified evidence without recomputing the whole SPN. Finally, it can be shown [7] that MPE inference can be performed by replacing all sum nodes with max nodes, while retaining the weights. Then, the indicators of the variables for which the MPE state is inferred are all set to 1 and a standard upward pass is performed. A downward pass then follows which recursively selects the highest valued child of each sum (max) node, and all children of a product node. The indicators selected by this process indicate the MPE state of the variables.



(a) Corridor (b) Doorway

(c) Small Office (d) Large Office

Fig. 2: Comparison of local environment observations used in our experiments, expressed as Cartesian and polar occupancy grids for places of different semantic categories.

SPNs lend themselves to be learned generatively or discriminatively using Expectation Maximization (EM) or gradient descent. In this work, we employ hard EM, which was shown to work well for generative learning [7]. As is often the case for deep models, the gradient quickly diminishes as more layers are added. Hard EM overcomes this problem, permitting learning of SPNs with hundreds of layers. Each iteration of the EM learning consists of an MPE inference of the implicit latent variables of each sum with training samples set as evidence (E step), and an update of weights based on the inference results (M step, for details see [7]). We achieved best results by modifying the MPE inference to use sums instead of maxes during the upwards pass, while selecting the max valued child during the downward pass. Furthermore, we performed additive smoothing when updating the weights corresponding to a Dirichlet prior and terminated learning after 300 iterations. No additional learning parameters are required.

IV. DGSM SPATIAL MODEL

A. Representing Local Environments

DGSM is designed to support real-time, online spatial reasoning on a mobile robot. To represent an observation of the local environment (a place), the model relies on local occupancy grids generated from laser range data. A real mobile robot almost always has access to more than one observation of a place. Thus, we use a particle filter grid mapping [29] to integrate the incoming laser scans into local robo-centric maps of 5m radius. This technique allows us to assemble more complete representations of places. During our experiments, the robot was exploring a new environment

driving with a constant speed, while continuously gathering data and performing inference based on the current state of the local map. This will still result in partial maps (especially when the robot enters a new room), but will help to accumulate observations over time.

Our goal is to model the geometry and semantics of a local environment only. We assume that larger-scale space model will be built by integrating multiple models of places. Thus, we constrain the observation to the parts of the environment that are visible from the robot (can be raytraced from the robot's location). As a result, walls occlude the view and the local map will mostly contain objects in a single room. In practice, additional noise is almost always present, but is averaged out during learning of the model. Examples of such local representations can be seen in Fig. 2.

Next, every local observation is transformed into a robocentric polar occupancy grid representation (compare polar and Cartesian grids in Fig. 2). The resulting observation contains higher-resolution details closer to the robot and lower-resolution information further away. This focuses the attention of the model to the nearby objects. Higher resolution of information closer to the robot is important for understanding the semantics of the exact robot's location (for instance when the robot is in a doorway). However, it also relates to how spatial information is used by a mobile robot when planning and executing actions. It is in the vicinity of the robot that higher accuracy of spatial information is required. A similar principle is exploited by many navigation components, which use different resolution of information for local and global path planning. Additionally, such representation corresponds to the way the robot perceives the world because of the limited resolution of its sensors. Our goal is to use a similar strategy when representing 3D and visual information in the future, by extending the polar representation to 3 dimensions. Finally, a high-resolution map of the complete environment can be largely recovered by integrating the polar observations over the path of the robot. The polar grids in our experiments assumed radius of $5m$, with angle step of 6.4 degrees and resolution decreasing with the distance from the robot.

B. Architecture of DGSM

The architecture of DGSM is based on a generative SPN illustrated in Fig. 3. The model learns a probability distribution $P(Y, X_1, \ldots, X_C)$, where Y represents the semantic category of a place, and $X_1, \ldots, X_C$ are input variables representing the occupancy in each cell of the polar grid. Each occupancy cell is represented by three indicators in the SPN (for empty, occupied and unknown space). These indicators constitute the bottom of the network (orange nodes).

The structure of the model is partially static and partially generated randomly according to the algorithm described in III-A. We begin by splitting the polar grid equally into 8 *views* (45 degrees each). For each view, we randomly generate an SPN by recursively building a hierarchy of decompositions of subsets of polar cells. Then, on top of all the sub-SPNs representing the views, we randomly generate an SPN representing complete place geometries for each

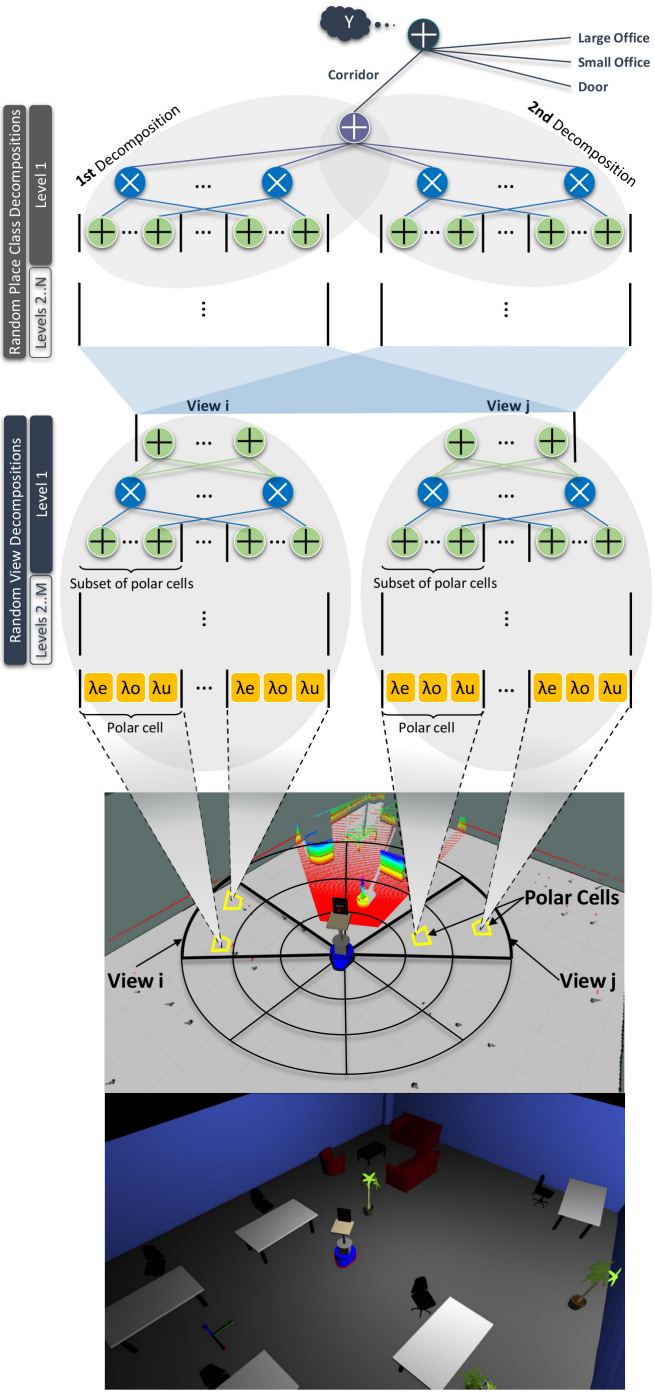


Fig. 3: The structure of the SPN implementing our spatial model. The bottom images illustrate a robot in an environment and a robocentric polar grid formed around the robot. The SPN is built on top of the variables representing the occupancy in the polar grid.

place class. The sub-SPNs for place classes are combined by a sum node forming the root of the network. The latent variable associated with the root sum node is made explicit as Y and is set to the appropriate class label during learning. Similarly, we infer its value when classifying observations.

Sub-dividing the representation into views allows us to use networks of different complexity for representing lower-level

features and geometry of each view, as well as for modeling the top composition of views into a place class. In our experiments, when representing views, we recursively decomposed the set of polar cells using a single decomposition at each level, into 2 random cell sub-sets, and generated 4 mixtures to model each such subset. This procedure was repeated until each subset contained a single variable representing a single cell. To increase the discriminative power of each view representation, we used 14 sums at the top level of the view sub-SPN.

Then, all sums modeling views were considered input to a randomly generated SPN structure representing *place classes*. To ensure that each class can be associated with a rich assortment of place geometries, we increased the complexity of the generated structure and at each level performed 4 random decompositions of the sets of mixtures representing views into 5 subsets. The performance of the model does not vary greatly with the structure parameters as long as the random sub-SPNs are sufficiently complex to capture the dependencies in the data.

Several straightforward modifications to the architecture can be considered. First, the weights of the sum nodes for each view could be shared, potentially simplifying the learning process. Second, the latent variables in the mixtures modeling each view can be accessed to explicitly reason about types of views discovered by the learning algorithm.

C. Types of Inference

As a generative model of a joint distribution between low-level observations and high-level semantic phenomena, DGSM is capable of various types of inferences.

First, the model can simply be used to classify observations into semantic categories, which corresponds to MPE inference of y: $y^* = \text{argmax}_y P(y|x_1, \ldots, x_C)$. Second, the probability of an observation can be used as a measure of novelty and thresholded: $\sum_y P(y, x_1, \ldots, x_C) > t$. We use this approach to separate test observations of classes known during training from observations of unknown classes.

If instead, we condition on the semantic information, we can perform MPE inference over the variables representing occupancy of polar grid cells:

$$x_1^*, \ldots, x_C^* = \underset{x_1,\ldots,x_C}{\text{argmax}}\ P(x_1, \ldots, x_C|y).$$

This leads to generation of most likely, prototypical examples for each class. Finally, we can use partial evidence about the occupancy and infer most likely state of a subset of polar grid cells for which evidence is missing:

$$x_J^*, \ldots, x_C^* = \underset{x_J,\ldots,x_C}{\text{argmax}} \sum_y P(y, x_1, \ldots, x_J, \ldots, x_C)$$

We use this technique to infer missing observations in our experiments.

V. GANS FOR SPATIAL MODELING

Recently, Generative Adversarial Networks [14] have received significant attention for their ability to learn complex visual phenomena in an unsupervised way [15], [24]. The idea behind GANs is to simultaneously train two deep networks: a generative model $G(\boldsymbol{z}; \theta_g)$ that captures the data distribution and a discriminative model $D(\boldsymbol{x}; \theta_d)$ that discriminates between samples from the training data and samples generated by G. The training alternates between updating θ_d to correctly discriminate between the true and generated data samples and updating θ_g so that D is fooled. The generator is defined as a function of noise variables $\boldsymbol{z}$, typically distributed uniformly (values from -1 to 1 in our experiments). For every value of $\boldsymbol{z}$ a trained G should produce a sample from the data distribution.

Although, GANs have been known to be unstable to train, several architectures have been proposed that result in stable models over a wide range of datasets. Here, we employ one such architecture called DC-GAN [15], which provides excellent results on such datasets as MNIST, LSUN, ImageNet [15] or CelebA [24]. Specifically, we used 3 convolutional layers (of dimensions $18\times18\times64$, $9\times9\times128$, $5\times5\times256$) with stride 2 and one fully-connected layer for D[3]. We used an analogous architecture based on fractional strided convolutions for G. We assumed $\boldsymbol{z}$ to be of size 100. DC-GANs do not use pooling layers, perform batch normalization for both D an G, and use ReLU and LeakyReLu activations for D and G, respectively. We used ADAM to learn the parameters.

Since DC-GAN is a convolutional model, we could not directly use the polar representation as input. Instead, we used the Cartesian local grid maps directly. The resolution of the Cartesian maps was set to $36x36$, which is larger then the average resolution of the polar grid, resulting in 1296 occupancy values being fed to the DC-GAN, as compared to 1176 for DGSM. We encoded input occupancy values into a single channel[3] (0, 0.5, 1 for unknown, empty, and occupied).

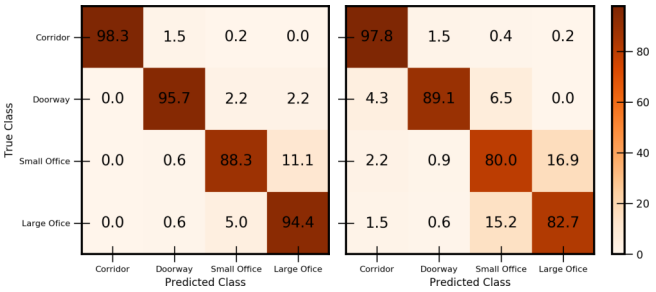
A. Predicting Missing Observations

In [24], a method was proposed for applying GANs to the problem of image completion. The approach first trains a GAN on the training set and then relies on stochastic gradient descent to adjust the value of $\boldsymbol{z}$ according to a loss function $\mathcal{L}(\boldsymbol{z}) = \mathcal{L}_c(\boldsymbol{z}) + \lambda\mathcal{L}_p(\boldsymbol{z})$, where $\mathcal{L}_c$ is a contextual loss measuring the similarity between the generated and true known input values, while $\mathcal{L}_p$ is a perceptual loss which ensures that the recovered missing values look real to the discriminator. We use this approach to infer missing observations in our experiments. While effective, it requires iterative optimization to infer the missing values. This is in contrast to DGSM, which performs inference using a single up/down pass through the network. We selected the parameter λ to obtain the highest ratio of correctly reconstructed pixels.

VI. EXPERIMENTS

We conducted four types of experiments corresponding to the types of inference described in Sec. IV-C. The same instance of a DGSM model was used for all inferences.

[3]We evaluated architectures consisting of 4 conv. layers and layers of different dimensions (depth of the 1st layer ranging from 32 to 256). We also investigated using two and three channels to encode occupancy information. The final architecture results in significantly better completion accuracy.



(a) DGSM (b) SVM + Geometric Features

Fig. 4: Normalized confusion matrices for the task of semantic place categorization.

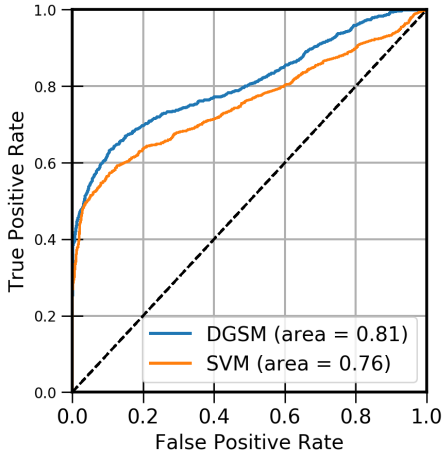


Fig. 5: ROC curves for novelty detection. Inliers are considered positive, while novel samples are negative.

A. Experimental Setup

Our experiments were performed on laser range data from the COLD-Stockholm database [2]. The database contains multiple data sequences captured using a mobile robot navigating with constant speed through four different floors of an office building. On each floor, the robot navigates through rooms of different semantic categories. Four of the room categories contain multiple room instances, evenly distributed across floors. There are 9 different *large offices*, 8 different *small offices*, 4 long *corridors* (1 per floor, with varying appearance in different parts), and multiple examples of observations captured when the robot was moving through *doorways*. The dataset features several other room categories: an elevator, a living room, a meeting room, a large meeting room, and a kitchen. However, with only one or two room instances in each. Therefore, we decided to use the four categories with multiple room instances for the majority of the experiments and designated the remaining classes as novel when testing novelty detection.

To ensure variability between the training and testing sets, we split the samples from the four room categories four times, each time training the model on samples from three floors and leaving one floor out for testing. The presented results are averaged over the four splits.

The experiments were conducted using *LibSPN*. SPNs are still a new architecture, and only few, limited domain-specific implementations exist at the time of writing. In contrast, our library offers a generic toolbox for structure generation, learning and inference and enables quick application of SPNs to new domains. It integrates with TensorFlow, which leads to an efficient solution capable of utilizing multiple GPUs, and enables combining SPNs with other deep architectures. The presented experiments are as much an evaluation of DGSM as they are of LibSPN.

B. Semantic Place Categorization

First, we evaluated DGSM for semantic place categorization and compared it to a well-established model based on an SVM and geometric features [12], [13]. The features were extracted from laser scans raytraced in the same local Cartesian grid maps used to form polar grids for DGSM. We raytraced the scans in high-resolution maps (2cm/pixel), to obtain 362 beams around the robot. To ensure the best SVM result, we used an RBF kernel and selected the kernel and learning parameters directly on the test sets.

The models were trained on the four room categories and evaluated on observations collected in places belonging to the same category, but on different floors. The normalized confusion matrices are shown in Fig. 4. We can see that DGSM obtains superior results for all classes. The classification rate averaged over all classes (giving equal importance to each class) and data splits is $85.9\% \pm 5.4$ for SVM and $92.7\% \pm 6.2$ for DGSM, with DGSM outperforming SVM for every split. Most of the confusion exists between the small and large office classes. Offices in the dataset often have complex geometry that varies greatly between room instances.

C. Novelty Detection

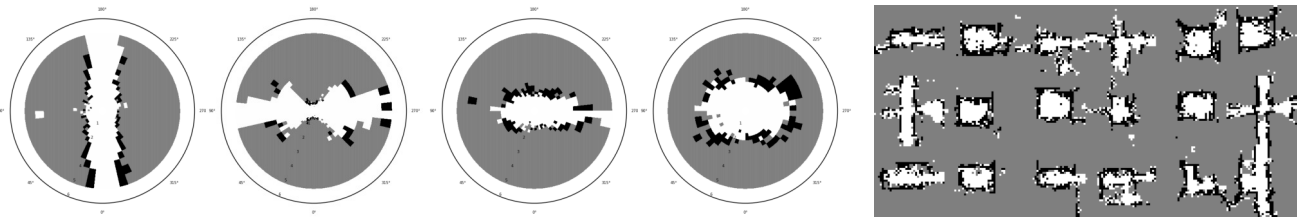
The second experiment, evaluated the quality of the uncertainty measure produced by DGSM and its applicability to detecting outliers from room categories not known during training. We used the same DGSM model and compared the marginal probability for samples in the test set and from novel categories. The cumulative ROC curve over all data splits is shown in Fig. 5.

We compared to a one-class SVM with an RBF kernel trained on the geometric features. The ν parameter was adjusted to obtain the largest area under the curve (AUC) on the test sets. We can observe that DGSM offers a significantly more reliable novelty signal, with AUC of 0.81 compared to 0.76 for SVM. This result is significant, since to the best of our knowledge, it demonstrates for the first time the usefulness of SPN-based models for novelty detection.

D. Generating Observations of Places

In this qualitative experiment, our aim was to assess properties of the generative models by examining generated place appearances. For DGSM, we conditioned on the semantic class variable and inferred the MPE state of the observation variables. The generated polar occupancy grids can be seen as prototypical appearances of places from each semantic

Preprint submitted to 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. Received March 1, 2017.



(a) Corridor (b) Doorway (c) Small Office (d) Large Office (e) GAN Samples

Fig. 6: Results of MPE inference over place appearances conditioned on each semantic category for DGSM ((a)-(d)); and place appearance samples generated using GAN (e).

category and are illustrated in Fig. 6a-d. For GANs, we plot samples generated for random values of the noise variables $\boldsymbol{z}$ in Fig. 6e.

We can compare the plots to the true examples depicted in Fig. 2. We can see that each polar grid is very characteristic of the class from which it was generated. The corridor is an elongated structure with walls on either side, and the doorway is depicted as a narrow structure with empty space on both sides. Despite the fact that, as shown in Fig. 2, large variability exists between the instances of offices within the same category, the generated observations of small and large offices clearly indicate a distinctive size and shape.

While the GAN architecture used for predicting missing observations in unlabeled samples cannot generate samples conditional on semantic category, we still clearly see examples of different room classes, including intra-class variations.

E. Predicting Missing Observations

Our final experiment provides a quantitative evaluation of the generative models on the problem of reconstructing missing values in partial observations of places. To this end, we masked a random chunk of 25% of the grid cells for each test sample in the dataset. In case of DGSM, we masked a random 90-degree view, which corresponds to a rectangular mask in polar coordinates. For GANs, since we use a Cartesian grid, we used a square mask in a random position around the edges of the grid map[4]. For DGSM, all indicators for the masked polar cells were set to 1 to indicate missing evidence and MPE inference followed. For GANs, we used the approach in Sec. V-A.

Fig. 7 shows examples of grids filled with predicted occupancy information to replace the missing values for both models. While the predictions are often consistent with the true values, both models do make mistakes. Analyzing the DGSM results more closely, we see that this typically occurs when the mask removes information distinctive of the place category. Interestingly, in some cases, the unmasked input grid might itself be partial due to missing observations during laser range data acquisition. When the missing observations coincide with a mask, the model will attempt to reconstruct them. Such example can be seen for a polar grid captured in a corridor shown in the bottom left corner of Fig. 7.

[4]We considered polar masks on top of Cartesian grid maps. However, this provided a significant advantage to GANs, since most of the masked pixels laid far from the robot, often outside a room, where they are easy to predict.

Overall, when averaged over all the test samples and data splits, the DGSM model correctly reconstructs $77.14\% \pm 1.04$ of masked cells, while the GAN model recovers $75.84\% \pm 1.51$. This demonstrates that the models have comparable generative potential, confirming state-of-the-art performance of DGSM.

F. Discussion and Model Comparison

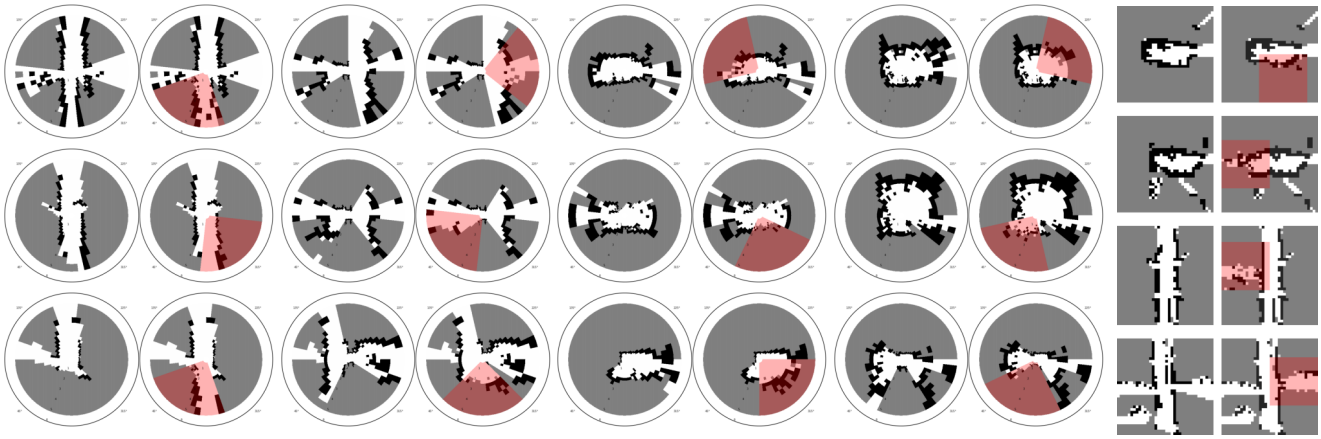
The experiments clearly demonstrate the potential of DGSM. Its generative abilities match (and potentially surpass) those of GANs on our problem, while being significantly more computationally efficient. DGSM naturally represents missing evidence and requires only a single upwards and downwards pass through the network to infer missing observations, while GANs required hundreds of iterations, each propagating gradients through the network. Additionally, DGSM in our experiments used a smaller network than GANs, requiring roughly a quarter of sum and product operations to compute a single pass, without the need for any nonlinearities. This property makes DGSM specifically well suited for real-time robotics applications.

DC-GANs, being a convolutional model, lend themselves to very efficient implementations on GPUs. DGSM uses a more complicated network structure. However, in our current implementation in LibSPN, DGSM is real-time during inference and very efficient during learning, obtaining much faster inference than GANs. As a result, extending the model to include additional modalities and capture visual appearance as well as 3D structure seems computationally feasible with DGSM.

The experiments with different inference types were all performed on the same model after a single training phase (separately for each dataset split). This demonstrates that our model spans not only multiple levels of abstraction, but also multiple tasks and applications. In contrast, SVMs and GANs were optimized to solve a specific task. In particular, the model retains high capability to discriminate, while being trained generatively to represent a joint distribution over low-level observations. Yet, as demonstrated in the novelty detection experiments, it produces a useful uncertainty signal in the form of marginal probability. Neither GANs nor SVMs explicitly represent marginal probability of the data.

VII. CONCLUSIONS AND FUTURE WORK

This paper presents DGSM, a unique generative spatial model, which to our knowledge, is the first application of



(a) Corridor (b) Doorway (c) Small Office (d) Large Office (e) GAN

Fig. 7: Examples of successful and unsuccessful completions of place observations with missing data: grouped by true semantic category for DGSM ((a)-(d)) and for GAN (e). For each example, a pair of grids is shown, with the true complete grid on the left, and the inferred missing data on the right. The part of the grid that was masked and inferred is highlighted.

sum-product networks to the domain of robotics. Our results demonstrate that DGSM provides an efficient framework for learning deep probabilistic representations of robotic environments, spanning low-level features, geometry, and semantic representations. We have shown that DGSM can be used to solve a variety of important robotic tasks, from semantic classification of places and uncertainty estimation to novelty detection and generation of place appearances based on semantic information. DGSM has appealing properties and offers state-of-the-art performance. While our results were based on laser range data, the approach is readily applicable to learning rich hierarchical representations from RGBD or 2D visual data. Our future efforts will explore such learning of robot environments as well as exploit the resulting deep representations for probabilistic reasoning and planning at multiple levels of abstraction.

References

[1] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *T-RO*, vol. 29, no. 4, 2013.

[2] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *ICRA*, 2012.

[3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *PAMI*, vol. 35, no. 8, 2013.

[4] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection." *ISER*, 2016.

[5] N. Sunderhauf, F. Dayoub, S. McMahon, B. Talbot, R. Schulz, P. Corke, G. Wyeth, B. Upcroft, and M. Milford, "Place categorization and semantic mapping on a mobile robot," in *ICRA*, 2016.

[6] R. Goeddel and E. Olson, "Learning semantic place labels from occupancy grids using CNNs," in *IROS*, 2016.

[7] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *UAI*, 2011.

[8] R. Peharz, S. Tschiatschek, F. Pernkopf, and P. Domingos, "On theoretical properties of Sum-product Networks," in *AISTATS*, 2015.

[9] R. Gens and P. Domingos, "Discriminative learning of sum-product networks," in *NIPS*, 2012.

[10] R. Peharz, P. Robert, K. Georg, M. Pejman, and P. Franz, "Modeling speech with Sum-Product Networks: Application to bandwidth extension," in *ICASSP*, 2014.

[11] M. Amer and S. Todorovic, "Sum-Product Networks for activity recognition," *PAMI*, 2015.

[12] O. M. Mozos, C. Stachniss, and W. Burgard, "Supervised learning of places from range data using AdaBoost," in *ICRA*, 2005.

[13] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *IJRR*, vol. 29, no. 2-3, Feb. 2010.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NIPS*, 2014.

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv:1511.06434 [cs.LG], 2015.

[16] A. Oliva and A. Torralba, "Building the gist of a scene: The role of global image features in recognition," *Brain Research*, vol. 155, 2006.

[17] A. Ranganathan, "PLISS: Detecting and labeling places using online change-point detection," in *RSS*, 2010.

[18] P. Buschka and A. Saffiotti, "A virtual sensor for room detection," in *IROS*, 2002.

[19] A. Tapus and R. Siegwart, "Incremental Robot Mapping with Fingerprints of Places," in *ICRA*, 2005.

[20] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view RGB-D object dataset," in *ICRA*, 2011.

[21] A. Eitel, J. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *IROS*, 2015.

[22] M. Madry, L. Bo, D. Kragic, and D. Fox, "ST-HMP: Unsupervised Spatio-Temporal Feature Learning for Tactile Data," in *ICRA*, 2014.

[23] D. Kingma and M. Welling, "Auto-encoding variational bayes," in *ICLR*, 2014.

[24] R. Yeh, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," arXiv:1607.07539 [cs.CV], 2016.

[25] W.-C. Cheng, S. Kok, H. Pham, H. Chieu, and K. Chai, "Language modeling with Sum-Product Networks," in *Interspeech*, 2014.

[26] R. Gens and P. Domingos, "Learning the structure of Sum-Product Networks," in *ICML*, 2013.

[27] A. Rooshenas and D. Lowd, "Learning Sum-Product networks with direct and indirect variable interactions," in *ICML*, 2014.

[28] A. Darwiche, "A differential approach to inference in bayesian networks," *Journal of the ACM*, vol. 50, no. 3, May 2003.

[29] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, 2007.