From Pixels to Buildings: End-to-end Probabilistic Deep Networks for Large-scale Semantic Mapping

Kaiyu Zheng[1] and Andrzej Pronobis[2]

***Abstract*— We introduce TopoNets, end-to-end probabilistic deep networks for modeling semantic maps with structure reflecting the topology of large-scale environments. TopoNets build a unified deep network spanning multiple levels of abstraction and spatial scales, from pixels representing geometry of local places to high-level descriptions of semantics of buildings. To this end, TopoNets leverage complex spatial relations expressed in terms of arbitrary, dynamic graphs. We demonstrate how TopoNets can be used to perform end-to-end semantic mapping from partial sensory observations and noisy topological relations discovered by a robot exploring large-scale office spaces. Thanks to their probabilistic nature and generative properties, TopoNets extend the problem of semantic mapping beyond classification. We show that TopoNets successfully perform uncertain reasoning about yet unexplored space and detect novel and incongruent environment configurations unknown to the robot. Our implementation of TopoNets achieves real-time, tractable and exact inference, which makes these new deep models a promising, practical solution to mobile robot spatial understanding at scale.**

I. INTRODUCTION

The ability to make uncertain inferences about *spatial information* is fundamental for a mobile agent planning and executing actions in large, unstructured environments [1], such as office buildings, airports and search and rescue sites. Robots, while exploring their environments, gather a growing body of knowledge captured at different spatial locations, scales (from places to buildings), and levels of abstraction (from sensory data, through place geometry and appearance, up to high-level semantic descriptions).

While such information is typically incomplete and noisy, it is also structured according to relationships that govern the human world. Discovering and leveraging relationships that span local and global spatial scales as well as multiple levels of abstraction can help improve robustness, resolve ambiguities, and enable predictions about latent and unobserved information [1][2][3]. Unfortunately, such relationships are also complex and noisy, making semantic mapping a difficult structured prediction problem. Additionally, semantic maps are dynamic structures, with dependencies often expressed in terms of graphs containing a different number of nodes and relations for every environment [2].

[1]Kaiyu Zheng is with Computer Science Dept., Brown University, Providence, RI, USA kaiyu_zheng@brown.edu

[2]Andrzej Pronobis is with Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA as well as Robotics, Perception and Learning Lab, KTH, Stockholm, Sweden pronobis@cs.washington.edu. This work was supported by Office of Naval Research (ONR) grant no. N00014-13-1-0817 and Swedish Research Council (VR) project 2012-4907 SKAEENet.

We would like to express our gratitude to Prof. Rajesh P. N. Rao for his unwavering support, encouragement, and invaluable advice.

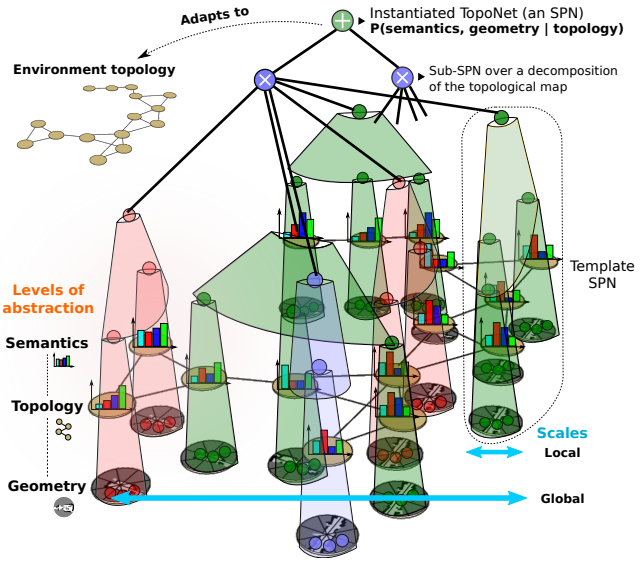

Fig. 1: Illustration of a TopoNet instantiated over a semantic map, with structure adapted to the topology of the environment. It incorporates spatial information across multiple levels of abstraction at both local and global spatial scales, and forms a probability distribution over semantic attributes and geometric representations of places.
Video: http://y2u.be/luv2XpaHeTU.

As a result, most deep approaches to semantic mapping fail to capture and exploit such relations. In particular, approaches utilizing convolutional neural networks focus on relationships constrained to local scenes [4] and require that the number of latent variables be constant and related through a similar global structure [5]. Other approaches compromise on the structure complexity [6], introduce prior structural knowledge [7], or make hard commitments about values of semantic attributes [2]. Additionally, these methods are often assembled from independent spatial models [2][8], which exchange information in a limited fashion.

To overcome these shortcomings, in this work, we present TopoNets, end-to-end deep networks for modeling semantic maps with dynamic structure adapted to topology of large-scale environments. TopoNets leverage the advantages of Sum-Product Networks (SPNs) and once *instantiated*, form a unified model that spans across abstractions and spatial scales, with guaranteed tractable, exact inference (Fig. 1). In our experiments, we evaluated TopoNets via the tasks of semantic place classification, inference of semantics of unexplored places, and detection of novel environment

configurations. In each task, we compare the end-to-end TopoNets with a more traditional model assembled from two components: a Markov Random Field (MRF) representing spatial relations and a deep model capturing place appearance. We show that TopoNets more effectively disambiguate noisy local predictions based on real-world observations, and perform uncertain reasoning about yet unexplored space. Furthurmore, we demonstrate that TopoNets exhibit generative properties useful for novelty detection, and achieve real-time performance while permitting exact probabilistic inference.

II. RELATED WORK

There have been numerous attempts to employ structured prediction to modeling semantic maps with topological spatial relations. Mozos et al. [6] used hidden Markov models (HMMs) to smooth sequences of AdaBoost classifications of place observations into semantic categories. Friedman et al. [7] proposed Voronoi Random Fields (VRFs) which are CRFs constructed according to a Voronoi graph extracted from an occupancy grid map. VRFs utilize pairwise potentials to model dependency between neighboring graph nodes and 4-variable potentials to model junctions. Pronobis and Jensfelt [2] applied Markov Random Fields to model pairwise dependencies between semantic categories of rooms according to a topological map. The categorical variables were connected to Bayesian Networks that reasoned about local environment features, forming a chain graph. This approach relied on a door detector to segment the environment into a topological graph with only one node per room. Overall, while probabilistic, these approaches employ approximate inference, leading to problems with convergence [7]. Moreover, additional prior knowledge or hard commitments about the semantics of some places is required in order to obtain a tractable model. In contrast, in this work, we make no such commitments and rely on topological maps built by a real robot while performing navigation and action execution. At the same time, probabilistic inference with our model remains exact and real-time.

Recently several deep structured prediction methods have been proposed [9][10][11]. Unfortunately, most are designed for computer vision tasks and are not applicable to the problem of modeling spatial relations in large-scale dynamic environments. Notably, Mahmood et al. [9] proposed a feature fusion method for conditional GAN which is conceptually similar to a Conditional Random Field (CRF). The approach does not consider the joint probability distribution of local observations and semantics as we do in this work. Wu et al. [10] proposed a deep variant of MRFs based on multiple recurrent neural networks for vision tasks. However, the method is applicable only to problems with fixed number of variables, while our approach handles graphs of arbitrary size and structure.

TopoNets build upon our previous work, which introduced Sum-Product Networks (SPNs) to the domain of robotics. First, Pronobis et al. [12] established the use of SPNs for local place classifiction via a deep generative architecture that models robot-centric laser range observations. Second, Zheng et al. [13] proposed a general probabilistic approach to structured prediction, named GraphSPN, that extended SPNs to allow for the modeling of arbitrary, dynamic graphs. That work provided a new theoretical framework, yet relied on synthetic local evidence. In this paper, we propose a unified, end-to-end architecture, and experiment with real-world robot data collected in office settings to demonstrate its practical value to the semantic mapping problem.

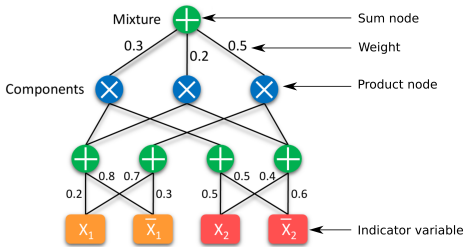

Fig. 2: A simple SPN for a naive Bayes mixture model $P(X_1, X_2)$, with three components over two binary variables. The bottom layer consists of indicators for different values of the variables X_1 and X_2. Weighted sum nodes, with weights attached to inputs, are marked with $+$, while product nodes are marked with $\times$.

III. PRELIMINARIES

We begin by giving a brief introduction to Sum-Product Networks (SPNs), which provide the fundamental theoretical framework for TopoNets. Then, we describe the structure of the semantic maps for which TopoNets are built.

A. Sum-Product Networks

SPNs are deep probabilistic models with solid theoretical foundations [14][15][16] that have been shown to provide state-of-the-art results in several domains [12][16][17][18]. One of the primary limitations of traditional 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 and involve a polynomial number of sum and product operations, permitting exact inference. SPNs combine these advantages with benefits of deep learning by acquiring hierarchical probabilistic models directly from high-dimensional, noisy data. 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, including speech [17] and language modeling [19], human activity recognition [18], image classification [16], image completion [15], and robotics [12].

As shown in Fig. 2, on a simple example of a naive Bayes mixture model, an SPN is a generalized directed acyclic graph composed of weighted sum and product operations. The sums can be seen as mixture models over subsets of variables, with weights representing mixture priors. Products

can be viewed as features or mixture components. Not all possible architectures consisting of sums and products result in valid probability distributions and certain constraints (completeness and decomposability [15][20]) must be followed to guarantee validity.

SPNs model joint or conditional distributions and can be learned generatively [15] or discriminatively [16] using Expectation Maximization (EM) or gradient descent (GD). Additionally, several algorithms were proposed for simultaneous learning of network parameters and structure [21][22][23]. In this work, we use a simple structure learning technique [12] which begins by initializing the SPN with a random dense structure that is later pruned. The approach recursively generates network nodes based on multiple random decompositions of the set of variables into multiple subsets until each subset is a singleton. The resulting structure is a deep network consisting of products combining the subsets in each decomposition and sums mixing different decompositions at each level. SPNs can be defined for both continuous and discrete variables, with evidence for categorical variables often specified in terms of binary indicators.

Inference in SPNs is accomplished by an upwards pass which calculates the probability of the evidence and a downwards pass which obtains gradients for calculating marginals or MPE (Most Probable Explanation) state of the missing evidence. The latter can be obtained by replacing sum operations with weighted max operations (the resulting network is sometimes referred to as Max-Product Network, MPN [16]). For a detailed explanation of SPNs, we refer the reader to [20][16][15].

B. Semantic Maps

In order to represent dynamic spatial relations at the scale of a building, we define semantic maps as *growing* topological graphs of places associated with observations of local geometry as well as semantic descriptions. Examples of the topological and semantic information in such maps acquired by a robot (without local geometries) are shown in Fig. 7. To obtain the representation of local place geometry, as the first step, we perform spatio-temporal integration of the sensory input. We rely on laser-range data, and use a particle-filter grid mapping [24] to maintain a robot-centric map of 5m radius around the robot. The goal of the local representation is to model geometry of a single place. Thus, we constrain the observation of a place to the information visible from the robot (structures that can be raytraced from the robot's location). As a result, walls occlude the view and the local map mostly contains information from a single room.

In our implementation, spatial relationships within each local place are modeled from the perspective of a mobile robot acting at that place. Therefore, in the next step, each local observation is transformed into a robot-centric polar occupancy grid. Examples of such local place representations acquired by a robot can be seen in Fig. 5. The resulting observation contains higher-resolution details closer to the robot and lower-resolution context further away. This 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. In the future, we plan to use a similar strategy when representing 3D and visual information, by extending the polar representation to 3 dimensions.

The topological graph of a complete semantic map is built and updated incrementally while the robot is exploring its environment [25]. The primarily purpose of the graph is to support the behavior of the robot. As a result, nodes in the graph represent *places* the robot can visit and the edges represent both navigability and spatial relations. The places are associated with their local geometry representations and latent variables representing semantics. Additional nodes in the graph, called *placeholders*, are created to represent exploration frontiers. Those frontiers are added at neighboring, reachable, but unexplored locations and connected to existing places. Then, once the robot performs an exploration action, a placeholder is converted into a place, to which a local geometric place representation is anchored.

IV. TOPONETS

TopoNets are deep SPNs that adapt their structure according to the topology and spatial relations in a semantic map. We begin with a formal definition of TopoNets, followed by a description of the learning and inference procedure.

A. Definition

Let us use $\boldsymbol{X}_i$ to denote local observations of place geometry, and $\boldsymbol{Y}_i$ to denote semantic attributes that describe the places. We can specify a semantic map as $M = (T, \boldsymbol{X}, \boldsymbol{Y})$, where $T = (\boldsymbol{V}, \boldsymbol{E})$ is a topological graph with vertices $\boldsymbol{V}$ and edges $\boldsymbol{E}$, and $\boldsymbol{X} = \{\boldsymbol{X}_i : i \in \boldsymbol{V}\}$, $\boldsymbol{Y} = \{\boldsymbol{Y}_i : i \in \boldsymbol{V}\}$.

A TopoNet is not specific to any particular semantic map, but rather a template-based model that can be *instantiated* for certain states of a semantic map to perform inference tasks. To define TopoNets, we start by specifying a set $\boldsymbol{\mathcal{T}} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\}$ of *sub-map templates*, which can be used to decompose a semantic map. We define a *sub-map template* $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ as a graph with $\boldsymbol{\mathcal{X}} = \{\boldsymbol{X}_i : i \in \mathcal{V}\}$ and $\boldsymbol{\mathcal{Y}} = \{\boldsymbol{Y}_i : i \in \mathcal{V}\}$. Such template can be a recurring topological structure in a given dataset of semantic maps. Following [13], we define the resulting decompositon as:

Definition 1: A decomposition of a semantic map $M = (T, \boldsymbol{X}, \boldsymbol{Y})$ using sub-map templates $\boldsymbol{\mathcal{T}}$ is a set of *map parts* $M_k = (T_k, \boldsymbol{X}_k, \boldsymbol{Y}_k)$, with $T_k = (\boldsymbol{V}_k, \boldsymbol{E}_k)$, such that T_k is isomorphic with any $\mathcal{T} \in \boldsymbol{\mathcal{T}}$, $\bigcup_k T_k = T$, $\forall_{k,l} T_k \cap T_l = \varnothing$, and the variables $\boldsymbol{X}_k$ and $\boldsymbol{Y}_k$ correspond to vertices of T_k: $\boldsymbol{X}_k = \{\boldsymbol{X}_i : i \in \boldsymbol{V}_k\}$, $\boldsymbol{Y}_k = \{\boldsymbol{Y}_i : i \in \boldsymbol{V}_k\}$.

With that, we can define TopoNets as a template model consisting of a set of *template SPNs*:

Definition 2: A *template SPN* $\mathcal{S}^{\mathcal{T}}[\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}]$ corresponding to a sub-map template $\mathcal{T}$ is an SPN that models the distribution $P^{\mathcal{T}}(\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}})$.

Definition 3: A TopoNet $\boldsymbol{\mathcal{S}}^{\boldsymbol{\mathcal{T}}}$ is a set of *template SPNs* such that $\boldsymbol{\mathcal{S}}^{\boldsymbol{\mathcal{T}}} = \{\mathcal{S}^{\mathcal{T}}[\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}] : \mathcal{T} \in \boldsymbol{\mathcal{T}}\}$.

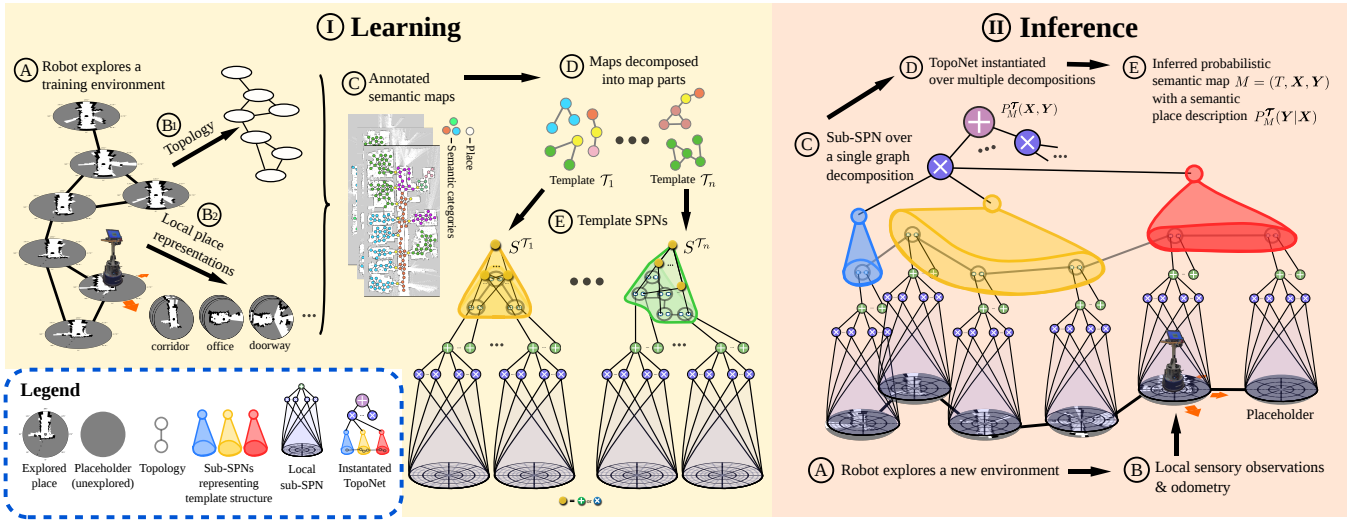

Fig. 3: Learning and inference process with TopoNets. See step-by-step description in Sec. IV-B and Sec. IV-C.

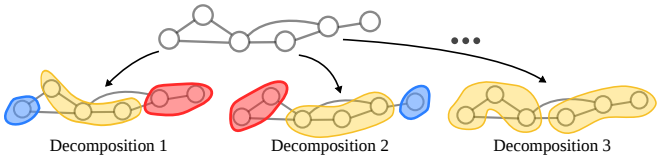

Fig. 4: A topological graph can be decomposed in multiple ways using the same set of sub-map templates.

B. Learning (Fig. 3, I)

As the robot explores a training environment (A), it can incrementally construct a topological graph (B_1) using the approach described in Sec. III-B and obtain local sensory observations at each place (B_2). Eventually, the robot collects a dataset of annotated semantic maps $\boldsymbol{M}_{train}$ (C). We specify a set $\boldsymbol{\mathcal{T}}$ of *sub-map templates*, each corresponding to a *template SPN*. We use $\boldsymbol{\mathcal{T}}$ to decompose $\boldsymbol{M}_{train}$, which leads to a dataset of *map parts* (D), defined in Def. 1. Then, the structure and parameters of each template SPN $\mathcal{S}^{\mathcal{T}}$ are learned using the dataset of matching map parts (E). The specific procedure used to obtain template SPNs in our experiments is described in Sec. V-B.

C. Inference (Fig. 3, II)

When the robot explores a new test environment (A), it constructs a topological graph with each place corresponding to a local geometry representation (B). Next, a trained TopoNet $\boldsymbol{\mathcal{S}}^{\mathcal{T}}$ is adapted to the topological structure of the underlying semantic map $\boldsymbol{M}_{test}(T,\boldsymbol{X},\boldsymbol{Y})$ and latent place semantics is inferred. This adaptation process is the *instantiation* of a TopoNet performed as follows. First, the semantic map is decomposed into *map parts* using sub-map templates $\boldsymbol{\mathcal{T}}$. For each map part $M_k = (T_k, \boldsymbol{X}_k, \boldsymbol{Y}_k)$, a template SPN $\mathcal{S}^{\mathcal{T}} \in \boldsymbol{\mathcal{S}}^{\mathcal{T}}$ is selected such that T_k is isomorphic with $\mathcal{T}$. The structure and weights of $\mathcal{S}^{\mathcal{T}}$ are instantiated as an SPN $\mathcal{S}^{\mathcal{T}}[\boldsymbol{X}_k,\boldsymbol{Y}_k]$ that models the distribution $P^{\mathcal{T}}(\boldsymbol{X}_k,\boldsymbol{Y}_k)$. The instantiated template SPNs for all map parts are combined with a product node, which forms a sub-SPN over a single, complete graph decomposition (C). Using the same $\boldsymbol{\mathcal{T}}$, a topological graph is decomposed in multiple different ways (Fig. 4). After N *different* decomposition attempts, the N product nodes become the children of the root sum node of the final network (D). This forms a distribution $P^{\boldsymbol{\mathcal{T}}}_{\boldsymbol{M}_{test}}(\boldsymbol{X},\boldsymbol{Y})$, which can be seen as a mixture model over the different decompositions.

Once instantiated, TopoNets can perform different types of probabilisitic inferences:

1) Semantic place classification: For all places explored by the robot, where local observations $\boldsymbol{X}$ are available, we can task TopoNets with inferring latent semantics:

$$\hat{\boldsymbol{y}}_{explored} = \underset{\boldsymbol{y}_{explored}}{\operatorname{argmax}} P(\boldsymbol{y}_{explored}|\boldsymbol{x}_{explored}) \tag{1}$$

2) Inferring semantics of unexplored space: We can increase the complexity of the task and infer semantic descriptions of both explored places and nearby unexplored placeholders, for which local evidence is not available:

$$\hat{\boldsymbol{y}}_{explored},\hat{\boldsymbol{y}}_{unexplored} = \underset{\substack{\boldsymbol{y}_{explored};\\ \boldsymbol{y}_{unexplored}}}{\operatorname{argmax}} P(\boldsymbol{y}_{explored},\boldsymbol{y}_{unexplored}|\boldsymbol{x}_{explored}) \tag{2}$$

3) Novelty detection: This inference task evaluates the generative properties of TopoNets. For a certain set of local observations $\boldsymbol{x}_{explored}$, we can evaluate the likelihood $P(\boldsymbol{x}_{explored})$ and use it as a measure of novelty. The likelihood can be thresholded to determine whether the complete environment is within the distribution of environments known during training:

$$\sum_{\boldsymbol{y}_{explored}} P(\boldsymbol{y}_{explored},\boldsymbol{x}_{explored}) > threshold \tag{3}$$

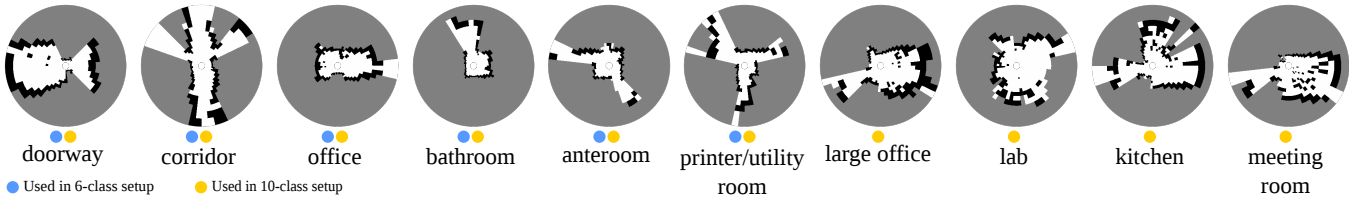

Fig. 5: Examples of local geometry observations for places of different groundtruth categories.

Fig. 6: Sub-map templates used in our experiments.

V. EXPERIMENTAL SETUP

A. Dataset

Our experiments were performed for semantic maps built from laser-range and odometry data from the COLD-Stockholm dataset[1]. The dataset contains 32 data sequences captured using a mobile robot navigating four floors (floors 4-7) of an office building. On each floor, the robot explored rooms of different semantic categories. We experimented with two place category setups, with 6 and 10 classes, shown in Fig. 5. Each class appeared on at least two of the four floors. We used the two setups to illustrate how TopoNets behave in settings of varying difficulty. To ensure variability between training and test sets, we split the dataset four times, each time training TopoNets on data from three floors and leaving one floor out for testing. Note that unlike in [13], where experiments were conducted with synthetic observations, we experimented with real data and more challenging configurations than the 4-class setup in [12].

B. Realization of TopoNets

The TopoNets framework can be adapted to the complexity of the topological maps and the type of sensory input available to the robot. In our experiments, we built TopoNets using a set of three simple sub-map templates shown in Fig. 6. For each sub-map template, the structure and weights of the corresponding template SPN were obtained using the same protocol described as follows.

The structure of each template SPN was partially designed based on domain knowledge and partially learned according to the algorithm described in Sec. III-A. At the bottom of the network, the geometry of each place in the sub-map template was modeled independently, using sub-SPNs resembling the place classification model proposed in [12]. For each place, the local sensory input integrated into a polar occupancy grid was captured using a set of indicator variables. The resolution of the local place geometries (Fig. 5) was 56 angular cells by 21 radial cells, resulting in 1176 random variables per place. Next, we split the polar occupancy grid equally into eight 45-degree views. For each view, we learned an independent sub-SPN. This allowed us to use networks of different complexity for representing low-level features and high-level structure of a place. On top of the sub-SPNs representing the views, we learned a sub-SPN representing a complete place geometry. That process was repeated for each semantic place class, resulting in either 6 or 10 sub-SPNs for each place in the sub-map template. The upper layers of the template SPN combined all the sub-SPNs into a single template distribution. The structure of those layers was initialized to best represent the characteristics of a specific sub-map template and learned as described in Sec. III-A.

The parameters of the model were learned using Gradient Descent via two different losses for different layers of a template SPN. For the bottom sub-SPNs representing features of specific semantic classes, we employed a cross-entropy discriminative loss in order to maximize classification performance. In contrast, the top layers were trained with Maximum-Likelihood generative loss in order to retain good generative abilities and estimation of likelihoods for complete semantic maps. This provided a good trade-off between the different abilities of the probabilistic representation.

C. Baseline

As discussed in Section II, existing deep approaches to structured prediction and semantic mapping could not be directly applied to the task formulated in this paper. Therefore, as a baseline, we used a more traditional model composed of two sub-models. First, similarly to the semantic mapping techniques in [2][26], we used a pairwise Markov Random Field (MRF) to capture dynamic spatial relations in topological graphs. However, since the complexity of the sensory observations requires a different perceptual model, we combined the MRFs with a deep representation capturing the geometry of local places. To this end, we emploed local SPN models, structured identically to the bottom layers of our TopoNets, and trained them discriminatively to infer semantic place categories of independent places based only on local evidence. The resulting model used evidence from local SPNs as unary potentials $\phi_i(Y_i = c) = P(\boldsymbol{X}_i|Y_i = c)$ in the MRF. The pairwise potentials were obtained as in [13] by computing co-occurrence statistics of semantic classes of neighboring places in the training topological graphs.

D. Software and Performance

The experiments with TopoNets were conducted using *LibSPN* [27][2], a library for learning and inference with SPNs

[1] http://coldb.org

[2] https://libspn.org.

A. Semantic Place Classification

Data Split	#classes	Local SPNs		Local SPNs + MRF		TopoNet	
		avg.	std.	avg.	std.	avg.	std.
456-7	6	95.22%	1.70%	96.35%	2.68%	**97.50%**	1.11%
	10	73.48%	1.92%	68.03%	4.55%	**74.69%**	2.73%
457-6	6	96.75%	1.98%	93.87%	2.24%	**97.39%**	1.25%
	10	81.49%	1.93%	81.63%	6.90%	**82.55%**	1.37%
467-5	6	92.70%	1.52%	**95.66%**	3.14%	94.46%	1.62%
	10	73.41%	2.06%	66.63%	5.38%	**74.58%**	2.48%
567-4	6	**99.16%**	0.94%	96.00%	3.21%	98.30%	1.64%
	10	87.88%	2.83%	84.31%	1.56%	**88.73%**	2.35%
Overall	6	95.96%	2.83%	95.47%	3.00%	**96.91%**	2.04%
	10	79.06%	6.45%	75.15%	9.34%	**80.14%**	6.35%

B. Inferring Semantics of Unexplored Space

Data Split	#classes	Local SPNs + MRF		TopoNet	
		avg.	std.	avg.	std.
456-7	6	94.07%	5.65%	**99.46%**	1.44%
	10	57.94%	3.33%	**70.24%**	10.18%
457-6	6	79.77%	6.49%	**83.29%**	4.26%
	10	**61.62%**	9.48%	61.30%	4.83%
467-5	6	94.72%	3.48%	**96.35%**	3.62%
	10	50.50%	5.48%	**54.48%**	3.38%
567-4	6	96.09%	3.50%	**98.75%**	3.31%
	10	71.72%	4.78%	**71.94%**	8.45%
Overall	6	91.16%	8.27%	**94.46%**	7.35%
	10	60.45%	9.84%	**64.49%**	10.11%

TABLE I: Results of the experiments with semantic place classification and inference of semantics of unexplored space.

and TensorFlow, as well as an implementation of Graph-SPNs [13][3]. For MRF experiments, we used an implementation of Loopy Belief Propagation provided by the *libDAI* library [28]. We compared the inference time for TopoNets and the baseline on semantic maps built for 10 classes. TopoNets were built for 40 decompositions of the semantic maps. For maps containing 105 and 155 nodes, TopoNets evaluated $P^{\mathcal{T}}_{\boldsymbol{M}_{test}}(\boldsymbol{X},\boldsymbol{Y})$ in 0.36s and 0.49s respectively, on a desktop computer with one GeForce GTX 1080 Ti GPU. In comparison, inferences with MRF often required more than 45s due to poor convergence and hard-stopping. Note that these run times correspond to inference over the entire semantic map. In practice, inference with TopoNets can be restricted to only those parts of the network affected by new evidence, drastically reducing the amount of required computations.

VI. RESULTS AND DISCUSSIONS

Below, we describe and discuss the results of three experiments corresponding to each of the inference tasks specified in Sec. IV-C.

1) Semantic Place Classification: First, we tasked TopoNets and the baseline employing MRFs and local SPNs with inferring the semantics of explored places given local sensory observations (Sec. IV-C.1). For this experiment, we used an additional baseline consisting of independent local SPNs, inferring semantic descriptions of independent places, without relying on topological spatial relations. The accuracy for each experiment was calculated as the percentage of all places in a test map for which the most likely inferred semantic class matched the groundtruth.

As shown in TABLE I-A, local SPNs obtained overall accuracy (over all test maps and data splits) of 95.96%($\pm$2.83) for the 6-class setup and 79.06%($\pm$6.45) for the 10-class setup. By incorporating the topological spatial relations, TopoNets improved that result to 96.91%($\pm$2.04) and 80.14%($\pm$6.35), respectively. At the same time, the solution employing MRFs for capturing spatial relations resulted in lower performance in both cases: 95.47%($\pm$3.00) and 75.15%($\pm$9.34). This trend was confirmed for most data splits, with TopoNets outperforming each baseline in 7 out of 8 cases, and MRFs lowering the performance compared to local SPNs in 5 out of 8 cases. In-depth analysis of the inference results revealed that the likelihoods of the latent semantic categories calculated independently for local places can be noisy [13]. This significantly impacted the performance of MRFs, while TopoNets remained largely unaffected by noisy local evidence.

As shown in the visualizations of place classification results in Fig. 7a-b, local SPNs provided a strong baseline (particularly in the 6-class setup). However, relying only on local evidence can often lead to perceptual aliasing and misclassification of a cluster of nearby places. That effect was more pronounced in the 10-class setup. In certain cases, the misleading local evidence overpowered the global structural information. This typically occurred in situations where the incorrect, alternative explanation for the local evidence agreed with the global environment structure captured in the training data (e.g. a meeting room misclassified as an office as shown in the highlighted area in Fig. 7b). In such a case, TopoNets could spread the misclassifications to nearby places within the same room. However, in most cases, TopoNets were able to exploit topological spatial relations to correct misclassifications, resulting in improvement in the overall accuracy.

2) Inferring Semantics of Unexplored Space: Next, we tasked TopoNets and the baseline with inference of semantic descriptions of unexplored placeholders lacking local evidence, based solely on observations attached to adjacent explored places (Sec. IV-C.2). Importantly, the semantics of the explored places was not provided and remained latent in this experiment. The accuracy was defined as the percentage of all placeholders in a test map for which the most likely semantic class matched the groundtruth. We report the results in TABLE I-B.

In this experiment, TopoNets outperformed the baseline even more significantly. TopoNets correctly inferred the semantics of 94.46%($\pm$7.35) and 64.49%($\pm$10.11) placeholders in the 6- and 10-class settings, respectively, compared to 91.16%($\pm$8.27) and 60.45%($\pm$9.84) for the baseline. In fact, MRFs coupled with local deep models outperformed TopoNets in only 2 out of 32 sequences in the 6-class setting, and 8 out of 32 sequences in the 10-class setting. As visualized in Fig. 7c-d, TopoNets could successfully exploit the knowledge about global environment structure to distribute evidence to unexplored space.

[3] https://github.com/zkytony/graphspn

Fig. 7: Visualization of TopoNet inference results for four sequences on different floors. The occupancy grid maps and the topological graphs were collected as the robot navigated the environment. The top row shows results for semantic place classification, while the bottom row shows results for inference of unexplored space. The accuracy for the corresponding tasks is shown in the bottom-right corners. Colors indicate the groundtruth or the most likely inferred class.

We observed that increasing the number of decompositions, had a positive influence on the performance of TopoNets on this task (92.43%($\pm$8.08) for 12 decompositions and 94.46% for 40 decompositions in the 6-class setting). Placeholders are exploration frontiers that reside at the outer perimeter of the semantic map, and a larger number of decompositions increases the chance of placeholders being covered by complex sub-map templates.

These results together with the computational efficiency of TopoNets (Sec. V-D) illustrate their practical benefits for spatial understanding in the open world.

3) Novelty Detection: Finally, we exploited the generative properties of the models to determine whether whole environments match the distribution obtained during training or can be considered novel. This property is particularly important for robots operating in open, unknown environments.

For this experiment, we required evidence gathered in novel environments, which were incongruent with the training data. To obtain novel semantic maps, we randomly selected pairs of groundtruth classes, and swapped the local evidence belonging to the two classes for all places in the test maps. For example, the evidence for rooms labeled as an office was swapped with the evidence for bathrooms, effectively creating new environments with different local geometries, where offices now appeared to be bathrooms, and vice versa. We randomized 10 different novel maps for the 6-class setup and 30 for the 10-class setup. At the same time, the original test set provided maps that, while previously unseen, were considered to be within the distribution of the training environments. Note that some of the random swaps generated environment configurations that were similar to those in the training data (e.g. a kitchen swapped with a meeting room), resulting in a difficult detection problem.

The novelty detection results are shown as ROC curves in Fig. 8. Both approaches performed well on this task, with MRFs outperforming TopoNets (average AUC of 0.99 for MRFs and 0.96 for TopoNets in the 6-class case), a similar result to the one reported in [13], despite differences in the setup. From the plots, we see that when taking into account all data splits and test maps, TopoNets were able to correctly

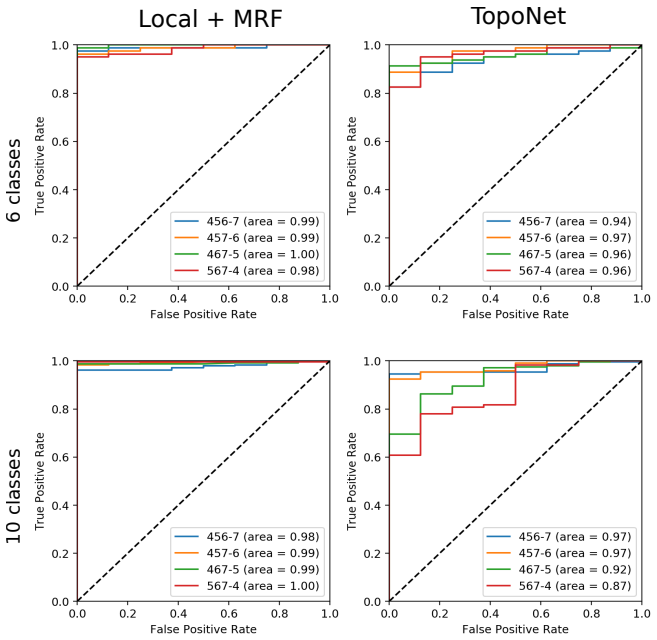

Fig. 8: ROC plots for the task of novelty detection.

detect 89% and 86.25% of novel maps, in 6 and 10-class setups, respectively, while missclassifying as false positives only 9.38% of test maps for the 6-class case and 15.63% for the 10-class case. Such level of performance can be sufficient for many real-world applications, where novelty detection can be used to avoid errors and trigger additional learning.

VII. CONCLUSIONS

This paper presents TopoNets, end-to-end deep networks for modeling semantic maps with structure adapting to dynamic environment topology. Through experiments with real-world robot sensory observations, we comprehensively evaluated and analyzed the inference behavior and generative properties of TopoNets. We demonstrated that TopoNets are an efficient and practical approach to spatial understanding. Furthermore, their properties make them ideal for supporting behavior planning and execution in robots operating in large, open, unknown environments. It is our hope that showcasing the benefits of SPN-based deep models will provide a new direction for research towards novel probabilistic inference techniques in robotics.

REFERENCES

[1] M. Hanheide, M. Göbelbecker, G. S. Horn, A. Pronobis, K. Sjöö, A. Aydemir, P. Jensfelt, C. Gretton, R. Dearden, M. Janicek, *et al.*, "Robot task planning and explanation in open and uncertain worlds," *Artificial Intelligence*, vol. 247, 2017.

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

[3] 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.

[4] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semantic fusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. of ICRA*, 2017.

[5] D. Belanger and A. McCallum, "Structured prediction energy networks," in *Proc. of ICML*, 2016.

[6] O. M. Mozos, R. Triebel, P. Jensfelt, A. Rottmann, and W. Burgard, "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, 2007.

[7] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields : Extracting the topological structure of indoor environments via place labeling," in *Proc. of IJCAI*, 2007.

[8] M. Brucker, M. Durner, R. Ambruş, Z. C. Márton, A. Wendt, P. Jensfelt, K. O. Arras, and R. Triebel, "Semantic labeling of indoor environments from 3d rgb maps," in *Proc. of ICRA*. IEEE, 2018, pp. 1871–1878.

[9] F. Mahmood, W. Xu, N. J. Durr, J. W. Johnson, and A. Yuille, "Structured prediction using cgans with fusion discriminator," *arXiv preprint arXiv:1904.13358*, 2019.

[10] Z. Wu, D. Lin, and X. Tang, "Deep markov random field for image modeling," in *Proc. of ECCV*. Springer, 2016, pp. 295–312.

[11] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2699184

[12] A. Pronobis and R. P. N. Rao, "Learning deep generative spatial models for mobile robots," in *Proc. of IROS*, 2017.

[13] K. Zheng, A. Pronobis, and R. P. N. Rao, "Learning Graph-Structured Sum-Product Networks for probabilistic semantic maps," in *Proc. of AAAI*, 2018.

[14] R. Peharz, R. Gens, F. Pernkopf, and P. Domingos, "On the latent variable interpretation in Sum-Product networks," *Transactions on Pattern Analysis and Machine Intelligence*, 2017.

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

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

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

[18] M. Amer and S. Todorovic, "Sum Product networks for activity recognition," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 4, 2015.

[19] W.-C. Cheng, S. Kok, H. V. Pham, H. L. Chieu, and K. M. A. Chai, "Language modeling with Sum-Product networks," in *Proc. of Interspeech*, 2014.

[20] R. Peharz, S. Tschiatschek, F. Pernkopf, and P. Domingos, "On theoretical properties of sum-product networks," in *Proc. of AISTATS*, 2015.

[21] W. Hsu, A. Kalra, and P. Poupart, "Online structure learning for sum-product networks with gaussian leaves," *preprint arXiv:1701.05265*, 2017.

[22] R. Gens and P. Domingos, "Learning the structure of Sum-product networks," in *Proc. of ICML*, 2013.

[23] R. Peharz, B. C. Geiger, and F. Pernkopf, "Greedy Part-Wise learning of Sum-Product networks," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2013, pp. 612–627.

[24] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *T-RO*, vol. 23, no. 1, 2007.

[25] A. Pronobis, F. Riccio, and R. P. N. Rao, "Deep spatial affordance hierarchy: Spatial knowledge representation for planning in large-scale environments," in *ICAPS 2017 Workshop on Planning and Robotics*, 2017.

[26] I. Posner, M. Cummins, and P. Newman, "A generative framework for fast urban labeling using spatial and temporal context," *Autonomous Robots*, vol. 26, no. 2-3, pp. 153–170, 2009.

[27] A. Pronobis, A. Ranganath, and R. P. N. Rao, "LibSPN: A library for learning and inference with Sum-Product Networks and TensorFlow," in *ICML 2017 Workshop on Principled Approaches to Deep Learning*, 2017.

[28] J. M. Mooij, "libDAI: A free and open source c++ library for discrete approximate inference in graphical models," *Journal of Machine Learning Research*, vol. 11, 2010.