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 this new class of deep models a promising practical solution to comprehensive spatial understanding for mobile robots.

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]. 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 [2], [3], [1]. Unfortunately, such relationships are also complex and noisy, making 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.

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 based on assembling 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 merit of Sum-Product Networks (SPNs) and once instantiated, form a unified model that spans across abstractions and spatial scales, with guaranteed tractable exact inference. 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 TopoNets with an assembled network constructed by combining a Markov Random Field (MRF) with sub-SPNs extracted from TopoNets for local place classification. We show that TopoNets successfully disambiguate noisy predictions based on only local observations and perform uncertain reasoning about yet unexplored space in comparison with the baseline, and that TopoNets exhibit useful generative properties in novelty detection. Furthermore, our implementation of TopoNets using LibSPN [11] achieve real-time tractable and exact inference.

II. RELATED WORK

There have been numerous attempts to employ structured prediction to modeling semantic maps with topological spatial relations. In [6], Hidden Markov Models were used to smooth sequences of AdaBoost classifications of place observations into semantic categories. [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. In [2], Markov Random Fields were used 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. While these approaches are probabilistic, they

1Kaiyu Zheng is with Computer Science Dept., Brown University, Providence, RI, USA kaiyu.zheng@brown.edu.
2Andrzej 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.
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.

We should note that our implementation of TopoNets is based on two of our previous work. First, Pronobis et. al.[12] established the use of SPN for local place classification 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 allows the modeling of arbitrary, dynamic graphs. Since their experiments in [13] used synthetic noise of local evidence, it was unclear whether this method holds up to real-world robot observations. However, in this paper, by proposing a unified semantic mapping architecture, we experiment with robot data collected in real-world office settings and demonstrate the practical value of our approach to the semantic mapping problem.

Fig. 1: 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 $\times$, 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 [16], [17], [18], [12]. 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. 1 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 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 are shown in Fig. 1 (without local geometries). 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. Examples of such local place representations can be seen in Fig. 4.

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. The resulting observation contains high-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, where local place representation is anchored.

IV. TOPONETS

TopoNets are SPNs that adapt their structure according to the topology and spatial relations in a semantic map. We begin by describing the learning and inference procedure as illustrated in Fig. 4 and 5, followed by the formal definition.

A. Learning (I)

As the robot explores a training environment (A), it can incrementally construct a topological graph (B₁) using the approach described in [23] and obtain local sensory observations at each place (B₂). Eventually, the robot collects a dataset of annotated semantic maps \( M_{\text{train}} \) (C). We define a set \( \mathcal{T} \) of sub-map templates, each corresponding to a template SPN. We use \( \mathcal{T} \) to decompose \( M_{\text{train}} \), which leads to a dataset of map parts, defined in Def. [11] (D). Then, the structure and parameters of the template SPNs is learned using the dataset of matching map parts (E).

B. Inference (II)

When the robot explores a new environment (A), it constructs a topological graph with each place corresponding to a local geometry representation (B). Next, a trained TopoNet \( \mathcal{S}^T \) is adapted to the topological structure of the underlying semantic map \( M \) and latent place semantics is inferred. This adaptation process is the instantiation of a TopoNet (C-D). First, we decompose the topological graph into map parts using \( \mathcal{T} \). For each part, the corresponding template SPN is instantiated to model places and spatial relations associated with the part. This results in multiple sub-SPNs sharing weights and structure. The resulting sub-SPNs for a single graph decomposition are combined with a product node (C). After \( N \) different decomposition attempts, we obtain \( N \) product nodes of all decompositions, which then become the children of the root sum node of the complete network (D). This forms a complete distribution \( P^T_M(\mathbf{X}, \mathbf{Y}) \), which can be seen as a mixture model over the different decompositions.

Once instantiated, TopoNets can perform different types of inferences just as a regular SPN does (see Sec. III-A). For example, by considering the local geometry observations \( \mathbf{X} \), TopoNets can infer all the latent semantics \( \mathbf{Y} \) (E). In addition to this, in our experiments, we also evaluate the behavior of TopoNets by inferring the semantics of places which were not explored (i.e. no local evidence \( \mathbf{X}_i \)) and by the task of novelty detection.

C. Definition

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

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 given semantic map to perform inference tasks. To define TopoNets, we start by specifying a set \( \mathcal{T} = \{\mathcal{T}_1, \cdots, \mathcal{T}_n\} \) of sub-map templates, which can be used to decompose a semantic map. We define a sub-map template \( \mathcal{T} = (\mathbf{V}, \mathcal{E}) \) as a graph with \( \mathbf{X} = \{\mathbf{X}_i : i \in \mathbf{V}\} \) and \( \mathbf{Y} = \{\mathbf{Y}_i : i \in \mathbf{V}\} \). 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 \( M = (T, X, Y) \) using sub-map templates \( T \) is a set of map parts \( M_k = (T_k, X_k, Y_k) \), with \( T_k = (V_k, E_k) \), such that \( T_k \) is isomorphic with any \( T \in T \), \( \bigcup_k T_k = T \), and the variables \( X_k \) and \( Y_k \) correspond to vertices of \( T_k \):

\[
X_k = \{X_i : i \in V_k\}, \quad Y_k = \{Y_i : i \in V_k\}.
\]

Next, we define template SPNs and TopoNets:

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

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

**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 40 data sequences captured using a mobile robot navigating with constant speed throughout different floors (floors 4-7) of an office building. On each floor, the robot navigates through rooms of different semantic categories. We experimented with two place category setups, one with 6 classes, and the other with 10 classes, as shown in Fig. 4. All classes appear at least on two out of the four floors. Our 10-class configuration is consistent with

\(^1\)http://coldb.org
previous work [13], where “utility room” includes “printer room”. The 6-class configuration is a simpler configuration with less granularity, where “large office” is also considered “office”, which reduces the ambiguity of local geometry observations but still exhibits a variety of room categories. We use this configuration, to show the quality of inference with TopoNets and how TopoNets behave in settings of varying difficulty. To ensure variability between the training and test sets, we split the data samples 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 local observations, here we experiment with real-robot data, and our class configurations are more challenging than the 4-class setup in [12]. The resolution of local geometry observations is 56 angular cells by 21 radial cells, resulting in 1176 occupancy cells.

### B. TopoNet Realization

To build TopoNets, we specified a set of 3 simple sub-map templates as shown in Fig. 5. For each sub-map template $T$ as defined in Sec. IV-C, the corresponding template SPN $S^T$ is structured as follows (see Fig. 2). The bottom layers consist of sub-SPNs that model $P^T(X_i | Y_i = k)$ where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$. The top layers belong to a sub-SPN that models $P^T(\mathcal{X}, \mathcal{Y})$ whose structure adapts to the underlying sub-map template.

The structure of the sub-SPNs at bottom layers is partially designed based on domain knowledge and partially learned [12]. We begin by splitting the polar grid equally into 8 45-degree views. For each view, we learn an independent sub-SPN. Then, on top of all the sub-SPNs representing the views, we learn a sub-SPN representing complete place geometries for each place class. We utilized the structure learning algorithms mentioned in Sec. III-A. Sub-dividing the representation into views allows us to use networks of different complexity for representing lower-level view features and high-level structure of a place.

The parameters were learned using Gradient Descent via two different losses for different layers of a template SPN. For sub-SPNs at bottom layers, we employed a cross-entropy discriminative loss in order to maximize the semantic classification performance, while the top sub-SPN 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. Tasks

We evaluate the behavior of our realization of TopoNets with three tasks. As the robot explores the test floor, there are explored places and unexplored placeholders. Therefore, for each test sequence, we can consider the underlying semantic map as $M_{test} = (T, X, Y)$ where $Y = Y_{\text{place}} \cup Y_{\text{placeholder}}$. Based on this understanding, we describe the tasks below:

1) **Semantic place classification:** Consider only places that the robot has explored and for which it obtained local observations, and leave out the unexplored places and their edge connections. Then, we use TopoNets to perform the following inference task for each $M_{test} = (T, X, Y_{\text{place}})$:

$$\hat{y} = \arg\max_{Y_{\text{place}}} P(Y_{\text{place}} | X)$$

(1)

Accuracy was measured by counting the total number of correctly inferred semantic categories of places over all places on the testing floor sequences.

2) **Inferring semantics of unexplored places:** Here, we task TopoNets to infer both $Y_{\text{place}}$ and $Y_{\text{placeholder}}$, even though we focus on the accuracy of $Y_{\text{placeholder}}$ in evaluation. Thus, for each $M_{test}$, this task is given by:

$$\hat{y}_{\text{placeholder}} = \arg\max_{Y_{\text{placeholder}}} P(Y_{\text{place}}, Y_{\text{placeholder}} | X)$$

(2)

Accuracy was measured by counting the total number of correctly inferred semantic categories of placeholders over all placeholders on the testing floor sequences.

3) **Novelty detection:** We then evaluate the generative properties of TopoNets. Consider only $Y_{\text{place}}$. Given a semantic map $\tilde{M} = (T, X, Y_{\text{place}})$, for each place $i$, there is a set of associated local SPNs at the bottom of the instantiated TopoNet (see Sec. IV-B). Since each models $P(X_i | Y_i = k)$ at place $i$, these networks produce an array of likelihoods for all semantic classes, denoted as $p_i = (p_i(k_1), \cdots, p_i(k_m))$. We randomly choose several pairs of classes $(k_1, k_2)$, specifically 10 pairs for 6-class setting and 30 pairs for 10-class setting, and swap their corresponding likelihoods in $p_i$ so that $\tilde{p}_{i,k_1} = p_{i,k_2}$ and $\tilde{p}_{i,k_2} = p_{i,k_1}$. This results in a semantic map that is not in the distribution of the training semantic maps, which TopoNets has never seen during learning. We collect the likelihoods of the complete TopoNet distribution
$P^T_M(X, Y_{place} | p)$ for test semantic maps that match the distribution of the training maps, as well as $P^T_M(X, Y_{place} | p)$ for the generated novel semantic maps for every test sequence.

D. Baseline: Local SPNs + Markov Random Fields

Due to the scarcity of recent general-purpose structured prediction approaches applicable to our problem (refer to Sec. II), we compared TopoNets to an assembled network. For this network, we first extract sub-SPNs in template networks that model local observation (referred to as local SPNs). They are assembled to pairwise Markov-Random Fields (MRFs), a standard graphical model method whose variants have been used in the semantic mapping literature [2][26]. We include local SPNs since MRFs are not capable of handling the complexity of raw sensory observations. In this assembled model, the local evidence is given by potential \( \phi(Y_i = k) = P(X_i | Y_i = k) \), and the pairwise potentials are given by, as in [13], generating co-occurrence statistics of variable values in the training graphs used for learning TopoNets.

E. Software and Performance

The experiments for TopoNets were conducted using LibSPN [27], a new library for learning and inference with SPNs and TensorFlow. For MRF experiments, we used Loopy BP inference implemented in the libDAI library [28]. We measured the inference time for TopoNets on semantic maps built for 10 classes compared to the baseline. The TopoNets were instantiated over 40 decomposition attempts of the semantic maps. We noticed that with a semantic map of 105 nodes and one of 155 nodes, the TopoNets evaluate $P^T_M(X, Y)$ (i.e. one upward pass of the network) in worse case 0.36s, 0.49s respectively. However, for MRF inferences, the worst case run time took over 45s due to hard-stopping the non-converging Loopy BP. Note that this is the run time for inference on the entire semantic map, which is not always necessary; we can evaluate only parts of a network when the semantic map expands.

VI. RESULTS AND DISCUSSIONS

Overall, TopoNets reliably outperform the baseline of local SPNs assembled with MRF. We describe and discuss the results for each task below. The quantitative results of semantic place classification and placeholder inference are shown in TABLE 1, while qualitative results are visualized in Fig. 7. The results of novelty detection is shown as ROC plots in Fig. 6.

1) Semantic Place Classification: Local SPNs which infer place semantics based on only local sensory observations obtain overall accuracies of 95.96% (±2.83) and 79.06% (±6.45) in the 6-class and the 10-class scenarios, respectively. TopoNets further improve these results by a small margin to 96.91% (±2.04) and 80.14% (±6.35), while the assembled network deteriorates the accuracies in both cases down to 95.47% (±3.00) and 75.15% (±9.34).

The results for each split indicate that even though the scale of improvement varies, TopoNets are more consistent and reliable in terms of disambiguating noisy local predictions than the baseline. The baseline approach decreases the overall accuracy in 5 out of the 8 trials. We investigated and observed that the likelihoods produced by local SPNs were noisy. Specifically, incorrect predictions are assigned with relatively high confidence, typically over 80%. This overconfidence causes MRFs to deteriorate overall accuracy due to their sensitivity to noise in local classification likelihoods [13], which in turn reflects the robustness of TopoNets against significant noise.

From the qualitative visualizations in Fig. 7, indeed, local SPNs provide strong classification results, more evidently in the 6-class setting. Nevertheless, in the 10-class setting, they sometimes misclassify the majority of places in a room, for example, large office, meeting room, and utility room in the visualization. The reason attributes to the similarity of certain local place geometries of these semantic categories with others on the training floors. This causes both TopoNets and the baseline to only be able to improve the classification accuracy by a small margin, since some correction attempts do not improve the accuracy value (for example, the attempt of classifying a meeting room node as an office node in Fig. 7b). This behavior of falsely overwriting some classifications made by the local SPNs, though seemingly reasonable, leads to moderate improvement in 6-class trials as well (for example, the expansion of corridor nodes in Fig. 7a). From these visualizations, we can also observe that TopoNets tend to smoothen the semantic classifications and classify clusters of places with the same category, reflecting the learning of human space characteristics.

2) Inferring semantics of unexplored places: The benefit of TopoNets are seen more evidently in the placeholder inference results, where TopoNets correctly infer the semantics of 94.46% (±7.35) and 64.49% (±10.11) in 6-class and 10-class settings, respectively, compared to the accuracy of 91.16% (±8.27) and 60.45% (±9.84) for the baseline. Additionally, we observed from experiments that as we increase the number of decomposition attempts, the performance of TopoNets improves, for example, from 92.43% (±8.08) with 12 decompositions, to around 94.46% with 40, in the 6-class setting. This is because placeholders are exploration frontiers that typically reside at the boundaries of the topological graph, and more decomposition attempts lead to a higher chance of these places being covered by any complex sub-map templates (i.e. the 3-node sub-map template in our case).

In both 6-class and 10-class settings, TopoNets outperform the competing assembled model by a significant margin in almost every split. The visualizations in Fig. 7 shows that TopoNets typically extend the semantics of nearby places to the placeholders. We also observed in some test sequences (particularly in 10-class settings) that some placeholders are classified by TopoNets as doorways, with a close-to-uniform underlying distribution of the class likelihoods, indicating a lack of coverage by more complex templates. Still, in combination with our performance analysis in Sec. V-E these
results showcase the practical value of TopoNets on mobile robot exploration.

3) Novelty detection: We thresholded the obtained likelihoods to decide whether a graph is likely to be generated from the distribution of training semantic maps, with results shown in Fig. 6. Both methods perform well on this task, with MRFs performing better than TopoNets, a similar outcome as in [13], despite the differences in the setup. We observed that MRFs-based model produce much higher likelihood for $P_T(X|Y_{\text{place}}(p))$, separating them far from $P_M^T(X,Y_{\text{place}}(p))$, while this is not is not the case for TopoNets. This also explains why MRFs are more sensitive to noise in local predictions since such noise tends to result in a semantic map which MRFs do not recognize. In addition, we noticed numerous cases where MRF inferences fail to converge, whereas this never happened for TopoNets. In conclusion, TopoNets are still a solid model for a real-world application.

VII. CONCLUSION

We have presented TopoNets, end-to-end deep networks for modeling semantic maps with structure adaptable to the environment topology. Through experiments with robot sensory observations, we comprehensively evaluated and analyzed the inference behavior and generative properties of TopoNets. We hope the results in this work provide a new direction of research for community towards using a unified probabilistic approach for semantic mapping.

REFERENCES

[1] M. Hanheide, M. Göbelbecker, G. S. Horn, A. Pronobis, K. Sjö, A. Aydemir, P. Jensfelt, C. Gretton, R. Dearden, M. Janieck, 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 Proceedings of the International Conference on Robotics and Automation (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 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017.

[5] D. Belanger and A. McCallum, “Structured prediction energy networks,” in Proceedings of the International Conference on Machine Learning (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, “Voronoï random fields: Extracting the topological structure of indoor environments via place labeling,” in Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 2007.

[8] M. Amer and S. Todorovic, “Sum product networks for activity recognition,” in Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018.

[9] Z. Wu, D. Lin, and X. Tang, “Deep markov random field for image modeling,” in European Conference on Computer Vision. Springer, 2016, pp. 295–312.

[10] L.-C. 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 transactions on pattern analysis and machine intelligence, vol. 40, no. 4, pp. 834–848, 2018.

[11] A. Pronobis and R. P. N. Rao, “Learning deep generative spatial models for mobile robots,” in Proceedings of the International Conference on Intelligent Robots and Systems (IROS), 2017.

[12] K. Zheng, A. Pronobis, and R. P. N. Rao, “Learning Graph-Structured Sum-Product Networks for probabilistic semantic maps,” in Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018.

[13] 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.

[14] H. Poon and P. Domingos, “Sum-product networks: A new deep architecture,” in Proceedings of the Conference on Uncertainty in Artificial Intelligence (UAI), 2011.

[15] R. Gens and P. Domingos, “Discriminative learning of sum-product networks,” in Advances in Neural Information Processing Systems (NIPS), 2012.

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

[17] M. Amer and S. Todorovic, “Sum product networks for activity recognition,” Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 4, 2015.

[18] W.-C. Cheng, S. Kok, H. V. Pham, H. L. Chieu, and K. M. A. Chai, “Language modeling with Sum-Product networks,” in Proceedings of Interspeech, 2014.

[19] R. Peharz, S. Tüechtschek, F. Pernkopf, and P. Domingos, “On theoretical properties of sum-product networks,” in Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS), 2015.

[20] W. Hsu, A. Kafra, and P. Poupart, “Online structure learning for sum-product networks with gaussian leaves,” preprint arXiv:1701.05265, 2017.

[21] R. Gens and P. Domingos, “Learning the structure of sum-product networks,” in Proceedings of the International Conference on Machine Learning (ICML), 2013.

[22] 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.
Fig. 7: Visualization of TopoNet inference experiments with 4 sequences on different floors (a: floor 5, b: floor 6, c: floor 7, d: floor 4). The occupancy grid map and topological graphs are collected as the robot navigates throughout the floors. The top row shows results for semantic classification, and the bottom row for inference of placeholders. The accuracy for the corresponding tasks is shown at bottom-right corner of the appropriate visualizations. Colors indicate the most likely class in the inferred likelihoods. Local SPNs are extracted from bottom layers of instantiated TopoNets.

| Semantic Place Classification | #classes | Local avg. | Local std. | Local + MRF avg. | Local + MRF std. | TopoNet avg. | TopoNet std. |
|-----------------------------|---------|-----------|-----------|------------------|-----------------|--------------|--------------|
| 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%        |
| 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%     | 98.00%           | 3.21%           | 98.30%       | 1.64%        |
|                            | 10      | 87.88%    | 2.83%     | 84.31%           | 1.56%           | 88.73%       | 2.35%        |

| Placeholder Inference       | #classes | Local + MRF avg. | Local + MRF std. | TopoNet avg. | TopoNet std. |
|-----------------------------|---------|------------------|------------------|--------------|--------------|
| Overall                     | 6       | 91.16%           | 8.27%            | 94.46%       | 7.35%        |
|                            | 10      | 60.45%           | 9.84%            | 64.49%       | 10.11%       |
| 456-7                       | 6       | 94.07%           | 5.65%            | 99.46%       | 1.44%        |
|                            | 10      | 57.94%           | 3.33%            | 70.34%       | 10.18%       |
| 457-6                       | 6       | 79.77%           | 6.49%            | 83.39%       | 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%        |

TABLE I: Results of the experiments with semantic place classification and placeholder and placeholder inference task.

[24] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *Transactions on Robotics*, 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.