Learning Semantic Maps with Topological Spatial Relations Using Graph-Structured Sum-Product Networks

Kaiyu Zheng, Andrzej Pronobis and Rajesh P. N. Rao

***Abstract*— We introduce Graph-Structured Sum-Product Networks (GraphSPNs), a probabilistic approach to structured prediction for problems where dependencies between latent variables are expressed in terms of arbitrary, dynamic graphs. While many approaches to structured prediction place strict constraints on the interactions between inferred variables, many real-world problems can be only characterized using complex graph structures of varying size, often contaminated with noise when obtained from real data. Here, we focus on one such problem in the domain of robotics. We demonstrate how GraphSPNs can be used to bolster inference about semantic, conceptual place descriptions using noisy topological relations discovered by a robot exploring large-scale office spaces. Through experiments, we show that GraphSPNs consistently outperform the traditional approach based on undirected graphical models, successfully disambiguating information in global semantic maps built from uncertain, noisy local evidence. Further, we exploit the probabilistic nature of the model to infer marginal distributions over semantic descriptions of as yet unexplored places.**

I. Introduction

It is essential for a mobile robot to maintain a representation of spatial knowledge, a framework that organizes the understanding of the environment. Mobile robots have access to information at both local and global scale. Therefore, it is desirable for a representation to enable integration of knowledge across spatial scales and levels of abstraction with the help of discovered spatial relations. Topological maps are an established framework for representing spatial relations between local places that enables anchoring high-level conceptual information and easy access for a planning algorithm. As a result, several semantic mapping approaches rely on topological graphs as part of their representation and associate topological nodes with semantic place attributes [1], [2], [3].

In order to integrate the collected spatial knowledge, resolve ambiguities, and make predictions about unobserved places, such frameworks often employ structured prediction algorithms. Unfortunately, the relations discovered by a robot exploring a real-world environment tend to be complex and noisy, resulting in difficult inference problems. Topological maps are dynamic structures, growing as the robot explores its environment, and containing a different number of nodes and relations for every environment. At the same time, many approaches to structured prediction place strict constraints on the interactions between inferred variables to achieve tractability [4] and require that the number of output latent variables be constant and related through a similar global structure [5]. This makes them either inapplicable or impractical in robotics settings and require compromising on the structure complexity [6], introducing prior structural knowledge [1] or making hard commitments about values of semantic attributes [3]. These problems are not unique to robotics and often present themselves in other domains, such as computer vision [7].

In this paper, we present Graph-Structured Sum-Product Networks (GraphSPNs), a general probabilistic framework for modeling graph-structured data with complex, noisy dependencies between a varying number of latent variables. Our framework builds on Sum-Product Networks (SPNs) [8], a probabilistic deep architecture with solid theoretical foundations [9]. SPNs can learn probabilistic models capable of representing context-specific independence directly from high-dimensional data and perform fast, tractable inference on high-treewidth models. GraphSPNs learn template SPN models representing distributions over attributes of sub-graphs of arbitrary complexity. Then, to perform inference for a specific, potentially expanding graph, they assemble a mixture model over multiple decompositions of the graph into sub-graphs, with each sub-graph modeled by an instantiation of an appropriate template.

We apply GraphSPNs to the problem of modeling large-scale, global semantic maps with noisy topological spatial relations built by robots exploring multiple office environments. We make no assumptions about the structure of the topological map that would simplify the inference over semantic attributes. Our approach is capable of disambiguating uncertain and noisy local information about semantic attributes of places as well as inferring distributions over semantic attributes for yet unexplored places, for which local evidence is not available. We compare the performance of our model with the traditional approach based on Probabilistic Graphical Models assembled according to the structure of the topological graph. We show that GraphSPNs significantly outperforms Markov Random Fields built from pairwise and higher-order potentials relying on the same uncertain evidence.

II. Related Work

Probabilistic graphical models (PGMs) [10] provide a flexible framework for structured prediction. Unfortunately,

The authors are with Paul G. Allen School of Computer Science & Engineering, University of Washington, Seattle, WA, USA. A. Pronobis is also with Robotics, Perception and Learning Lab, KTH, Stockholm, Sweden. {kaiyuzh,pronobis,rao}@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.

inference in PGMs is generally intractable, with the exception of low treewidth models [4]. In practice, PGMs often require approximate inference techniques with no guarantee of convergence, such as Loopy Belief Propagation (BP) [11], when the graph structure involves loops (as in this work).

An increasing number of structure prediction approaches utilize deep architectures (e.g. [7], [12] [7], [12]). Unfortunately, many of the deep approaches are not probabilistic and are mostly applicable to data of the same global structure and number of output labels as the training examples.

Sum-Product Networks (SPNs) are a class of deep and probabilistic architecture capable of performing tractable inference on high-treewidth models. In [13], Relational-SPNs were proposed that model graph-based relational data based on first-order logic. This method models graphs with potentially varying sizes by summarizing multiple variables with an aggregate statistic. In contrast, we directly model each output variable associated with nodes of the graph, and construct an SPN structure specific to each graph instance.

There have been numerous attempts to employ structured prediction to modeling semantic maps with topological spatial relations. [1] proposed Voronoi Random Fields (VRFs) which are CRFs constructed according to a Voronoi graph extracted from an occupancy grid map. In [3], 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. These approaches rely on approximate inference using Loopy BP leading to problems with convergence [1]. Moreover, in both cases, additional prior knowledge or hard commitments about the semantics of some places were employed in order to obtain a clean and manageable topological graph structure. In contrast, in this work, we rely on a graph built primarily to support navigation and execution of actions by the robot. Such graph provides a better coverage, but results in more noisy structure. Furthermore, we make no hard commitments about the semantics of the places at time of structure creation and defer such inference to the final model. Our experiments show, that under such conditions, graphical models with pairwise or higher-order potentials deteriorate quickly.

III. Preliminaries

We begin by giving a brief introduction to SPNs. For details, the reader is referred to [9], [8]. Then, we describe the topological mapping framework we used to build the topological graphs.

A. Sum-Product Networks

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 joint or conditional distributions with partition functions that are guaranteed to be tractable and involve a polynomial number of sum and product operations, permitting exact inference.

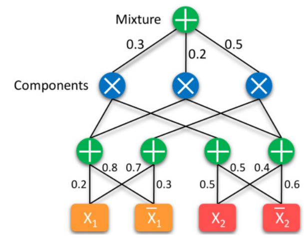

Fig. 1: An simple SPN for a naive Bayes mixture model $P(X_1, X_2)$, with three components over two binary variables. The bottom layer consists of indicators for X_1 and X_2. Weighted sum nodes, with weights attached to inputs, are marked with $+$, while product nodes are marked with $\times$.

As shown in Fig. 1, an SPN is a 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 combinations of features. SPNs can be defined for both continuous and discrete variables, with evidence for categorical variables often specified in terms of binary indicators.

Formally, following [8], 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 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 architectures consisting of sums and products result in a valid probability distribution. While a less constraining condition on validity has been derived in [8], a simpler condition, which does not limit the power of the model is to guarantee that the SPN is *complete* and *decomposable* [9].

Definition 2: A sum-product network is complete *iff* all children of the same sum node have the same scope.

Definition 3: A sum-product network is decomposable *iff* no variable appears in more than one child of a product node.

The scope of a node is defined as the set of variables that have indicators among the descendants of the node.

A valid SPN will compute unnormalized probability of evidence expressed in terms of indicators. However, the weights of each sum can be normalized, in which case the value of the 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.

Partial or missing evidence can be expressed by setting the appropriate indicators to 1. Inference is then accomplished by an upwards pass which calculates the probability of the evidence and a downwards pass which obtains gradients for calculating marginals or MPE state of the missing evidence. The latter can be obtained by replacing sum operations with weighted max operations [9].

Parameters of an SPN can be learned generatively [8] or discriminatively [14] using Expectation Maximization (EM) or gradient descent. Additionally, several algorithms were proposed for simultaneous learning of network parameters and structure [15], [16]. In this work, we use a simple structure learning technique [17] to build template SPNs representing each sub-graph. We begin by initializing the SPN with dense structure by recursively generating nodes based on multiple random decompositions of the set of variables into multiple subsets until each subset is a singleton. The resulting structure consists of products combining the subsets in each decomposition and sums mixing different decompositions at each level. Then, we employ hard EM to learn the model parameters, which was shown to work well for generative learning [17] and overcomes the diminishing gradient problem. After parameter learning, the generated structure can be pruned by removing edges associated with weights close to zero.

B. Topological Graphs

GraphSPNs are applicable to arbitrary graphs. However, in this work, we apply them specifically to topological graphs built by a mobile robot exploring a large-scale environment [18]. The primarily purpose of our topological 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 navigability. The graph nodes are associated with latent variables representing semantics and the edges can be seen as spatial relations forming a global semantic map. Local evidence about the semantics of a place might be available and we assume that such evidence is inherently uncertain and noisy. Additional nodes in the graph are created to represent exploration frontiers, possible places the robot has not yet visited, but can navigate to. We call such nodes placeholders, and assume that the robot has not yet obtained any evidence about their semantics.

The topological graph is assembled incrementally based on dynamically expanding 2D occupancy map. The 2D map is built from laser range data captured by the robot using a grid mapping approach based on Rao-Blackwellized particle filters [19]. Placeholders 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 and local evidence captured by the robot about the semantic place category is anchored to the graph node.

We formulate the problem of finding placeholder locations as sampling from a distribution that models location relevance and suitability. Specifically, the distribution is specified as: $P(E|G,\mathcal{E}) = \frac{1}{Z}\prod_i \phi_S(E_i|G)\phi_N(E_i|\mathcal{E})$, where $E_i \in \{0,1\}$ represents the existence of a new place at location i in the occupancy map, G is the occupancy grid, and $\mathcal{E}$ is the set of locations of all existing places. The potential ϕ_S ensures that placeholders are located in areas that are safe and preferred for navigation (are within safe distance from obstacles, with the preference towards centrally located places). The potential ϕ_N models the neighborhood of a place, and guarantees sufficient coverage of the environment by promoting positions at a certain distance d_n from existing places. Final location of a new placeholder is chosen through MPE inference in $P(E|G,\mathcal{E})$. An edge is then created to represent navigability. It connects the placeholder to an existing place in the graph based on A* search directly over the potential ϕ_S. An example of such semantic-topological map is shown in Fig. 4.

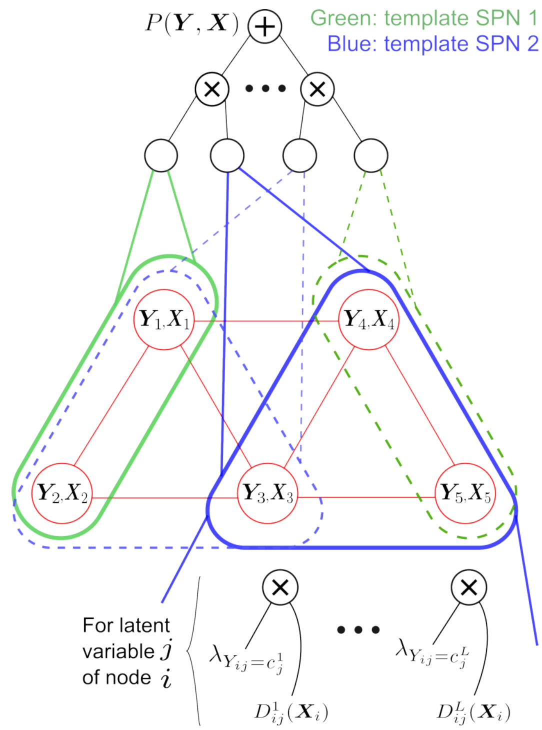

Fig. 2: An instance GraphSPN modeling a simple 5-node graph (red) with variables $\boldsymbol{X}_i$ and $\boldsymbol{Y}_i$ associated with graph nodes. Solid lines illustrate one decomposition of the graph based on two template sub-graphs and SPNs (green and blue), while dashed lines illustrate another decomposition.

IV. GraphSPNs

GraphSPNs learn a template model over arbitrary graph-structured data, with local evidence $\boldsymbol{X_i}$ and latent variables $\boldsymbol{Y}_i = \{Y_{i1}, \cdots, Y_{iM}\}$ for each graph node or edge i, with dependencies between the latent variables expressed in terms of the graph structure. Then, an *instance GraphSPN* distribution $P(\boldsymbol{X}_1, \boldsymbol{Y}_1, \ldots, \boldsymbol{X}_N, \boldsymbol{Y}_N)$ is assembled for a specific graph to perform inference.

To this end, we define a set $\boldsymbol{\mathcal{S}}$ of *template sub-graphs*, and associate each *template sub-graph* $\mathcal{S} \in \boldsymbol{\mathcal{S}}$ with a separate *template SPN* modeling the distribution over variables X_i and Y_i corresponding to the nodes and edges of the *template sub-graph*. The structure and parameters of each *template SPN* can be learned directly from data obtained by decomposing training graphs into sub-graphs corresponding to $\boldsymbol{\mathcal{S}}$.

Given a set of trained *template SPNs*, and a specific graph to be modeled, an *instance GraphSPN* is assembled as illustrated in Fig. 2. First, the graph is decomposed multiple times, each time differently, into sub-graphs using *sub-graph templates* $\boldsymbol{\mathcal{S}}$ in descending order of the template size (i.e. more complex templates have priority). The sub-graphs should not overlap in each decomposition and the corresponding *template SPNs* should cover all variables $\boldsymbol{X}_i, \boldsymbol{Y}_i$ in the model. This condition guarantees completeness and decomposability resulting in a valid *instance GraphSPN*. For each decomposition and each sub-graph, we instantiate the corresponding *template SPN* resulting in multiple SPNs sharing weights and structure. The instantiations for a single graph decomposition are combined with a product node and the product nodes for all decompositions become children of a root sum node realizing the complete mixture model.

In order to incorporate the latent variables Y_{ij}, we include an intermediate layer of product nodes into the *template SPNs*. As shown in Fig. 2, each such product node combines an arbitrary distribution $D_{ij}^k(\boldsymbol{X}_i)$ with an indicator $\lambda_{Y_{ij}=c_j^k}$ for a specific value c_j^k of Y_{ij}. The *template SPN* built on top of the product nodes can be learned from data and the distributions $D_{ij}^k(\boldsymbol{X}_i)$ can be arbitrary, potentially also realized with an SPN with data-driven structure.

In our experiments, we assumed only one latent variable (semantic place category) Y_i per graph node i, with $Val(Y_i) = \{c^1, \dots, c^L\}$, and we defined $D_i^k(\boldsymbol{X}_i)$ for a single hypothetical binary observation x_i, which we assumed to be observed:

$$D_i^k(X_i) = \begin{cases} \alpha_i^k & X_i = x_i \\ 1 - \alpha_i^k & X_i = \bar{x}_i \end{cases} \qquad (1)$$

Such simplification allows us to thoroughly evaluate GraphSPNs for the problem of learning topological semantic maps by directly simulating hypothetical evidence about the semantic category of varying uncertainty and under various noise conditions. Furthermore, it allows us to compare GraphSPNs with Markov Random Fields using the same α_i^k as the value of local potentials, i.e. $\phi_i(Y_i = c^k) = \alpha_i^k$. The proposed approach naturally extends to the case where a more complex distribution is used to model semantic place categories based on robot observations, such as the SPN-based approach presented in [17]. Note, that we still learn the structure of the *template SPNs* built on top of distributions $D_i^k(X_i)$.

V. Experimental Procedure

A. Dataset

The semantic maps with topological relations used in our experiments were obtained by deploying the topological mapping on sequences of laser range data and odometry captured by a mobile robot exploring multiple large-scale environments [20]. The dataset contains 99 sequences, and as a result 99 topological graphs, captured on 11 floors of 3 buildings in different cities. We identified 10 semantic place classes that are common for all buildings (e.g. a corridor, a doorway, a 1-person office, see Fig. 4 for a complete list) and annotated each topological graph node with its groundtruth class.

Our goal in this work is to evaluate the ability of GraphSPNs to disambiguate semantic place classes despite noisy and uncertain local evidence by exploiting spatial relations captured in a noisy topological graph. Probabilistic place classification algorithms, such as the SPN-based approach in [17] associate decisions based on local observations with probability estimates. However, the certainty of a decision can be low or the decision can be incorrect. In order to measure how sensitive the evaluated approaches are to such noise, we simulate local evidence attached to topological graph nodes by adding increasing noise to groundtruth information.

NL	$D_{groundtruth}$	$D_{incorrect}$
1	0.991 (+/-0.001)	0.0
2	0.913 (+/-0.015)	0.085 (+/- 0.056)
3	0.720 (+/-0.040)	0.090 (+/- 0.061)
4	0.434 (+/-0.054)	0.092 (+/-0.062)
5	0.316 (+/-0.030)	0.154 (+/-0.055)
6	0.154 (+/-0.021)	0.217 (+/-0.074)

TABLE I: Noise levels used in our experiments.

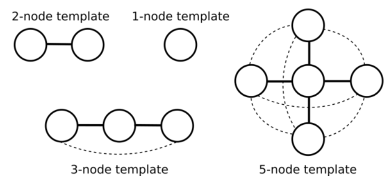

Fig. 3: Sub-graph templates used in our experiments. Dashed edges are ignored when matching the template.

To this end, for each node i in each topological graph, we generated a local evidence distribution with values $P(Y_i = c^k, X_i = x_i) = \alpha_i^k$. For each graph, we first randomly selected 20% of all nodes for which the most likely local result should be incorrect. For those, we selected a random incorrect class to be associated with the highest probability value. Then, we randomized the value $D_{incorrect}$, which is a difference between the highest probability and the probability of the groundtruth class, from a uniform distribution in a range depending on the noise level. For the remaining 80% of nodes, we ensured that the groundtruth class is associated with the highest probability. However, we simulated uncertainty by randomizing the value $D_{groundtruth}$, which is a difference between the probability of the groundtruth class and the second highest probability. With these constraints, we used random values for the remaining likelihoods and made sure that each distribution is normalized. Intuitively, lower $D_{groundtruth}$ indicates higher uncertainty and higher $D_{incorrect}$ indicates stronger noise. The statistics of the values of $D_{incorrect}$ and $D_{groundtruth}$ for the final evidence at different noise levels are shown in Tab. I.

B. Learning GraphSPNs

We learned GraphSPNs from a simple set of *sub-graph templates* shown in Fig. 3, matching from 1 to 5 nodes and simple edge configurations. We assumed that each node is associated with a single latent variable Y_i representing the semantic place class. For each template with at least 2 nodes, we learned a *template SPN* of specific structure and parameters from sub-graph examples in the training set in a supervised fashion (Y_i was set to the groundtruth).

Each training graph was partitioned in 10 different ways to obtain sub-graphs. For a single-node template we simply assumed a uniform SPN. During testing, we built the *instance GraphSPN* based on 5 different graph decompositions (see Fig. 4). In our experiments, we always learned GraphSPN from all graphs from two buildings in the dataset and tested on graphs with different evidence noise levels from the remaining building. GraphSPNs are implemented using LibSPN [21].

C. Constructing Markov Random Fields

We compared GraphSPNs to a traditional approach based on MRFs structured according to the represented graph. The MRF was constructed from two types of potentials: potential $\phi_i(Y_i = c^k) = \alpha_i^k$ used to provide local evidence, and potentials modeling latent variable dependencies. For the latter, we tried two models: using pairwise potentials for each pair of variables associated with connected nodes or defined over three variables for three connected nodes in any configuration. In each case, the potentials were obtained by generating co-occurrence statistics of variable values in the training graphs used for learning GraphSPNs. Inference in the MRF was performed using Loopy BP implemented in the libDAI library [22].

VI. Experimental Results

We performed several experiments comparing the learned GraphSPN model to the MRF models with pairwise potentials (marked as MRF-2) and three-variable potentials (marked as MRF-3). First, we tasked all models with disambiguating noisy local evidence about semantic place class for places visited by the robot. For each topological graph in the test set, we performed marginal inference[1], based on which we selected the final classification result $\mathrm{argmax}_k P(Y_i = c^k|\boldsymbol{X} = \boldsymbol{x})$.

The percentage of correctly classified nodes in the graph averaged over all test graphs is shown in Tab. II and a visualization of results for a single graph together with the decompositions used to build the *instance GraphSPN* is shown in Fig. 4. We evaluated all assignments of the three buildings into training and test sets as well as different noise levels listed in Tab. I. Each test set consisted of 5 topological maps, each with 3 random sets of noisy local evidence resulting in 15 different test graphs. Since the local evidence for 20% of nodes in each graph indicates an incorrect class as the most likely one, only accuracy greater than 80% demonstrates that the model was able to recover from the noise using learned spatial relations. Lower accuracy suggests that the incorrect evidence was too strong or that the correct evidence was too uncertain to influence the semantic class of a place.

Analyzing the performance reported in Tab. II, we see that pairwise MRF performs well when there is little noise in the evidence, however it deteriorates quickly with increasing noise levels. At the same time, GraphSPN generating robust results (accuracy greater than 80%) despite substantial noise and uncertainty. With substantial noise, approximate Loopy BP inference for MRF converges to a solution consisting primarily of the dominant class (the corridor). At the same time, we see that using higher-order potentials with MRF actually hurts performance.

In the second experiment, we tasked the models with inferring marginal distributions over the semantic classes of places not yet visited by the robot (placeholders) for which local evidence is unavailable. We used the same setup as in the previous experiment, with local evidence for explored places including noise and uncertainty. Examples of such marginal distributions are shown in Fig. 5, while the classification accuracy when considering the most likely class is reported in Tab. III. Again, GraphSPN significantly outperformed the MRF for this inference task. If we analyze the marginal distributions for three representative placeholders shown in Fig. 5, we see that GraphSPN is confident about the correct class for the two placeholders for which nearby

[1]We experimented with MPE inference over all latent variables achieving inferior results with all models.

GraphSPN			
NL	Freiburg	Saarbrücken	Stockholm
1	96.13%(+/-2.41)	95.45%(+/-3.05)	93.98%(+/-1.90)
2	96.63%(+/-2.39)	96.37%(+/-3.03)	94.01%(+/-2.12)
3	92.45%(+/-2.43)	93.43%(+/-2.85)	92.04%(+/-2.57)
4	91.88%(+/-2.76)	92.91%(+/-2.72)	86.31%(+/-2.06)
5	90.13%(+/-3.47)	90.12%(+/-3.56)	83.51%(+/-3.13)
6	80.83%(+/-5.21)	80.08%(+/-3.95)	69.49%(+/-5.69)
MRF-2			
NL	Freiburg	Saarbrücken	Stockholm
1	91.54%(+/-6.89)	82.32%(+/-15.74)	72.71%(+/-13.42)
2	78.37%(+/-10.03)	76.15%(+/-16.63)	53.09%(+/-10.22)
3	59.91%(+/-12.99)	56.05%(+/-17.10)	28.74%(+/-4.89)
4	44.17%(+/-10.54)	50.77%(+/-17.35)	24.84%(+/-5.04)
5	44.74%(+/-8.92)	47.09%(+/-14.41)	23.20%(+/-2.90)
6	44.30%(+/-10.53)	50.07%(+/-15.19)	22.98%(+/-4.42)
MRF-3			
NL	Freiburg	Saarbrücken	Stockholm
1	45.27%(+/-7.43)	50.61%(+/-15.03)	28.65%(+/-4.90)
2	48.70%(+/-7.30)	49.50%(+/-15.23)	26.58%(+/-3.75)
3	43.27%(+/-8.75)	55.47%(+/-21.02)	26.17%(+/-5.13)
4	47.24%(+/-9.81)	49.18%(+/-12.27)	24.18%(+/-4.47)
5	45.23%(+/-9.98)	49.76%(+/-17.81)	25.48%(+/-5.34)
6	47.80%(+/-10.37)	51.85%(+/-17.54)	25.25%(+/-5.19)

TABLE II: Semantic place classification accuracy for all models and test buildings, and at different noise levels.

GraphSPN			
NL	Freiburg	Saarbrücken	Stockholm
2	67.58%(+/-10.42)	78.15%(+/-9.95)	67.57%(+/-11.11)
5	40.59%(+/-12.22)	55.18%(+/-19.67)	37.56%(+/-10.44)
MRF-2			
NL	Freiburg	Saarbrücken	Stockholm
2	28.32%(+/-7.53)	39.85%(+/-19.42)	12.44%(+/-3.46)
5	24.23%(+/-11.40)	30.58%(+/-5.57)	10.04%(+/-2.59)
MRF-3			
NL	Freiburg	Saarbrücken	Stockholm
2	28.71%(+/-5.43)	31.94%(+/-5.26)	10.11%(+/-0.51)
5	18.02%(+/-7.49)	28.86%(+/-6.16)	8.96%(+/-1.19)

TABLE III: Accuracy of semantic class inference for placeholders without local evidence, for all models and test buildings, and at two representative noise levels.

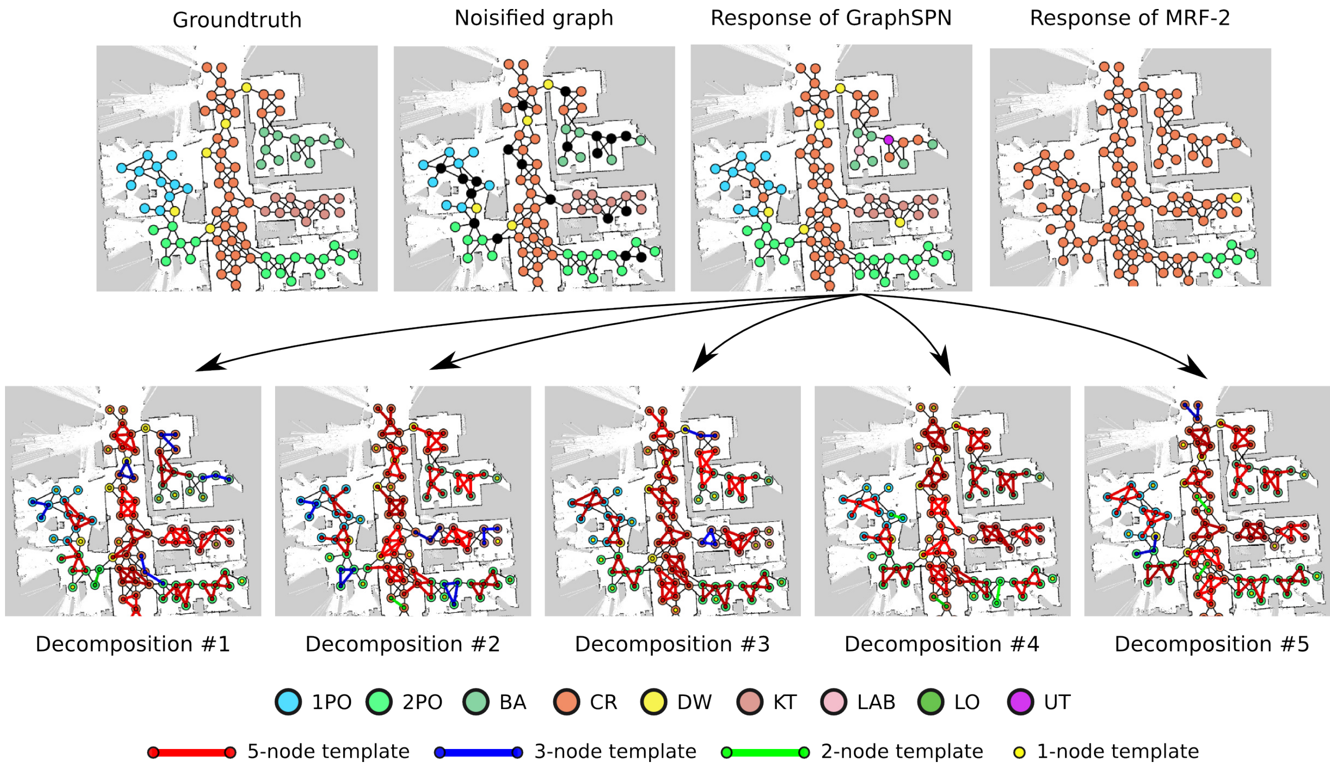

Fig. 4: Visualization of the results for a graph from Freiburg at noise level 4. The top row shows: the semantic map with groundtruth semantic place classes; the 20% of nodes for which the most likely evidence indicates an incorrect class (black nodes); the semantic classes inferred by the GraphSPN and the MRF-2. The place classes are: 1-person office (1PO), 2-person office (2PO), bathroom (BA), corridor (CR), doorway (DW), kitchen (KT), laboratory (LAB), large office (LO), meeting room (MR), utility room (UT). The bottom row illustrates the 5 decompositions used when assembling the *instance GraphSPN* (different colors indicate different sub-graph templates applied).

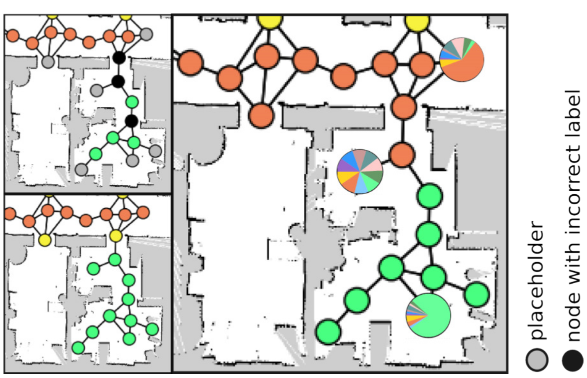

Fig. 5: Visualization of results for the experiment involving placeholders without local evidence, at noise level 2. Left, bottom: the semantic map with groundtruth semantic place classes (including for placeholders). Left, top: the 20% of nodes for which the most likely evidence indicates an incorrect class (black) and placeholders with no evidence (gray). Right: inferred marginal distributions over semantic classes of placeholders (pie charts).

nodes provide correct (albeit uncertain) evidence. For the placeholder connected to nodes for which evidence indicates an incorrect class the marginal distribution is almost uniform. This is an indication of the ability of GraphSPN to generate useful confidence signal in presence of noisy evidence.

VII. Conclusions

We presented GraphSPNs, a probabilistic deep model for graph-structured data that learns a template distribution allowing for making inferences about graphs of different global structure. While existing works applied SPNs to data organized as fixed-size grids or sequences, this paper presents a novel attempt at deploying SPNs on arbitrary graphs of varying size. Based on GraphSPNs, we proposed a method for learning topological spatial relations in semantic maps constructed by a mobile robot. Our method is robust to noise and uncertainty inherent in real-world problems where information about the environment is captured with robot sensors. Our framework is universal and compatible with any distributions defined over local evidence. However, it is particularly well suited for integration with other SPN-based models. In the future, by combining a GraphSPN learned over semantic maps with the generative place model proposed in [17], we intend to achieve a unified, deep, and hierarchical representation of spatial knowledge spanning from local sensory observations to global conceptual descriptions.

References

[1] S. Friedman, H. Pasula, and D. Fox, "Voronoi random fields : Extracting the topological structure of indoor environments via place labeling," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[2] H. Zender, O. Martínez Mozos, P. Jensfelt, G.-J. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robotics and autonomous systems*, vol. 56, no. 6, 2008.

[3] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA'12)*, 2012.

[4] F. R. Bach and M. I. Jordan, "Thin junction trees," in *Advances in Neural Information Processing Systems 14*, 2002.

[5] D. Belanger and A. McCallum, "Structured prediction energy networks," in *International Conference on Machine Learning*, 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] A. G. Schwing and R. Urtasun, "Fully connected deep structured networks," 2015, arXiv:1503.02351.

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

[9] 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.

[10] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[11] K. P. Murphy, Y. Weiss, and M. I. Jordan, "Loopy belief propagation for approximate inference: An empirical study," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999.

[12] L.-C. Chen, A. Schwing, A. Yuille, and R. Urtasun, "Learning deep structured models," in *International Conference on Machine Learning*, 2015.

[13] A. Nath and P. Domingos, "Learning relational Sum-Product networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

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

[15] W. Hsu, A. Kalra, and P. Poupart, "Online structure learning for Sum-Product networks with gaussian leaves," 2017.

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

[17] 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.

[18] 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.

[19] 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.

[20] A. Pronobis and B. Caputo, "COLD: COsy Localization Database," *The International Journal of Robotics Research (IJRR)*, vol. 28, no. 5, 2009.

[21] A. Pronobis, A. Ranganath, and R. P. N. Rao, "LibSPN: A library for learning and inference with Sum-Product Networks and TensorFlow," 2017. [Online]. Available: http://www.libspn.org

[22] 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.