2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

V-PRISM: Probabilistic Mapping of Unknown Tabletop Scenes

Herbert Wright[1] Weiming Zhi[2] Matthew Johnson-Roberson[2] Tucker Hermans[1,3]

***Abstract*—The ability to construct concise scene representations from sensor input is central to the field of robotics. This paper addresses the problem of robustly creating a 3D representation of a tabletop scene from a segmented RGB-D image. These representations are then critical for a range of downstream manipulation tasks. Many previous attempts to tackle this problem do not capture accurate uncertainty, which is required to subsequently produce safe motion plans. In this paper, we cast the representation of 3D tabletop scenes as a multi-class classification problem. To tackle this, we introduce V-PRISM, a framework and method for robustly creating probabilistic 3D segmentation maps of tabletop scenes. Our maps contain both occupancy estimates, segmentation information, and principled uncertainty measures. We evaluate the robustness of our method in (1) procedurally generated scenes using open-source object datasets, and (2) real-world tabletop data collected from a depth camera. Our experiments show that our approach outperforms alternative continuous reconstruction approaches that do not explicitly reason about objects in a multi-class formulation.**

I. Introduction

As robots continue to be deployed in the world, there is an ongoing need for methods that allow them to safely and robustly operate in unknown, noisy scenes. Many such scenes contain objects that robots must delicately move around or interact with to complete their assigned tasks. The planning techniques for such tasks often require an accurate 3D map of the objects within the scene. These are often unseen objects with unknown geometry that are only partially observed.

The safe operation of robots necessitates not only accuracy but also introspection and uncertainty-awareness. These notions of uncertainty about the geometry of the scene can then be incorporated into downstream motion planning solvers for added robustness and safety. However, many learning algorithms typically used in robot learning, such as neural networks, lack the ability to reason about uncertainty and confidently predict incorrect labels [1], [2]. In this work, we take a Bayesian learning approach which captures uncertainty in a principled manner.

We propose V-PRISM: Volumetric, Probabilistic, and Robust Instance Segmentation Maps*. V-PRISM is a framework for building differentiable segmentation and occupancy maps of tabletop scenes that contain multiple unseen objects. Importantly, our method results in maps with a principled and understandable uncertainty metric. To construct these maps, we rely on depth measurements with corresponding instance segmentations. Such instance segmentations can be easily obtained for real-world scenes using pre-existing models such as those proposed in [3]–[5]. We take inspiration from Bayesian Hilbert Maps (BHMs) [6] and transform points into an embedding induced by a set of chosen hinge points in order to perform Bayesian updates to our map. These updates are made in a variational manner with an expectation maximization (EM) algorithm. In order to effectively learn the geometry of the scene, we propose a negative sampling method for encoding depth sensor information in object-centric scenes. The learned map can be used to reconstruct the objects in the scene as well as measure the uncertainty about the geometry in different areas of the scene. This is pictured in Figure 1.

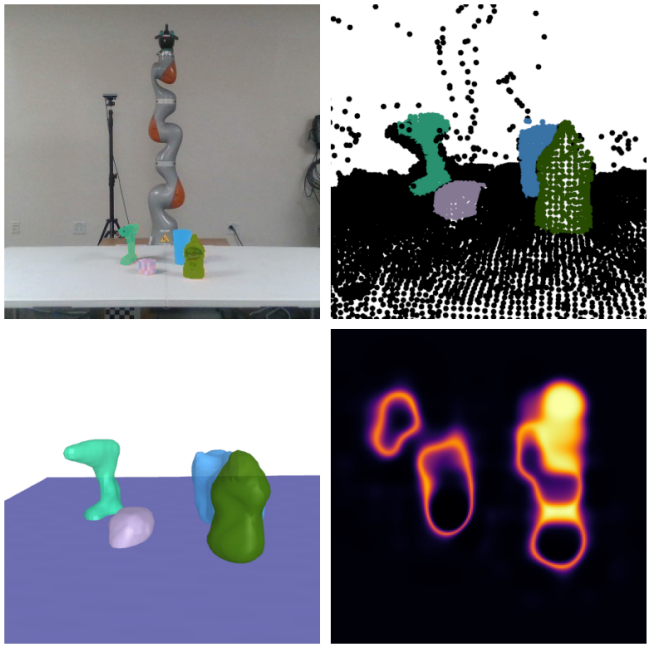

Figure 1: Our method takes a segmented (**top left**) point cloud observation (**top right**) and builds a continuous probabilistic map. This map can be used to reconstruct the scene (**bottom left**) or measure uncertainty about the scene (**bottom right**). The heat map shows uncertainty in a 2D slice parallel with the table plane. Uncertainty is high in occluded areas.

We evaluate our method in simulation and the real world. The simulation scenes are constructed using objects from existing mesh datasets by placing them in a random configuration within a simulator similar to [7]. We run extensive experiments and report measurements of two commonly used metrics: intersection over union (IoU) and Chamfer distance of reconstructed meshes. Qualitative reconstructions and uncertainty estimates are computed on real world scenes of objects belonging to unknown classes to demonstrate robustness to noise associated with real-world cameras.

Concretely, our technical contributions include:

- The formulation of 3D scene reconstruction as a multi-

[1] University of Utah Robotics Center and Kahlert School of Computing, University of Utah, Salt Lake City, UT, USA
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[3] NVIDIA Corporation, Santa Clara, CA, USA

*Website for project is https://herb-wright.github.io/v-prism/

class mapping problem
- A principled *Bayesian* framework to learn *continuous* maps for tabletop scene representation
- An object-centric sampling method that enables accurate and efficient reconstructions

Our paper is organized as follows. In Section II, an overview of related research is provided. In Section III, we review the basics of BHMs and formulate the problem our method aims to solve. Following this, Section IV provides a high level overview of our proposed method. The math behind our EM algorithm and Bayesian model are discussed Section V. We propose a novel negative sampling method in Section VI specifically for object-centric mapping. In Section VII We justify our decisions through quantitative and qualitative experiments. We also give qualitative examples of how our method provides desireable uncertainty measurements. This is followed by brief conclusion in Section VIII.

II. Related Works

3D Mapping. Constructing a 3D map of an environment has been a common problem in the field of robotic perception. Voxel based approaches such as truncated signed distance functions [8] and OctoMaps [9] are a common approach to the problem. Hilbert Maps [10], and their extension to Bayesian Hilbert Maps [6] are both methods for mapping 3D environments. Recently, learning based approaches such as [11] have grown in popularity. While these works serve as inspiration to our work, mapping a surrounding environment is a different problem than the one we approach in this paper.

Multiple View Synthesis. One common way of reconstructing the geometry of an object or scene is to combine multiple camera views and observations into a 3D representation. Recently, neural radiance fields [12] and variants such as Plenoxels [13] learn an implicit density field using multiple views without depth information. Another learning based method for reconstructing scenes with multiple objects from multiple views is introduced in [14], where a voxel encoder-decoder network is used. 3D Gaussian Splatting has also been used to reconstruct density fields from multiple images [15]. While there has been a lot of interesting work around recovering 3D geometry by synthesizing multiple views, this work is directed at the harder problem of recovering 3D geometry using only a single view.

Reconstructing Single Objects. Many methods have been proposed to reconstruct single objects. In [16], superquadrics are fit to the observed points to generate an objects geometry. Another method for recovering object geometry is Gaussian process implicit surfaces [17] which implicitly reconstructs objects using a handful of surface points and their corresponding normals specifically for the robotic task of grasping. More recently, deep learning has been utilized to predict the geometry of objects from a single view. One deep learning approach is proposed in [18], where a single image is used as input to a neural network that predicts object geometry. DeepSDF [19] and PointSDF [20] are both learning based approaches that predict a signed distance function given a point cloud. Occupancy networks [21] also take a single point cloud or image as input, but instead predict an occupancy function that maps each point in space to a probability of being occupied. Other work, such as [22] utilizes different modes of input like tactile measurements. 3D-R2N2 [23] creates a voxelized representation of objects using a recurrent neural network that allows for single and multi-view reconstruction. NKF [24] uses neural splines and kernel ridge regression to build an implicit signed distance function. The GenRe algorithm aims to predict the geometry of specifically unknown object classes as proposed in [25], where the authors explain that trying to generalize to unseen classes is much more difficult than traditional reconstruction. Occlusion is also a problem for many single-object reconstruction methods, as they generally assume that no other objects are present to partially obstruct an object.

Reconstructing Multiple Objects. Some methods have been proposed that attempt to reconstruct scenes containing multiple objects with some occlusion. 3DP3 [26] is a method specifically for multi object scenes that assumes known object classes and uses probabilistic programming to reconstruct the scene. In [27], silhouettes are used to refine the voxelized predicted geometry of objects under occlusion. Another reconstruction technique proposed recently is ARM [7], where scenes are encoded as voxels and a loss function is used that includes terms for connectivity and stability in order to increase generalizability. In contrast to our method, these approaches either assume known object classes or do not accurately measure the uncertainty over the scene.

III. Preliminaries

A. Sigmoid Bayesian Hilbert Maps

Hilbert Maps. Introduced in [10], Hilbert Maps are a method for continuous occupancy mapping of a robotic environment. A map $m : \mathbb{R}^d \to [0, 1]$ is built from a feature transform $\phi(\mathbf{x})$ and a set of n' point observations $\{\mathbf{x}_i\}_{i\in[n']}$ from a sensor at position $\mathbf{o}$. The observed point cloud is labeled with $y_i = 1$ and unoccupied negative samples drawn on the line segments between each $\mathbf{x}_i$ and $\mathbf{o}$ are labeled with $y_i = 0$. The observed points, sampled points, and labels form the input data, $\{(\mathbf{x}_i, y_i)\}_{i\in[n]}$. Gradient descent is used to find the optimal weights for a map of the form

$$m(\mathbf{x}) = \sigma(\mathbf{w}^\top \phi(\mathbf{x})) = (1 + \exp(-\mathbf{w}^\top \phi(\mathbf{x})))^{-1},$$

where $\sigma : \mathbb{R} \to (0, 1)$ is the sigmoid function. This is equivalent to performing logistic regression over the transformed points $\{(\phi(\mathbf{x}_i), y_i)\}_{i\in[n]}$.

Usually, the feature transform ϕ is constructed from a kernel function k and a set of hinge points $\mathbf{h}_1, ..., \mathbf{h}_m \in \mathbb{R}^3$. Usually, these hinge points are chosen to be an evenly spaced 3D grid of points. The feature transform is then given by:

$$\phi(x) = \begin{bmatrix} k(\mathbf{x}, \mathbf{h}_1) \\ k(\mathbf{x}, \mathbf{h}_2) \\ ... \\ k(\mathbf{x}, \mathbf{h}_m) \\ 1 \end{bmatrix}. \tag{1}$$

Bayesian Extension. Hilbert Maps were extended to the Bayesian setting in [6]. Instead of an individual weight

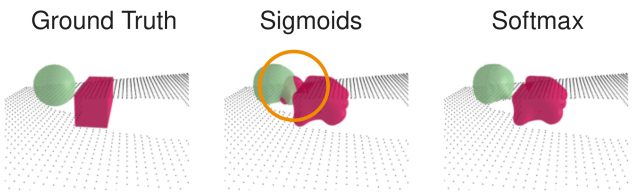

Figure 2: Running a separate sigmoid model per object can cause unwanted intersections between the reconstructions (circled). Our multi-class formulation uses a softmax model that avoids this problem.

vector, the weight is treated as a normally distributed random variable, $\mathbf{w} \sim P(\mathbf{w})$. Variational Bayesian logistic regression as described in [28] is then performed over data $D = \{(\phi(\mathbf{x}_i), y_i)\}_{i\in[n]}$ in order to obtain the approximate posterior distribution:

$$\hat{P}(\mathbf{w}|D) \propto Q(D|\mathbf{w};\xi)P(\mathbf{w}) \approx P(D|\mathbf{w})P(\mathbf{w}),$$

where the variational parameter ξ is introduced. The method relies on an EM algorithm that alternates between calculating the posterior $\hat{P}(\mathbf{w}|D) = \mathcal{N}(\hat{\mu}, \hat{\Sigma})$ from an approximate likelihood function and obtaining a better likelihood approximation. The specific approximation used for the likelihood takes the form of a normal distribution, and ensures that the approximated likelihood is conjugate to a normal prior $P(\mathbf{w}) = \mathcal{N}(\bar{\mu}, \bar{\Sigma})$.

Once the posterior weight distribution is obtained, the map m is defined by the expectation:

$$m(\mathbf{x}) = \mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^\top \phi(\mathbf{x}))].$$

Because there is not an analytic solution for this expectation, approximations are used. The most common approximation is

$$\mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^\top \phi(\mathbf{x}))] \approx \sigma\left(\frac{\mathbb{E}_{\mathbf{w}}[\mathbf{w}^\top \phi(\mathbf{x})]}{\sqrt{1 + \frac{\pi}{8}\text{Var}(\mathbf{w}^\top \phi(\mathbf{x}))}}\right), \quad (2)$$

which is easily obtained for any $\mathbf{w}$ following a normal distribution.

Extensions of BHMs include Bayesian treatment of kernel parameters and hinge point placement [29], fusing two BHMs [30], and mapping environments with moving actors [31].

B. Problem Formulation

Instead of predicting an occupancy map for each object, we phrase our problem as a multi-class mapping problem. This ensures that each point in space can only be occupied by a single object. Without this constraint, reconstructions of objects can intersect each other as shown in Figure 2. Formally, we receive observations $\{(\mathbf{x}_i, y_i)\}_{i\in[n]}$ where $\mathbf{x}_i \in \mathbb{R}^d$ corresponds to an observed point with segmented class $y_i \in [c]$. We assume $y_i = 1$ denotes $\mathbf{x}_i$ being segmented to no specific object and is part of the background or table. We also assume that these observations came from a camera with a known location $\mathbf{o} \in \mathbb{R}^d$. The goal is to build a map function $m : \mathbb{R}^d \to [0, 1]^c$ such that $m(\mathbf{x})$ corresponds to the probability distribution over classes that the point $\mathbf{x}$ could belong to.

We would like our map to satisfy that $m(\mathbf{x}_i) \approx \mathbf{e}_{y_i}$ for all i, where $\mathbf{e}_{y_i}$ is the one hot encoding of y_i. We can infer that for any $\mathbf{x}_i$, because the camera ray started at $\mathbf{o}$ and terminated at $\mathbf{x}_i$, all points in between are unoccupied. We would like our map to reflect this realization. This forms the basis for the negative sampling performed in [6]. We will also assume that objects in the scene are resting on or above a planar surface. While this typically means a table, our method is agnostic to the type of surface.

IV. Method Overview

Our method builds a map $m(\mathbf{x})$ from segmented camera depth observations of a multi-object scene through two main steps. A high level overview is displayed in Figure 3. First, negative sampling is performed as described in Section VI, where additional points are added to the observed ones in order to form a new labelled point cloud. During this step, the RANSAC [32] algorithm is run in order to recover the surface plane the objects are resting on. The points are also subsampled in order to increase efficiency. We then generate a set of hinge points that are used to construct a feature transform according to Equation (1). This transform, along with our sampled points, creates a set of augmented data.

Once we have our tranformed data, we perform Bayesian multi-class regression over the data with an expectation maximization (EM) algorithm. The specific technique makes use of mathematical ideas from [33]. The full EM algorithm and model are explored in Section V. Efficiently evaluating $m(\mathbf{x})$ for query $\mathbf{x}$ values is also covered in Section V, where we make use of an approximation proposed in [34]. The segmentation map produced maps each point in 3D space to a distribution over c classes, where one class denotes not belonging to an object and the other $c-1$ classes denote the segmented objects observed.

Once we have our map, we can use it to evaluate how likely different points are to be in occupied by different objects. This is useful in many motion planning algorithms in order to minimize unwanted collisions. We can also reconstruct the meshes of each object by running the marching cubes algorithm [35]. These meshes can be used to create a signed distance function, simulate physics, or to visualize the scene. Our map also encodes principled uncertainty about the geometry of the scene which can be used for active inference.

V. Softmax EM Algorithm

A. Training

To create a Bayesian multi-class map, we consider using a weight matrix $\mathbf{W} \in \mathbb{R}^{c\times m}$ where each row is normally distributed, giving the following likelihood function:

$$P(y = k|\mathbf{W}, \mathbf{x}) = \text{softmax}(\mathbf{W}\phi(\mathbf{x}))_k,$$

where the softmax function is defined as

$$\text{softmax}(\mathbf{W}\phi(\mathbf{x}))_k = \frac{\exp(\mathbf{W}\phi(\mathbf{x})_k)}{\sum_{i=1}^{c}\exp(\mathbf{W}\phi(\mathbf{x})_i)}.$$

Because a conjugate prior for the softmax likelihood doesn't exist, we must use variational inference to find a posterior Gaussian distribution. In our case, we will maximize

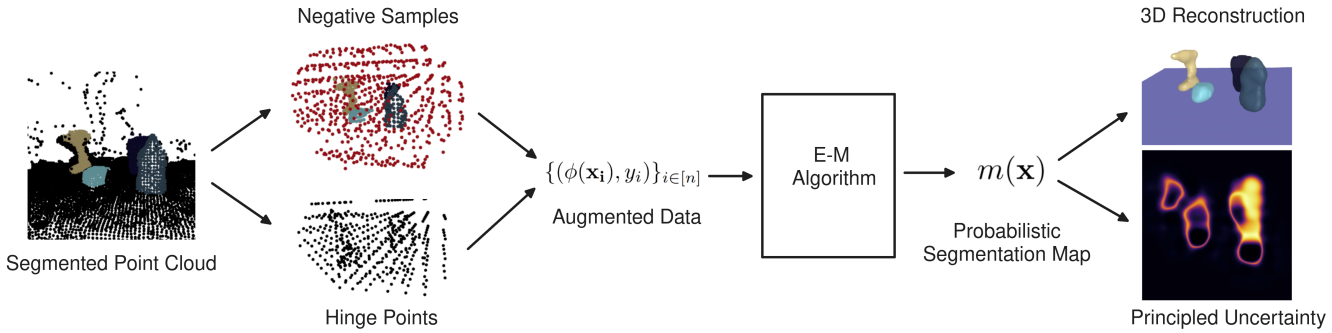

Figure 3: Overview of our method, V-PRISM. We take a segmented point cloud and output a probabilistic segmentation map over 3D space that can be used for both object reconstruction and principled uncertainty. Our method first generates negative samples and hinge points, then uses these to create an augmented dataset. Then the probabilistic map is constructed by running an EM algorithm over this dataset.

a lower bound on the likelihood. A useful inequality for this is given in [33], and is stated in the following theorem:

***Theorem** 1:* From [33]. Let $\mathbf{z} \in \mathbb{R}^c$, $\alpha \in \mathbb{R}$, and $\xi \in \mathbb{R}^c_+$. Then the following inequality holds:

$$\ln \sum_{k=0}^{c} \exp(\mathbf{z}_k) \leq \alpha + \sum_{k=0}^{c} \frac{\mathbf{z}_k - \alpha - \xi_k}{2} + \lambda(\xi_k)((\mathbf{z}_k - \alpha)^2 - \xi_k^2) + \ln(1 + \exp(\xi_k)),$$

where $\lambda(\xi_k) = ((1 + \exp(-\xi_k))^{-1} - (1/2))/2\xi_k$.

Applying Theorem 1 to $\mathbf{z} = \mathbf{W}\phi(\mathbf{x})$, we can bound the likelihood by introducing the two variational parameters α and ξ with the inequality,

$$\ln P(y = k|\mathbf{W}, \mathbf{x}) \geq \ln Q(y = k|\mathbf{W}, \mathbf{x}; \alpha, \xi).$$

We can maximize this lower bound and use it as an approximation to the true likelihood by solving the following:

$$\arg\max_{\alpha,\xi} \mathbb{E}_{\mathbf{W}} \left[\ln Q(y = y_i|\mathbf{x}_i, \mathbf{W}; \alpha, \xi)\right].$$

This can be analytically solved for $\mathbf{W}_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, yielding the following optimal values found in [33]:

$$\alpha_i = \frac{\frac{1}{2}(\frac{c}{2} - 1) + \sum_{k=1}^{c} \lambda(\xi_k)\mu_k^\top \phi(\mathbf{x}_i)}{\sum_{k=1}^{c} \lambda(\xi_k)}, \tag{3}$$

$$\xi_{i,k}^2 = \phi(\mathbf{x}_i)^\top \Sigma_k \phi(\mathbf{x}_i) + (\mu_k^\top \phi(\mathbf{x}_i))^2 + \alpha_i^2 - 2\alpha_i \mu_k^\top \phi(\mathbf{x}_i). \tag{4}$$

Due to the inequality used, $P(y = k|\mathbf{W}, \mathbf{x}; \alpha, \xi)$ is normally distributed for any α, ξ and will be conjugate to our prior weight distribution. Thus, we have a closed-form for the approximate posterior distribution, $P(\mathbf{W}|y = k, \mathbf{x}) = \mathcal{N}(\hat{\mu}, \hat{\Sigma})$. The update equations mirror those found in [33] and are as follows:

$$\hat{\Sigma}_k^{-1} = \bar{\Sigma}^{-1} + 2\sum_{i=1}^{n} \lambda(\xi_{i,k})\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top \tag{5}$$

$$\hat{\mu}_k = \hat{\Sigma}_k \left[\bar{\Sigma}_k^{-1}\bar{\mu}_k + \sum_{i=1}^{n} \left(y_{i,k} - \frac{1}{2} + 2\alpha_i \lambda(\xi_{i,k})\right)\phi(\mathbf{x}_i)\right]. \tag{6}$$

We can use Equation (3), Equation (4), Equation (5), and Equation (6) to create an EM algorithm to iterate between calculating our posterior distribution and optimizing our variational parameters, shown in Algorithm 1. The size of $\hat{\Sigma}_k$ scale quadratically with the feature dimension.

Algorithm 1 V-PRISM

Input:
Observed, segmented points $o = \{(\mathbf{x}_i, y_i)\}_{i\in[n']}$
Prior means $\{\bar{\mu}_k\}_{k\in[c]}$ and covariances $\{\bar{\Sigma}_k\}_{k\in[c]}$

$\mathcal{D} \leftarrow$ NEGATIVESAMPLE(o)
$\phi \leftarrow$ HINGEPOINTTRANSFORM(o)
$\xi_{i,k} \leftarrow 1$ for $i \in [m], k \in [c]$
$\alpha_i \leftarrow 0$ for $i \in [m]$
for p iterations **do**
  $\hat{\Sigma}^{-1} \leftarrow \bar{\Sigma}^{-1} + 2\sum_i |\lambda(\xi_i)|\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top$
  $\hat{\mu}_k \leftarrow \hat{\Sigma}\left(\bar{\Sigma}^{-1}\bar{\mu} + \sum_i (y_i - \frac{1}{2} + 2\alpha_i\lambda(\xi_{i,k}))\phi(\mathbf{x}_i)\right)$
  $\alpha_i \leftarrow$ UPDATEALPHA$(\xi_i, \mathbf{x}_i, \hat{\mu}, \hat{\Sigma})$ with Equation (3)
  $\xi_{i,k} \leftarrow$ UPDATEXI$(\alpha_i, \mathbf{x}_i, \hat{\mu}, \hat{\Sigma})$ with Equation (4)
end for
return $\hat{\mu}, \hat{\Sigma}$

B. Inference

In order to make predictions about new points we need to evaluate the following expectation:

$$\hat{P}(y = k|\mathbf{x}) = \mathbb{E}_{\mathbf{W}} \left[\text{softmax}(\mathbf{W}\phi(\mathbf{x}))\right]_k. \tag{7}$$

There is not a closed form solution to this expectation, so we must approximate it. While we could use sampling to estimate the expectation, we instead use a more computationally efficient approximation.

As described in [34], we can write the softmax in terms of the sum of sigmoidal terms with the following equality:

$$\text{softmax}(\mathbf{a})_k = \frac{1}{2 - c + \sum_{i\neq k} \sigma(\mathbf{a}_k - \mathbf{a}_i)^{-1}},$$

where c is the number of classes. This is then used as motivation for the approximating the expectation with

$$\mathbb{E}_{\mathbf{W}} \left[\text{softmax}(\mathbf{W}\phi(\mathbf{x}))\right]_k \approx \frac{1}{2 - c + \sum_{i\neq k} \mathbb{E}[\sigma(\tilde{\mathbf{z}}_i)]^{-1}},$$

with $\tilde{\mathbf{z}}_i = [\mathbf{W}\phi(\mathbf{x})]_k - [\mathbf{W}\phi(\mathbf{x})]_i$. When combined with the sigmoidal approximation in Equation (2), this becomes an easily computable approximation to Equation (7).

VI. Object-Centric Negative Sampling

Similar to many mapping methods, V-PRISM requires sampling negative unoccupied points along depth camera rays. The traditional negative sampling used, mentioned in Section III-A, is meant for mapping environments where the robot is in an enclosed space and each camera ray is detecting a wall or sufficiently large object. This sampling performs poorly when the goal is to map a relatively small object resting on a tabletop or other surface. To fully utilize the tabletop structure within the environment, we propose a new negative sampling method designed for object-centric mapping. Our sampling method rests on two main realizations:

1) Along the ray, negative samples are most useful when near known objects.
2) Points below a surface plane cannot be occupied by objects resting entirely on or above that surface.

We assume we have a segmented point cloud of the scene $\{(\mathbf{x}_i, y_i)\}_{i\in[n']}$ where each y_i corresponds to the segmentation label of the respective $\mathbf{x}_i$. We also assume a known position of the camera $\mathbf{o}$. Our sampling method begins by finding the center of the smallest axis-aligned bounding box that contains all of the segmented points for each individual object in the scene. We denote these centers with $\mathbf{o}_k$. We then perform stratified uniform sampling along each ray, only keeping points that are within r_{obj} distance from at least one $\mathbf{o}_k$. Sampled points within the desired radius of a center are labeled as unoccupied and added to the collection of points for the algorithm.

Next, we run RANSAC [32] on the observed point cloud to recover the table plane. Once we have the plane, we uniformly randomly sample points within r_{obj} from each object center and keep any such points that fall below the plane. These points are labelled as unoccupied and added to our collection.

Finally, we perform grid subsampling as described in [36] with each label in parallel in order to reduce the number of points our algorithm is fed. In practice, we choose different resolutions to subsample empty points and points on object surfaces. This can dramatically increase the efficiency of our method by removing redundant points. The entire negative sampling process is shown in Figure 4.

The resulting points are then transformed to construct our set of augmented data. The transform used is induced by a set of hinge points according to Equation (1). In practice, we choose a set of hinge points consisting of a fixed grid around the scene as well as a fixed number of random points sampled from the surface points of each object.

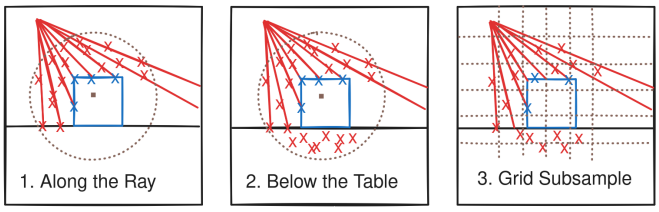

Figure 4: Overview of our sampling method. 1. We perform stratified sampling along camera rays within r_{obj} of the object. 2. Points are sampled below the table within r_{obj} of the object. 3. Grid subsampling is performed.

VII. Experiments

We perform experiments aimed to answer the following questions: 1. Does our method result in accurate reconstructions? 2. Does our sampling method improve map quality for object-centric mapping? 3. Is our method robust to unknown, noisy scenes? 4. Does our map accurately capture uncertainty about the scene geometry? We test 1 and 2 in Section VII-B, 3 in Section VII-C, and 4 in Section VII-D. We implement V-PRISM in PyTorch and run our algorithm on an NVIDIA GeForce RTX 2070 GPU.

A. Baselines and Metrics

Baselines: We compare our method to two different baselines. The first is a voxel-based heuristic that labels observed unoccupied voxels as unoccupied, observed occupied voxels as their corresponding segmentation label, and unobserved voxels with the same label as the nearest observed voxel. To prevent incorrect predictions below the table plane, we also run RANSAC during our baseline and label all voxels under the plane as unoccupied. We refer to this approach as the **Voxel** baseline. The second baseline is a learning-based approach using a state of the art neural network architecture for continuous object reconstructions in robotics. We take the PointSDF architecture from [20] and replace the final activation with a sigmoid function to predict occupancy probabilities. We train this model on a dataset of scenes similar to those discussed in Section VII-B. The scenes are composed of a subset of the ShapeNet [37] dataset. Training it on these scenes instead of the original dataset PointSDF was trained on allows it to better function under occlusion and different scales. We refer to this baseline as **PointSDF**.

Metrics: We use two main metrics for comparison: **intersection over union (IoU)** and **Chamfer distance**. IoU is calculated by evaluating points in a fixed grid around each object. Chamfer distance is calculated by first reconstructing the predicted mesh by running the marching cubes algorithm [35] on a level set of $\hat{P}(y = 1|x) = \tau$ for a chosen τ of the prediction function. Then, points are sampled from both the predicted mesh and ground truth mesh and the Chamfer distance is calculated between these two point clouds.

B. Generated Scenes from Benchmark Object Datasets

In this section, we evaluate our method against the two baseline methods on procedurally generated scenes, from large object datasets. We generate a scene by randomly picking a mesh and placing it at a random pose within predefined bounds with a random scale. We draw meshes from the ShapeNet [37], YCB [38], and Objaverse [39] datasets. We generate 100 scenes for each dataset with up to 10 objects in each scene. Objects are placed relatively close together in order to ensure significant occlusion in the scenes. Once the poses have been selected, we simulate physics for a fixed period of time to ensure objects can come to rest.

Our first experiment on simulated scenes compares our method with the two baselines. Similar to [21], we use a

Method	Continuous	Principled Uncertainty	ShapeNet Scenes IoU ↑	ShapeNet Scenes Chamfer (m) ↓	YCB Scenes IoU ↑	YCB Scenes Chamfer (m) ↓	Objaverse Scenes IoU ↑	Objaverse Scenes Chamfer (m) ↓
Voxel	N	N	0.198	0.014	0.324	0.018	0.336	0.024
PointSDF [20]	Y	N	**0.360**	**0.010**	0.460	0.015	0.347	0.025
V-PRISM (ours)	Y	Y	0.309	0.011	**0.500**	**0.012**	**0.464**	**0.018**

Table I: Quantitative experiments comparing our method to two baseline methods on procedurally generated scenes from benchmark mesh datasets.

Method	ShapeNet Scenes IoU ↑	ShapeNet Scenes Chamfer (m) ↓	YCB Scenes IoU ↑	YCB Scenes Chamfer (m) ↓	Objaverse Scenes IoU ↑	Objaverse Scenes Chamfer (m) ↓
V-PRISM w/ BHM Sampling	0.156	0.031	0.313	0.030	0.326	0.035
V-PRISM (ours)	**0.309**	**0.011**	**0.500**	**0.012**	**0.464**	**0.018**
V-PRISM w/o Under the Table	0.291	0.019	**0.500**	0.014	0.439	0.024
V-PRISM w/o Stratified Sampling	0.145	0.024	0.294	0.023	0.291	0.029

Table II: Ablation experiments on our negative sampling method.

level set other than $\tau = 0.5$ for constructing the mesh with the neural network. We found $\tau = 0.3$ to provide the best reconstructions for our version of PointSDF. For other methods, we use $\tau = 0.5$. We report the IoU and Chamfer distance in Table I. PointSDF outperforms other methods on the ShapeNet scenes, where the meshes are drawn from the same mesh dataset that it was trained on. On other datasets, our method outperforms PointSDF. This aligns with other work demonstrating that neural networks perform worse the further from the training distribution you get. Because our method has no reliance on a training distribution, it shows consistency across all datasets. Both our method and PointSDF consistently outperform the voxel baseline on most datasets and metrics. The only exception is Chamfer distance on Objaverse scenes, where the voxel baseline outperforms PointSDF. The performance of our method relative to our baselines indicate that our method results in accurate reconstructions

Our second experiment on simulated scenes ablates our negative sampling method. We observe the effect of removing sampling under the table plane and removing the stratified sampling along the ray. In order to remove the stratified sampling, we replace it with taking discrete, fixed steps along each ray instead. We also compare against the original BHM sampling method explained in [6], where there negative samples are drawn randomly along the whole ray instead of near objects. This is labeled as **BHM Sampling**. The IoU and Chamfer distance are reported in Table II. Our negative sampling method outperforms the others on each dataset and metric. This implies that our proposed sampling method does improve reconstruction quality when compared to alternatives.

The hyperparameters used for the simulated experiments are shown in Table III. These were kept constant across all procedurally generated datasets and corresponding experiments.

C. Real World Scenes

We evaluate our method by qualitatively comparing reconstructions on real world scenes. We use a Intel RealSense D415C camera to obtain point clouds of tabletop scenes. In order to get accurate segmentations of the scene, we use the Segment Anything Model (SAM) [3]. We compute

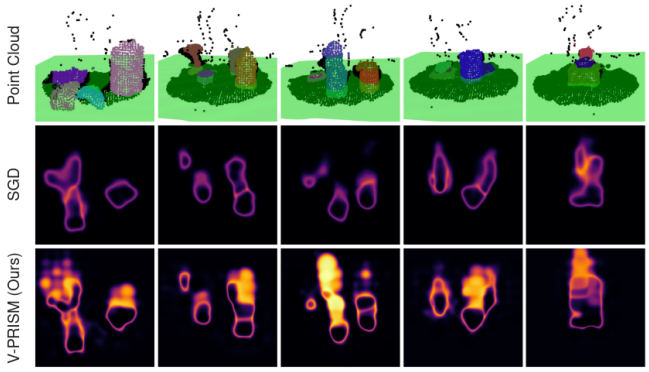

Figure 5: Qualitative comparison of uncertainty. **Top row:** the observed point cloud with a green plane corresponding to the 2D slice where the heat maps were calculated. We compare a non-probabilistic variant of V-PRISM trained with gradient descent (**middle row**) and our method (**bottom row**). In the heat maps, the bottom is closer to the camera and the top is farther from the camera. Lighter areas correspond to more uncertainty. Our method predicts high uncertainty in occluded areas of the scene.

Hyperparameters (Learning)	Value	Hyperparameters (Sampling)	Value (cm)
kernel type	Gaussian	grid length	5.0
kernel γ	1000	sampling r_{obj}	25.0
surface hinge pts.	32	subsample res. (objects)	1.0
iterations	3	subsample res. (empty)	1.5

Table III: Hyperparameters for experiments on procedurally generated scenes.

reconstructions on five scenes consisting of multiple objects. Each map of these scenes took between 2 and 5 seconds to compute. We compare our method to PointSDF. The qualitative comparison can be seen in Figure 6. Because these scenes are significantly more noisy than simulated scenes, PointSDF struggles to coherently reconstruct the scene. In contrast, our method is capable of producing quality reconstructions even with very noisy input point clouds. This suggests that our method is capable of bridging the sim to real gap and is robust to unknown, noisy scenes.

D. Principled Uncertainty

To show how our model captures uncertainty about the scene, we need a way to quantify uncertainty. We use the

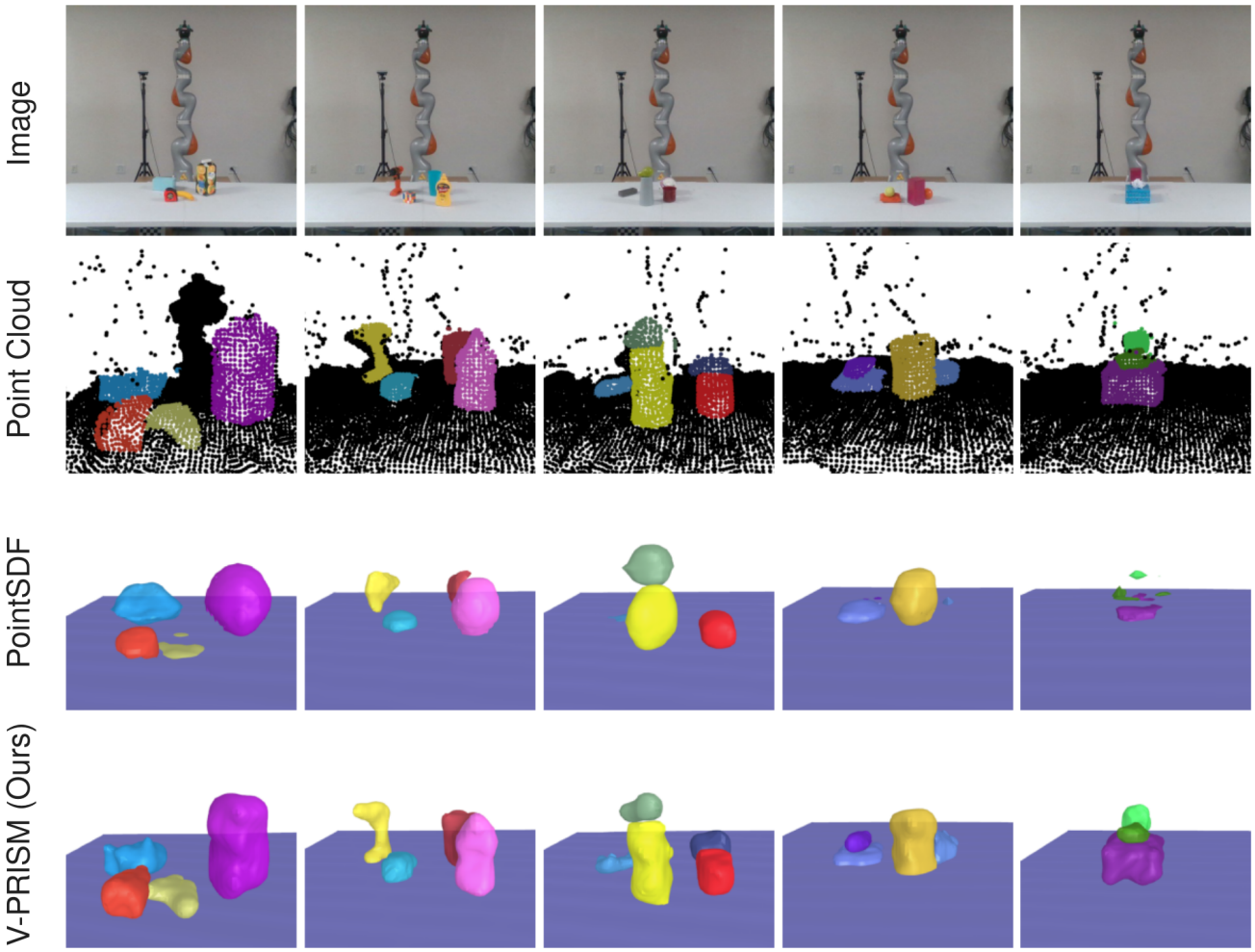

Figure 6: Qualitative comparisons with PointSDF reconstructions. **First row:** RGB images. **Second row:** the segmented point cloud used as input. **Third row:** PointSDF reconstructions. **Last row:** V-PRISM's (our method) reconstructions. V-PRISM results in quality reconstructions on noisy scenes.

entropy of our map at each point in space as a measurement of uncertainty:

$$H_m(\mathbf{x}) = -\sum_{k=1}^{c} \hat{P}(y = c|\mathbf{x}) \ln \hat{P}(y = c|\mathbf{x}).$$

This is maximized when the model predicts a uniform distribution over classes and minimized when the model predicts a single class with a probability of 1.

We compare our method with an alternate non-Bayesian version of our method, where we train a single weight vector with stochastic gradient descent (SGD) instead of the EM algorithm, to minimize the negative log-likelihood of our augmented data.

To visualize this uncertainty, we calculate this uncertainty over a 2D slice from each of our 5 real world scenes. The heat maps for each slice can be seen in Figure 5. Qualitatively, we can see that our method obtains high uncertainty values in occluded sections of the scene. This contrasts to the non-probabilistic model that does not accurately capture uncertainty about occluded regions. The heat maps showing occlusion-aware uncertainty suggest our model captures principled and accurate uncertainty measures.

VIII. Conclusion and Future Work

Principled uncertainty is necessary for the safety of many robotics tasks. We proposed a framework for robustly constructing multi-class 3D maps of tabletop scenes named V-PRISM. Our method works by iterating an EM algorithm on augmented data to produce a volumetric Bayesian segmentation map. To fully incorporate the information from received depth measurements of a tabletop scene, we proposed a novel negative sampling technique. The resulting map was shown to have desirable properties including quality reconstructions and accurate uncertainty measures through both quantitative experiments in simulation and qualitative experiments with real-world, noisy scenes. Future directions of this work include: (1) using our method's uncertainty to inform active learning; (2) extending V-PRISM to represent dynamic tabletop scenes.

REFERENCES

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[2] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

[3] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[4] C. Xie, Y. Xiang, A. Mousavian, and D. Fox, "Unseen object instance segmentation for robotic environments," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1343–1359, 2021.

[5] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[6] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning*, pp. 458–471, PMLR, 2017.

[7] W. Agnew, C. Xie, A. Walsman, O. Murad, Y. Wang, P. Domingos, and S. Srinivasa, "Amodal 3d reconstruction for robotic manipulation via stability and connectivity," in *Conference on Robot Learning*, pp. 1498–1508, PMLR, 2021.

[8] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart, "Signed distance fields: A natural representation for both mapping and planning," in *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*, University of Michigan, 2016.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[10] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.

[11] E. Sucar, S. Liu, J. Ortiz, and A. J. Davison, "imap: Implicit mapping and positioning in real-time," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6229–6238, 2021.

[12] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[13] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5501–5510, 2022.

[14] E. Sucar, K. Wada, and A. Davison, "Nodeslam: Neural object descriptors for multi-view shape reconstruction," in *2020 International Conference on 3D Vision (3DV)*, pp. 949–958, IEEE, 2020.

[15] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, 2023.

[16] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian, "Robust and accurate superquadric recovery: A probabilistic approach," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2676–2685, 2022.

[17] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *2011 IEEE International Conference on Robotics and Automation*, pp. 2845–2850, IEEE, 2011.

[18] S. Tulsiani, S. Gupta, D. F. Fouhey, A. A. Efros, and J. Malik, "Factoring shape, pose, and layout from the 2d image of a 3d scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 302–310, 2018.

[19] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

[20] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11516–11522, IEEE, 2020.

[21] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4460–4470, 2019.

[22] L. Rustler, J. Matas, and M. Hoffmann, "Efficient visuo-haptic object shape completion for robot manipulation," in *IEEE/RSG International Conference on Intelligent Robots and Systems (IROS)*, pp. 3121–3128, IEEE, 2023.

[23] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pp. 628–644, Springer, 2016.

[24] F. Williams, Z. Gojcic, S. Khamis, D. Zorin, J. Bruna, S. Fidler, and O. Litany, "Neural fields as learnable kernels for 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18500–18510, 2022.

[25] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu, "Learning to reconstruct shapes from unseen classes," *Advances in neural information processing systems*, vol. 31, 2018.

[26] N. Gothoskar, M. Cusumano-Towner, B. Zinberg, M. Ghavamizadeh, F. Pollok, A. Garrett, J. Tenenbaum, D. Gutfreund, and V. Mansinghka, "3dp3: 3d scene perception via probabilistic programming," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9600–9612, 2021.

[27] L. Li, S. Khan, and N. Barnes, "Silhouette-assisted 3d object instance reconstruction from a cluttered scene," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[28] T. S. Jaakkola and M. I. Jordan, "A variational approach to bayesian logistic regression models and their extensions," in *Sixth International Workshop on Artificial Intelligence and Statistics*, pp. 283–294, PMLR, 1997.

[29] R. Senanayake, A. Tompkins, and F. Ramos, "Automorphing kernels for nonstationarity in mapping unstructured environments.," in *CoRL*, pp. 443–455, 2018.

[30] W. Zhi, L. Ott, R. Senanayake, and F. Ramos, "Continuous occupancy map fusion with fast bayesian hilbert maps," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 4111–4117, IEEE, 2019.

[31] R. Senanayake and F. Ramos, "Building continuous occupancy maps with moving robots," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[33] G. Bouchard, "Efficient bounds for the softmax function and applications to approximate inference in hybrid models," in *NIPS 2007 workshop for approximate Bayesian inference in continuous/hybrid systems*, vol. 6, 2007.

[34] J. Daunizeau, "Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables," *arXiv preprint arXiv:1703.00091*, 2017.

[35] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, (New York, NY, USA), p. 163–169, Association for Computing Machinery, 1987.

[36] H. Thomas, *Learning new representations for 3D point cloud semantic segmentation*. PhD thesis, Université Paris sciences et lettres, 2019.

[37] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[38] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*, pp. 510–517, IEEE, 2015.

[39] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.