2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Uncertainty-aware 3D Object-Level Mapping with Deep Shape Priors

Ziwei Liao[*,1], Jun Yang[*,1], Jingxing Qian[*,1], Angela P. Schoellig[1,2], and Steven L. Waslander[1]

***Abstract*— 3D object-level mapping is a fundamental problem in robotics, which is especially challenging when object CAD models are unavailable during inference. We propose a framework that can reconstruct high-quality object-level maps for unknown objects. Our approach takes multiple RGB-D images as input and outputs dense 3D shapes and 9-DoF poses (including 3 scale parameters) for detected objects. The core idea is to leverage a learnt generative model for a category of object shapes as priors and to formulate a probabilistic, uncertainty-aware optimization framework for 3D reconstruction. We derive a probabilistic formulation that propagates shape and pose uncertainty through two novel loss functions. Unlike current state-of-the-art approaches, we explicitly model the uncertainty of the object shapes and poses during our optimization, resulting in a high-quality object-level mapping system. Moreover, the estimated shape and pose uncertainties, which we demonstrate can accurately reflect the true errors of our object maps, can be useful for downstream robotics tasks such as active vision. We perform extensive evaluations on indoor and outdoor real-world datasets, achieving substantial improvements over state-of-the-art methods. Our code is available at `https://github.com/TRAILab/UncertainShapePose`.**

I. INTRODUCTION

3D object-level mapping is an important problem in robotics. A challenging task is to reconstruct the shape and pose of objects in the scene observed with multiple RGB-D views. Compared to traditional approaches that employ low-level geometric primitives (e.g., points and voxels) [1], [2], [3], object-level mapping provides a rich representation of the scene and is extremely valuable for many downstream tasks, such as navigation, planning and manipulation [4], [5], [6]. Early approaches require pre-scanned CAD models for each object and then construct an object-level map by estimating an object pose for each CAD model [4], [7], [8], [9], [10], [11]. However, these works cannot generalize to previously unseen objects. When CAD models are unavailable, some works segment each object in the scene and reconstruct objects using multi-view depth fusion [12], [13]. These methods can reconstruct arbitrary shapes, but the reconstruction often remains incomplete as objects tend to be only partially visible during robotic operation.

In this work, we take advantage of the recent advances in deep learning for learnt shape representation to enhance the reconstruction of complete objects in object-level mapping [5], [14], [6]. These approaches reconstruct objects with both dense 3D models and their relative poses. While shape priors can be used for building object-level maps for unseen objects, most methods [5], [14], [6] only output a single deterministic estimation for each object's shape and pose. In contrast, for many robotic applications (e.g., robot grasping), capturing the underlying uncertainties associated with these outputs is critical. Moreover, as demonstrated in recent CAD-based methods, estimating uncertainties can improve the system performance and build high-quality object maps even in challenging scenes, where object symmetry and heavy occlusions exist [8], [11], [7].

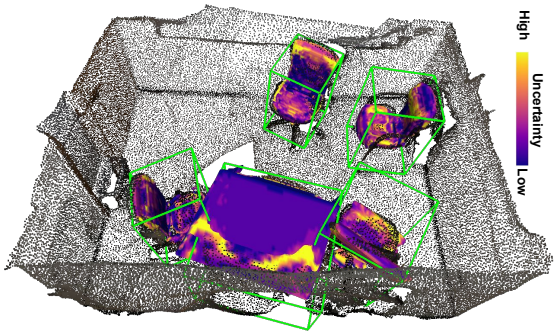

Fig. 1: Our approach takes RGB-D images as inputs and builds a 3D object-level map with dense object models, 9-DoF relative poses, and associated uncertainties.

The above findings motivate us to build an uncertainty-aware object-level mapping system to estimate the 3D model and pose for foreground objects. For each object category, we learn its shape distributions into a latent space through a 3D generative model [15]. The generative model can decode the input latent code to a detailed 3D object shape. During inference, we use the generative model to jointly optimize the latent code and the object's pose. To estimate uncertainties, we design a novel probabilistic optimization framework and use a combination of a 3D surface loss and a 2D rendering loss. The final outputs include dense 3D models, 9-DoF relative poses (3D translation, 3D rotation, and 3 scales along each axis), and associated state uncertainties for target objects. We summarize the following key contributions:

- A novel probabilistic optimization framework that leverages a learnt generative model as shape priors to jointly optimize objects' shapes, poses, and associated state uncertainties.
- Two probabilistic loss functions, a 3D surface loss and a 2D rendering loss, that propagate uncertainties from the shape priors and object pose in a differentiable way.
- An uncertainty-aware object-level mapping system that

*Equal contribution.

[1]The authors are with the University of Toronto Institute for Aerospace Studies and the University of Toronto Robotics Institute, Toronto, Canada. {ziwei.liao, jun.yang, jingxing.qian, steven.waslander}@robotics.utias.utoronto.ca

[2]The author is with the Technical University of Munich and the Munich Institute of Robotics and Machine Intelligence (MIRMI). angela.schoellig@tum.de

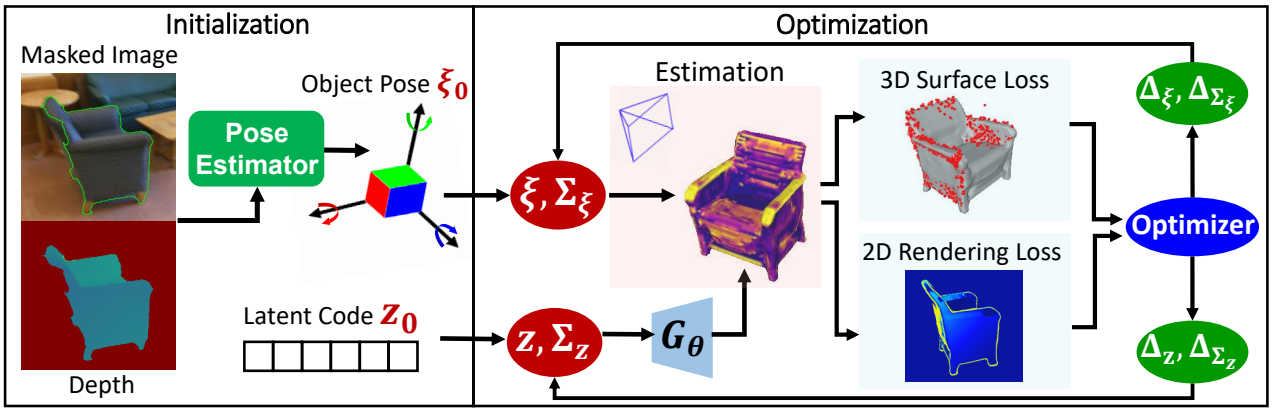

Fig. 2: An overview of the proposed uncertainty-aware object-level mapping system. We take the RGB-D images as the input and output the 3D models, 9-DoF poses, and the associated state uncertainties for the target unseen objects.

can recover the 3D model, 9-DoF pose, and the state uncertainties for target unseen objects from multi-view RGB-D observations.

We demonstrate our approach's high accuracy and reliable uncertainty estimation on two public real datasets. We will release the code of our approach and all baselines.

II. RELATED WORK

A. 3D Object-Level Mapping

Prior works tackle this task by either simultaneously localizing camera poses and mapping objects, known as object-level SLAM [4], [6], [7], [5], [16], or by reconstructing the objects with known camera poses [8], [9]. Early methods build object-level maps using pre-scanned CAD models [4], [17], [8], [11], [9], [7]. For example, the pioneering work, SLAM++ [4], builds a CAD model database in advance and estimates the 6D object poses. These approaches can reconstruct objects completely but cannot generalize to unknown objects outside the database. Recent works have dropped the requirement for CAD models and reconstructed the object directly by fusing multi-view depth maps [12], [13]. Although these approaches can reconstruct arbitrary objects, the reconstructed shape is usually incomplete. In our work, by exploiting a learnt shape prior, we build object-level maps with complete shapes and can generalize to unknown objects. We concentrate on the object mapping and assume camera poses are known in advance for convenience.

B. 3D Reconstruction with Shape Priors

With the recent advances in deep learning, many approaches leverage learnt generative models as shape priors to reconstruct objects [5], [14], [6]. Based on neural representations at the category level [18], [19], [20], [15], these methods can reconstruct unseen objects with complete detailed shapes. For example, NodeSLAM [5] uses a variational autoencoder [18] as the shape prior and builds an object-level SLAM system capable of reconstructing full dense shapes and relative object poses. DSP-SLAM is the most closely related approach to ours. It uses DeepSDF [15] as the shape prior and reconstructs the object-level map. While all these methods perform well when reconstructing unseen objects, they do not consider uncertainties underlying these estimates.

C. Uncertainty Estimation in Object-Level Mapping

In many robotic applications, it is important to estimate state uncertainties before taking action, such as robot grasping [21] and active perception [22], [23]. To this end, in many recent CAD-based approaches, the uncertainties are incorporated when estimating object poses [24], [25], [8], [7]. These works have demonstrated that modelling state uncertainties can significantly improve object-mapping performance. When leveraging shape priors for unseen objects, NodeSLAM [5] develops a rendering function to incorporate the uncertainty and improve the optimization. However, NodeSLAM does not output the uncertainty for estimated object shape or pose, limiting its use cases. In our work, we explicitly estimate uncertainties for every object's shape and pose and integrate them into our optimization framework.

III. METHODOLOGY

A. Approach Overview and Problem Formulation

We summarize our overall framework in Figure 2. Our framework takes as input multi-view RGB-D images and builds an object-level map of a scene. For each object, we aim to estimate its 3D model (a dense mesh), $\boldsymbol{Q}_o$, in the object canonical coordinate frame, O, and a 9-DoF relative pose, $\boldsymbol{T}_{ow} \in \mathbb{R}^{4\times4}$, from the global (world) coordinate frame, W, to the object coordinate frame.

Parametrization and Notations. We parameterize the object's shape with an optimizable latent shape code, $\boldsymbol{z} \in \mathbb{R}^{64}$, which can be passed through a decoder network, G_θ, to reconstruct its 3D canonical model, $\boldsymbol{Q}_o$. We employ the DeepSDF [15] as the shape decoder, which takes the latent shape code, $\boldsymbol{z} \in \mathbb{R}^{64}$, and a 3D query point, $\boldsymbol{p}_o \in \mathbb{R}^3$, under the object coordinate, O, as inputs, and returns the signed distance function (SDF) value, s, of the 3D point:

$$s = G_\theta(\boldsymbol{z}, \boldsymbol{p}_o) \tag{1}$$

The SDF represents the distance to the nearest object's surface, which can be converted to a mesh via Marching Cubes [26]. The decoder network, G_θ, was trained offline

on a large collection of CAD models [27], and the network weights, θ, are fixed during the online mapping inference. A 9-DoF pose, $\boldsymbol{T} \in \mathbb{R}^{4\times4}$, is constructed from a 3-DoF translation vector, $\boldsymbol{t} \in \mathbb{R}^3$, a 3-DoF rotation vector, $\boldsymbol{\phi} \in \mathfrak{so}(3)$, and a 3-DoF scaling vector, $\boldsymbol{s} \in \mathbb{R}^3$:

$$\boldsymbol{T} = \begin{bmatrix} \exp\left(\boldsymbol{\phi}^{\wedge}\right) & \boldsymbol{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathsf{diag}\left(\boldsymbol{s}\right) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2}$$

where $\exp\left(\cdot\right)$ is the exponential mapping from Lie Algebra space to the Lie Group space. The operator $\left(\cdot\right)^{\wedge}$ converts a vector to a skew-symmetric matrix. For simplicity, we combine the translation, rotation, and scaling vectors and represent the Lie Algebra space of the 9-DoF pose, $\boldsymbol{T} \in \mathbb{R}^{4\times4}$, with:

$$\boldsymbol{\xi} = [\boldsymbol{t}, \boldsymbol{\phi}, \boldsymbol{s}] \in \mathbb{R}^9 \tag{3}$$

We assume camera poses, $\boldsymbol{T}_{wc} \in SE(3)$, are known relative to the world frame. This can be achieved using off-the-shelf SLAM approaches with a hand-held camera [2], or robot kinematics when the camera is mounted on a robot arm [28].

Uncertainty Representation. With object depth measurements, $\boldsymbol{D}_{1:k}$, up to viewpoint k, our goal is to estimate the joint posterior distribution of the latent code and the object pose, $P\left(\boldsymbol{z}, \boldsymbol{\xi}_{ow} | \boldsymbol{D}_{1:k}\right)$. To represent the uncertainties, we formulate the distribution $P\left(\boldsymbol{z}, \boldsymbol{\xi}_{ow} | \boldsymbol{D}_{1:k}\right)$ with the following Gaussian distributions:

$$\boldsymbol{z} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}\right) \quad , \quad \boldsymbol{\xi}_{ow} \sim \mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}\right) \tag{4}$$

where $\left(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}\right)$ and $\left(\boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}\right)$ are the mean and covariance for the latent code, $\boldsymbol{z}$, and object pose, $\boldsymbol{\xi}_{ow}$, respectively. To simplify the problem, we assume that each dimension of $\boldsymbol{z}$ and $\boldsymbol{\xi}_{ow}$ is independent, which allows the covariance matrices $\boldsymbol{\Sigma}_{\boldsymbol{z}}$ and $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}$ to be diagonal.

Optimization Formulation. We estimate all parameters, $\mathbf{X} = \left\{\mu_{\boldsymbol{\xi}_{ow}}, \Sigma_{\boldsymbol{\xi}_{ow}}, \mu_{\boldsymbol{z}}, \Sigma_{\boldsymbol{z}}\right\}$, with a joint optimization formulation and solve it in an iterative manner:

$$\mathbf{X}^* = \underset{\mathbf{X}}{\operatorname{argmin}} \ \left(L_{3D} + L_{2D}\right) \tag{5}$$

In our approach, we propose two probabilistic losses, a 3D surface loss, L_{3D}, and a 2D rendering loss, L_{2D}. The 3D surface loss minimizes the distance between the 3D point cloud measurement and the object surface, but is insufficient to fully constrain the object's shape and pose. As illustrated in DSP-SLAM [6], the reconstructed object with 3D loss only may be significantly larger than its actual size in the case of partial observation. To address this issue, we introduce a novel probabilistic rendering loss function, which considers the uncertainty, to penalize shapes that grow outside the object mask and constrain its scale. Note that our optimization framework is agnostic to the particular 2D rendering function and can adapt to others [29], [5], [6].

In Section III-B, we first introduce a method to propagate the distribution of the latent code, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}\right)$, and the object pose, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}\right)$, to the SDF, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\sigma}_{\boldsymbol{s}}\right)$. Then, in Section III-C and Section III-D, we will describe how to use the propagated SDF distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\sigma}_{\boldsymbol{s}}\right)$, to compute the 3D and 2D losses.

B. Uncertainty Propagation

In our mapping system, each object has its latent code distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}\right)$, and the relative object pose distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}\right)$. Given a 3D point measurement, $\mathbf{p}_w$, defined in the world frame, W, we aim to estimate its SDF distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\sigma}_{\boldsymbol{s}}\right)$, in the object frame, O.

For a 3D point, $\mathbf{p}_w$, we compute its SDF mean, $\boldsymbol{\mu}_s$, by transforming it to the object frame, O, and then passing it through the decoder network, G_θ, following Equation 1:

$$\boldsymbol{\mu}_s = G_\theta\left(\boldsymbol{\mu}_{\boldsymbol{z}} \ , \ \mathbf{p}_o\right) = G_\theta\left(\boldsymbol{\mu}_{\boldsymbol{z}} \ , \ \boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}} \mathbf{p}_w\right) \tag{6}$$

To acquire the SDF variance, $\boldsymbol{\sigma}_s$, we linearize the entire system (Equation 6) and propagate the covariance of the latent code, $\boldsymbol{\Sigma}_{\boldsymbol{z}}$, and object pose, $\boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}$. Specifically, we derive the Jacobian of the SDF value with respect to the latent code, $\mathbf{J}_{\boldsymbol{z}}$, and object pose, $\mathbf{J}_{\boldsymbol{\xi}_{ow}}$, as:

$$\mathbf{J}_{\boldsymbol{z}} = \frac{\partial s}{\partial \boldsymbol{z}} = \frac{\partial G_\theta\left(\boldsymbol{z}, \boldsymbol{p}_o\right)}{\partial \boldsymbol{z}} \tag{7}$$

$$\mathbf{J}_{\boldsymbol{\xi}_{ow}} = \frac{\partial s}{\partial \boldsymbol{\xi}_{ow}} = \frac{\partial G_\theta\left(\boldsymbol{z}, \boldsymbol{p}_o\right)}{\partial \boldsymbol{\xi}_{ow}} = \frac{\partial G_\theta\left(\boldsymbol{z}, \boldsymbol{p}_o\right)}{\partial \boldsymbol{p}_o} \frac{\partial \boldsymbol{p}_o}{\partial \boldsymbol{\xi}_{ow}} \tag{8}$$

where the terms $\frac{\partial G(\boldsymbol{z}, \boldsymbol{p}_o)}{\partial \boldsymbol{z}}$ and $\frac{\partial G(\boldsymbol{z}, \boldsymbol{p}_o)}{\partial \boldsymbol{p}_o}$ can be obtained via the back-propagation of the decoder network, G_θ. With the linearization of Equation 6, the SDF variance, $\boldsymbol{\sigma}_s^2$, is finally acquired via the following forward-propagation:

$$\boldsymbol{\sigma}_s^2 = \left[\mathbf{J}_{\boldsymbol{z}} \ , \ \mathbf{J}_{\boldsymbol{\xi}_{ow}}\right] \begin{bmatrix} \boldsymbol{\Sigma}_{\boldsymbol{z}} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}} \end{bmatrix} \left[\mathbf{J}_{\boldsymbol{z}} \ , \ \mathbf{J}_{\boldsymbol{\xi}_{ow}}\right]^T \tag{9}$$

C. Uncertainty-aware 3D Surface Loss

We construct the 3D surface loss, L_{3D}, by aligning the object's depth measurements with the SDF field of the target object model. For each pixel's depth, we compute its SDF distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\sigma}_{\boldsymbol{s}}\right)$, for the optimization.

Given the input depth image, $\boldsymbol{D}_k(\mathbf{u})$, from the k^{th} camera frame, we first segment the object's mask, $\boldsymbol{V}_k$, and obtain the object's point cloud, $\mathbf{P}_{c,k}$, via back-projection:

$$\boldsymbol{P}_{c,k} = \left\{\boldsymbol{D}_k(\mathbf{u}) \, \boldsymbol{K}^{-1} \, \mathbf{u}^T \ , \ \mathbf{u} \in \boldsymbol{V}_k\right\} \tag{10}$$

where $\boldsymbol{K}$ denotes the camera intrinsic matrix and $\mathbf{u}$ represents the pixel from the object mask. The multi-view acquired point cloud is finally transformed to the global world frame, W, with the known camera poses, $\boldsymbol{T}_{wc,k}$:

$$\boldsymbol{P}_w = \left\{\boldsymbol{T}_{wc,k} \, \boldsymbol{P}_{c,k} \ , \ k = 1 : K\right\} \tag{11}$$

where $\mathbf{P}_w$ is the point cloud in the world coordinate.

Ideally, the point cloud, $\mathbf{P}_w$, should align perfectly with the object surface, leading to a zero SDF mean, $\boldsymbol{\mu}_{\boldsymbol{s}} = 0$, when applying the Equation 6 on each 3D point, $\mathbf{p}_w \in \mathbf{P}_w$. For a 3D point, we measure the loss between the SDF distribution, $\mathcal{N}\left(\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\sigma}_{\boldsymbol{s}}\right)$, and the target measurement (zero SDF value). Following our previous work for the shape reconstruction only [30], we use the Energy Score (ES),

which shows to be a proper scoring rule [31]. We use it with the Monte Carlo approximation:

$$\mathrm{ES}_{3D} = \frac{1}{M}\sum_{m=1}^{M}\|\mathbf{s}_m - \bar{\mathbf{s}}\| - \frac{1}{2(M-1)}\sum_{m=1}^{M-1}\|\mathbf{s}_m - \mathbf{s}_{m+1}\| \tag{12}$$

where $\bar{\mathbf{s}} = 0$ is the target SDF, and $\mathbf{s}_m$ denote the m^{th} $i.i.d$ sample from the SDF distribution, $\mathcal{N}(\boldsymbol{\mu}_s, \boldsymbol{\sigma}_s)$. We set $M = 1000$ in the optimization with very little computational overhead. Considering a point cloud that has N points, we compute the energy score, $\mathrm{ES}_{3D,n}$, for each 3D point measurement, and acquire its 3D loss, L_{3D}, with:

$$L_{3D} = \frac{1}{N}\sum_{n=1}^{N}\mathrm{ES}_{3D,n} \tag{13}$$

D. Probabilistic Differentiable Rendering

We design our 2D loss function via the differentiable SDF rendering, illustrated in Figure 3. Compared to most previous approaches [6], [29], which are deterministic, our rendering function, $R(\cdot)$, optimizes the object shape, pose, and state uncertainties from rendered views. It takes as input the distributions of latent code, $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}})$, object pose, $\mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}})$, and the known camera pose, $\boldsymbol{T}_{wc}$, and renders a depth map with uncertainties:

$$\hat{\boldsymbol{D}}_{\mu}\ ,\ \hat{\boldsymbol{D}}_{\sigma} = R\left(\boldsymbol{\mu}_{\boldsymbol{z}}, \boldsymbol{\Sigma}_{\boldsymbol{z}}, \boldsymbol{\mu}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{\Sigma}_{\boldsymbol{\xi}_{ow}}, \boldsymbol{T}_{wc}\right) \tag{14}$$

where $\hat{\boldsymbol{D}}_{\mu}$ and $\hat{\boldsymbol{D}}_{\sigma}$ are the rendered depth map and uncertainties from viewpoint, $\boldsymbol{T}_{wc}$. In the following sections, we describe how to obtain pixel depths and uncertainties, which will be used for computing the 2D loss.

*1) **SDF Sampling**:* Following [5], [6], we develop our SDF renderer using differentiable ray-tracing. For each pixel, $\mathbf{u}$, we uniformly sample $\mathcal{M}$ points along the back-projected ray, $\mathbf{r}$, within the depth range $\left[\hat{d}_{min}, \hat{d}_{max}\right]$. We denote each sampled depth with $\hat{d}_i = \hat{d}_{min} + \frac{i}{M}\left(\hat{d}_{max} - \hat{d}_{min}\right)$. The corresponding point under camera frame is $\mathbf{p}_i^c = \mathbf{o} + \hat{d}_i\mathbf{r}$, with $\mathbf{o}$ being camera optical center. For a single point, $\mathbf{p}^c$, we approximate its SDF with a Gaussian, $s \sim \mathcal{N}(\mu_s, \sigma_s^2)$, by transforming it to the world frame, W. To achieve better efficiency and accuracy, as in DSP-SLAM [6], we only consider sampled points within a small fixed offset to the predicted surface $|\mu_s| <= \delta$.

*2) **Occupancy Probability Estimation**:* To utilize SDF in the volumetric rendering process, a common practice [5], [6] is to convert the SDF prediction, s, to an occupancy probability, o, with a hard sigmoid function. In comparison, we further model the occupancy probability into a distribution. We propose to transform the SDF distribution, $s \sim \mathcal{N}(\mu_s, \sigma_s^2)$, through a smooth mirrored sigmoid function with a slope parameter l:

$$o \coloneqq \mathrm{sigmoid}(-ls) = \frac{1}{1+\exp(ls)} \tag{15}$$

where the l value encodes the cut-off threshold and controls the smoothness of the transition. We use $l = 400$ in our implementation such that the cut-off threshold is around 0.01 meters as in [6]. When the Gaussian SDF is passed through the sigmoid function, the resulting occupancy follows the logit-normal distribution with the following continuous density function:

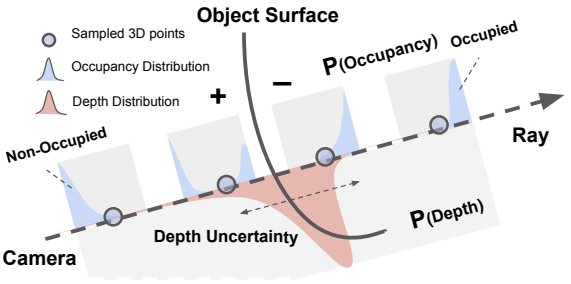

Fig. 3: Probabilistic differentiable rendering. We model uncertainty into the occupancy and termination probability of each sampled 3D point along a camera ray. Then, we generate a depth distribution of this ray (Sec. III-D).

$$p(o \mid \mu_s, \sigma_s^2) = \frac{1}{o(1-o)l\sqrt{2\pi\sigma_s^2}}\exp\left(-\frac{\left(\frac{-\mathrm{logit}(o)}{l} - \mu_s\right)^2}{2\sigma_s^2}\right) \tag{16}$$

where the logit-normal density function, $p(o \mid \mu_s, \sigma_s^2)$, is defined for $o \in (0,1)$ and `logit` represents the logit function. We illustrate the logit-normal occupancy distribution in Figure 3.

*3) **Termination and Escape Probability**:* When tracing points along the ray, $\mathbf{r}$, the ray either ends on a surface point or escapes without hitting any point. Following previous works [5], [6], we compute the termination probability, ϕ_i, for each sampled point, $\mathbf{p}_i^c$, and the escape probability, ϕ_{M+1}, for a camera ray, $\mathbf{r}$:

$$\begin{aligned} p(\phi_i) &= p(o_i \mid \mu_{s,i}, \sigma_{s,i})\prod_{j=1}^{i-1}p(1-o_j \mid \mu_{s,j}, \sigma_{s,j}) \\ &= p(o_i \mid \mu_{s,i}, \sigma_{s,i})\prod_{j=1}^{i-1}p(o_j \mid -\mu_{s,j}, \sigma_{s,j}), i = 1, \ldots, M \\ p(\phi_{M+1}) &= \prod_{j=1}^{M}p(1-o_j \mid \mu_{s,j}, \sigma_{s,j}) = \prod_{j=1}^{M}p(o_j \mid -\mu_{s,j}, \sigma_{s,j}) \end{aligned} \tag{17}$$

where $p(1-o_j \mid \mu_s, \sigma_s) = p(o_j \mid -\mu_s, \sigma_s)$ represents the symmetry property of logit-normal distributions. Since the logit-normal distribution is not closed under multiplication, we propose to approximate the termination probability distributions via Quasi-Monte Carlo. Specifically, given the quantile function (the inverse cumulative distribution function), $Q(\cdot)$, of the logit-normal distribution with slope l:

$$Q(o_i \mid \mu_{s,i}, \sigma_{s,i}) = \mathrm{sigmoid}\left(l\sqrt{2\sigma_{s,i}^2}\mathrm{erf}^{-1}(2o_i - 1) - l\mu_{s,i}\right), \tag{18}$$

where erf is the error function [32], we estimate the the event posterior mean $\mu_{\phi,i}$, $\mu_{\phi,M+1}$ and variance $\sigma_{\phi,i}$, $\sigma_{\phi,M+1}$ from the SDF means and variances with the inverse transform method [33], [34]. We use 128 samples from the Sobol sequence [35] in our implementation.

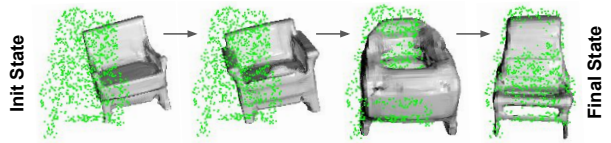

Fig. 4: The optimization process of the shape and pose of a chair instance from ScanNet with different iterations.

*4) **Depth Estimation and 2D Loss Term**:* With the estimated event probabilities, we acquire the depth distribution by computing the expectation, $\hat{\mu}_d$, and variance, $\hat{\sigma}_d^2$, over the sampled points. As in 3D loss, for each pixel $\mathbf{u}$, with the rendered depth mean $\hat{\mu}_d$, and variance, $\hat{\sigma}_d^2$, we compute its 2D loss, $\mathrm{ES}_{2D,\mathbf{u}}$, using the energy score (Equation 12). The final rendering term is defined as:

$$L_{2D} = \frac{1}{|\boldsymbol{V}_s|} \sum_{\mathbf{u} \in \boldsymbol{V}_s} \mathrm{ES}_{2D,\mathbf{u}} \tag{19}$$

where $\boldsymbol{V}_s = \boldsymbol{V}_o \cup \boldsymbol{V}_b$ is the union of object surface pixels, $\boldsymbol{V}_o$, and background pixels, $\boldsymbol{V}_b$. Object surface pixels, $\boldsymbol{V}_o$, are the set of pixels from the object's mask. The background pixels, $\boldsymbol{V}_b$, are not on object surfaces but inside the object's 2D bounding box. Following [5], [6], we assign background pixels a depth of $\hat{d}_{M+1} = 1.1\hat{d}_{max}$.

E. Optimization

Our final loss is the weighted sum of the 3D loss, L_{3D}, 2D losses, L_{3D} and a shape code regularization term, $||\mathbf{z}||$:

$$\mathbf{L}_{final} = \lambda_s L_{3D} + \lambda_r L_{2D} + \lambda_c ||\mathbf{z}||^2 \tag{20}$$

where λ_s, λ_r and λ_c are the weights for each loss term. In our approach, we initialize the shape prior with a code $\mathbf{z} = \mathbf{0}$. The initial object pose, $\mathbf{T}_{ow}$, is obtained by matching the initial object shape (corresponds to the code $\mathbf{z} = \mathbf{0}$) to the object point cloud. We generate 18 hypotheses by rotating the initial object shape around the $Z-$axis with a constant interval. ICP is then applied to each hypothesis and the solution with the minimum point-to-point error becomes the initial pose. Note that the pose initialization can be replaced with any 3D object detector. We initialize the covariance matrix for shape code, $\Sigma_{\mathbf{z}}$, and object pose, $\Sigma_{\boldsymbol{\xi}_{ow}}$, by placing a constant value on their diagonal elements. In our implementation, we set $1 \times e^{-6}$ and $1 \times e^{-4}$ for $\Sigma_{\mathbf{z}}$ and $\Sigma_{\boldsymbol{\xi}_{ow}}$, respectively. In our work, we use the Adam optimizer [36] to solve the optimization problem. We use 0.005 learning rate and run the optimizer for 200 iterations. Figure 4 shows an example of our optimization with different iterations.

IV. EXPERIMENTS

We evaluate our framework on two real-world datasets, indoor ScanNet [37] and outdoor KITTI-3D [38]. The ScanNet dataset provides RGB-D video sequences of multiple objects in complex indoor scenes. The KITTI-3D dataset includes different vehicles in outdoor environments and was captured with a synchronized RGB camera and a LIDAR sensor.

Quantitatively, we compare our framework with the most closely related approaches, DSP-SLAM [6] and Node-SLAM [5]. DSP-SLAM estimates the object model and 7-DoF pose with deterministic 3D and 2D loss functions.

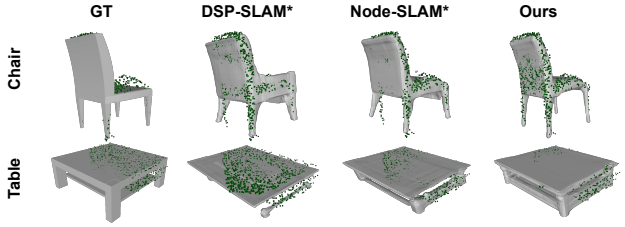

Fig. 5: Qualitative results on the ScanNet dataset.

Views	Methods	Chair			Table		
		9-DoF Pose	IoU>0.25	CD<0.2	9-DoF Pose	IoU>0.25	CD<0.2
1	Node-SLAM*	0.200	0.487	0.555	0.098	0.252	0.359
	DSP-SLAM*	0.224	0.603	0.607	0.105	0.329	0.380
	Ours w/ Det2D	0.225	0.631	0.636	0.103	**0.339**	0.380
	Ours w/ Samp2D	**0.226**	0.621	0.633	**0.140**	0.316	**0.439**
	Ours	0.214	**0.635**	**0.655**	0.123	0.332	0.395
2	Node-SLAM*	0.202	0.505	0.576	0.117	0.248	0.374
	DSP-SLAM*	0.228	0.637	0.634	0.130	0.324	0.382
	Ours w/ Det2D	**0.236**	0.660	0.662	0.119	0.345	0.396
	Ours w/ Samp2D	0.231	0.648	**0.666**	**0.144**	0.328	**0.432**
	Ours	0.224	**0.669**	0.656	0.138	**0.360**	0.430
3	Node-SLAM*	0.215	0.503	0.601	0.140	0.246	0.390
	DSP-SLAM*	0.239	0.657	0.647	0.134	0.338	0.398
	Ours w/ Det2D	**0.253**	**0.696**	**0.695**	0.128	**0.349**	0.417
	Ours w/ Samp2D	0.252	0.675	0.692	**0.143**	0.339	0.426
	Ours	0.226	0.682	0.662	0.132	0.332	**0.449**

TABLE I: Quantitative results on the ScanNet dataset.

To fairly compare, we extend the DSP-SLAM open-source codebase to output a 9-DoF object pose, which we refer to as *DSP-SLAM**. Node-SLAM is not open-sourced, so we implement it from scratch and follow its original design of using only the 2D rendering loss. Node-SLAM measures the rendered depth uncertainties by computing the sampled depth variance along the camera ray, and minimizes the NLL loss. In our experiments, we notice that the NLL loss has limited numerical stability, which makes it difficult to find converging hyperparameters. We therefore implement Node-SLAM with the energy score as a loss, named as *Node-SLAM**. Since our optimization framework is adaptable to any 2D rendering function, we additionally implement our approach with two variants by exploiting other rendering functions: deterministic rendering (from DSP-SLAM) and sampling-based rendering (from Node-SLAM). We refer to them as *Ours w/ Det2D* and *Ours w/ Samp2D*, respectively. For all the baselines, variants, and our method, we use DeepSDF as the shape model and Adam [36] as the optimizer. We provide the same inputs, including depth, 2D masks, camera poses, initial object poses, and latent code for all comparisons.

A. Results on ScanNet Dataset

We perform the evaluation on two common categories, chair and table, with all video sequences. We use 954 chair instances and 256 table instances from the ScanNet dataset [37]. For each object, it provides the ground truth CAD model and 9-DoF object pose [39]. We visualize the reconstruction results in Figure 5. Compared to the baselines, our approach reconstructs the objects with far fewer artifacts and better alignment to the point cloud.

For the quantitative evaluation, we calculate the correct detection rate with three metrics: absolute 9-DoF pose, Intersection over Union (IoU) and Chamfer Distance (CD). The 9-DoF pose metric considers an object reconstruction correct if the pose error is within thresholds of 0.2-meter translation,

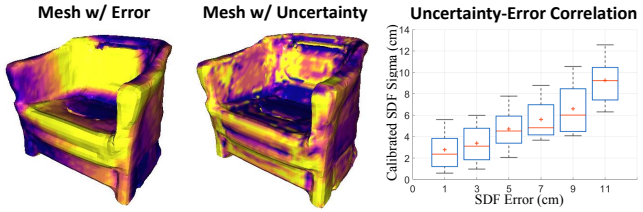

Fig. 6: Uncertainty and error correlation. Our estimated uncertainty correlates with the true reconstruction errors.

Methods	Ours w/ Det2D	Ours w/ Samp2D	Ours
Chair	0.738	0.740	**0.749**
Table	0.661	0.668	**0.742**

TABLE II: Uncertainty evaluation on the ScanNet dataset with the Pearson correlation score metric.

20-degree rotation, and 20%-scale. We use IoU with 0.25 and CD with 0.2 meters as thresholds for the other two metrics. For each category, we evaluate the correct detection rate with 1, 2, and 3 viewpoints. We show quantitative results in Table I. One of our approaches ranks first on all numbers of views across all three reconstruction metrics. This shows the benefits of our uncertainty-aware framework for reconstructing the object shape and relative poses. It is noteworthy that our approach exceeds other variants on the single-view test set, where the task is challenging due to incomplete depth and occlusions. However, with more available views, the performance improvement is less noticeable when compared with other variants. This is likely due to using a uni-modal Gaussian distribution to approximate the actual depth distribution, which is usually multi-modal. This phenomenon is more obvious in multi-view cases. We consider using a more advanced modelling technique, such as Gaussian mixture model [11], [10], as future work.

Although the accuracy improvement on multi-view cases is less obvious when using our rendering function, its probabilistically complete rendering process brings us a better ability to estimate the uncertainty in pose and shape. To evaluate the estimated uncertainties, we first generate GT depth data by sampling $10K$ points from the GT object model under the world frame, W. For each point, we transform it to the object frame, O, using the estimated object pose, $\boldsymbol{\xi}_{ow}$, and compute the SDF mean and variance with the estimated latent code, $\boldsymbol{z}$. A well-estimated variance should correlate to the actual SDF error. Figure 6 first shows a qualitative example of our estimated SDF variance, which correlate well with the true SDF errors. Quantitatively, we evaluate this correlation against our variants with the Pearson correlation score. We evaluate on well-reconstructed objects under the 3-view setup, including 145 chairs and 17 tables. Table II shows the uncertainty evaluation on the ScanNet dataset. Our approach achieves higher Pearson scores than the other two variants, demonstrating the effectiveness of our probabilistic rendering for uncertainty estimation.

B. Results on KITTI Dataset

We use the KITTI-3D object detection dataset [38] to evaluate the system performance in the outdoor environment.

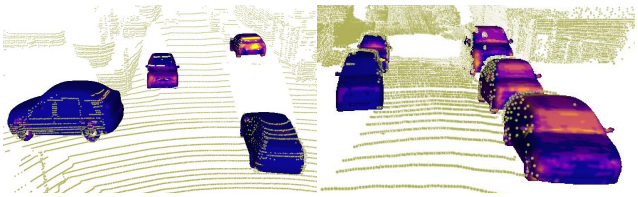

Fig. 7: Qualitative results on KITTI-3D.

Methods	Mean IoU	IoU>0.6	IoU>0.75
Node-SLAM*	0.677	78.6	21.4
DSP-SLAM*	0.690	80.2	25.7
Ours w/ Det2D	0.721	88.9	42.3
Ours w/ Samp2D	**0.741**	**93.9**	53.7
Ours	0.738	90.5	**54.9**

TABLE III: Quantitative results on KITTI-3D.

We obtain initial object poses using the PointPillars 3D detector [40] and acquire object masks from the Mask2Former segmentation algorithm [41]. To investigate the upper bound of reconstruction accuracy of each method, we consider an object for evaluation only if its initial 3D detection error is within thresholds of 1.0-meter translation and 20-degree heading. In total, we performed the evaluation on 253 vehicles from the validation set.

Since object shape annotation is not available, we evaluate the performance with the IoU metric in Table III. Our approaches, including two variants, exceed the baselines, *DSP-SLAM** and *Node-SLAM**, by a large margin. We also notice that our approach exceeds other variants when using a more strict metric (IoU>0.75). Further qualitative results in Figure 7 demonstrate the high performance when recovering the shape and relative poses for different vehicles, even with partial Lidar observations.

C. Computation Analysis

We ran the experiments on a 16GB V100 GPU. It costs 0.2s with each iteration for our approach. Depending on the task, the shape resolution (currently 64^3), the sampling number, and the iteration steps can be modified to trade-off between the efficiency and accuracy.

V. CONCLUSION

We presented an object-level mapping approach that can recover the 3D dense models and poses for unknown objects. The core idea is leveraging a learnt category-specific shape prior to formulate an uncertainty-aware optimization framework. We introduce two probabilistic loss functions that model the uncertainties of shape and pose in the optimization. We compare our approach against the state-of-the-art approaches on challenging real-world datasets, ScanNet and KITTI-3D. The results demonstrate that our approach can reconstruct higher-quality object-level maps. Moreover, our estimated uncertainties accurately correlate with the true errors of the estimated shapes and poses, which is valuable for downstream robotic applications. This work represents an important step toward our future development of an uncertainty-aware object-level SLAM system that jointly estimates camera poses and actively selects camera viewpoints for building detailed object-level maps in the open world.

REFERENCES

[1] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*, pp. 834–849, Springer, 2014.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*, pp. 127–136, Ieee, 2011.

[4] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1352–1359, 2013.

[5] 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.

[6] J. Wang, M. Rünz, and L. Agapito, "Dsp-slam: Object oriented slam with deep shape priors," in *2021 International Conference on 3D Vision (3DV)*, pp. 1362–1371, IEEE, 2021.

[7] N. Merrill, Y. Guo, X. Zuo, X. Huang, S. Leutenegger, X. Peng, L. Ren, and G. Huang, "Symmetry and uncertainty-aware object slam for 6dof object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14901–14910, 2022.

[8] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, 2021.

[9] K.-K. Maninis, S. Popov, M. Nießner, and V. Ferrari, "Vid2cad: Cad model alignment using multi-view constraints from videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 45, no. 1, pp. 1320–1327, 2022.

[10] J. Yang, W. Xue, S. Ghavidel, and S. L. Waslander, "6d pose estimation for textureless objects on rgb frames using multi-view optimization," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2905–2912, IEEE, 2023.

[11] J. Fu, Q. Huang, K. Doherty, Y. Wang, and J. J. Leonard, "A multi-hypothesis approach to pose ambiguity in object-based slam," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7639–7646, IEEE, 2021.

[12] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric object-level slam," in *2018 international conference on 3D vision (3DV)*, pp. 32–41, IEEE, 2018.

[13] M. Runz, M. Buffier, and L. Agapito, "Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects," in *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pp. 10–20, IEEE, 2018.

[14] M. Runz, K. Li, M. Tang, L. Ma, C. Kong, T. Schmidt, I. Reid, L. Agapito, J. Straub, S. Lovegrove, *et al.*, "Frodo: From detections to 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14720–14729, 2020.

[15] 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.

[16] Z. Liao, Y. Hu, J. Zhang, X. Qi, X. Zhang, and W. Wang, "So-slam: Semantic object slam with scale proportional and symmetrical texture constraints," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4008–4015, 2022.

[17] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3343–3352, 2019.

[18] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[19] M. Vakalopoulou, G. Chassagnon, N. Bus, R. Marini, E. I. Zacharaki, M.-P. Revel, and N. Paragios, "Atlasnet: Multi-atlas non-linear deep networks for medical image segmentation," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part IV 11*, pp. 658–666, Springer, 2018.

[20] 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.

[21] J. Lundell, F. Verdoja, and V. Kyrki, "Robust grasp planning over uncertain shape completions," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1526–1532, IEEE, 2019.

[22] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5986–5992, IEEE, 2019.

[23] J. Yang, J. Yao, and S. L. Waslander, "Active pose refinement for textureless shiny objects using the structured light camera," *arXiv preprint arXiv:2308.14665*, 2023.

[24] S. Peng, Y. Liu, Q. Huang, X. Zhou, and H. Bao, "Pvnet: Pixel-wise voting network for 6dof pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4561–4570, 2019.

[25] B. Okorn, M. Xu, M. Hebert, and D. Held, "Learning orientation distributions for object pose estimation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10580–10587, IEEE, 2020.

[26] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Seminal graphics: pioneering efforts that shaped the field*, pp. 347–353, 1998.

[27] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[28] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, "Self-supervised 6d object pose estimation for robot manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3665–3671, IEEE, 2020.

[29] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2626–2634, 2017.

[30] Z. Liao and S. L. Waslander, "Multi-view 3d object reconstruction and uncertainty modelling with neural shape prior," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3098–3107, 2024.

[31] A. Harakeh and S. L. Waslander, "Estimating and evaluating regression predictive uncertainty in deep object detectors," *arXiv preprint arXiv:2101.05036*, 2021.

[32] M. Abramowitz, I. A. Stegun, and D. Miller, "Handbook of mathematical functions with formulas, graphs and mathematical tables (national bureau of standards applied mathematics series no. 55)," *Journal of Applied Mechanics*, vol. 32, pp. 239–239, 1965.

[33] R. Y. Rubinstein and D. P. Kroese, *Simulation and the Monte Carlo method*. John Wiley & Sons, 2016.

[34] S. Raychaudhuri, "Introduction to monte carlo simulation," in *2008 Winter simulation conference*, pp. 91–100, IEEE, 2008.

[35] I. M. Sobol, "On the distribution of points in a cube and the approximate evaluation of integrals," *Ussr Computational Mathematics and Mathematical Physics*, vol. 7, pp. 86–112, 1967.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[37] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

[38] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*, pp. 3354–3361, IEEE, 2012.

[39] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2cad: Learning cad model alignment in rgb-d scans," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 2614–2623, 2019.

[40] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12697–12705, 2019.

[41] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention mask transformer for universal image segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1290–1299, 2022.