Grasping Under Uncertainties: Sequential Neural Ratio Estimation for 6-DoF Robotic Grasping

Norman Marlier, Olivier Brüls, and Gilles Louppe

***Abstract*—We introduce a novel approach to 6-DoF robotic grasping based on simulation-based inference. Our approach combines sequential neural ratio estimation with a neural implicit representation for the Bayesian inference of hand configurations in cluttered environments. We propose to compute the maximum a posteriori by gradient descent, more specifically using Riemannian gradient descent, to preserve the geometry of the rotation space and capitalize on the full differentiability of our model. We demonstrate the capabilities of our approach on a grasping benchmark both in simulation and on a real robot. Our performance generalizes well across different scenarios, achieving high success rates.**

***Index Terms*—Deep learning in grasping and manipulation, planning under uncertainty, probabilistic inference.**

I. Introduction

GRASPING is a fundamental skill for robots. While industrial robots perform tasks in very controlled environments, novel applications require robots to adapt to new unstructured environments. Determining robust grasp poses from raw perception data is, for this reason, essential to the wider deployment of robots in the real world. However, dealing with uncertainties arising from rich nonsmooth contact dynamics and sensor noise is still an open problem. To address these challenges, Bayesian inference provides a principled framework to recast grasping as an inference problem under uncertainties. The nature of robotic grasping, however, often involves highly complex dynamics relating the tentative grasp pose to the outcome, making the necessary likelihood function intractable. Additionally, grasp poses in these tasks come with hard constraints from the robot's kinematics, further complicating the inference procedure.

In this paper, we tackle the problem of generating robust grasp poses by proposing an original approach based on simulation-based inference algorithms [1], a family of methods that learn a component of the Bayes rule through stochastic forward simulations. Our contributions are summarized as follows:

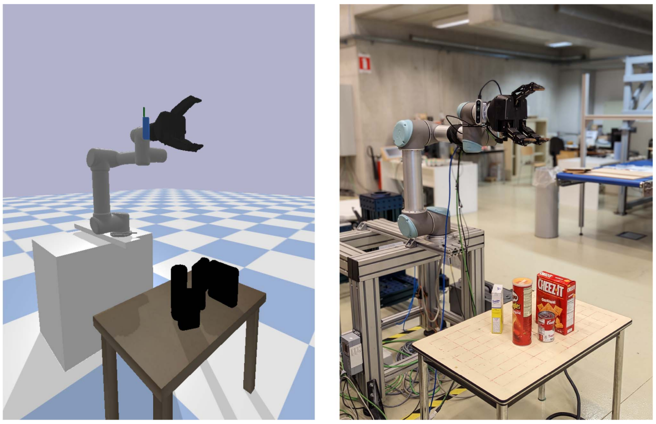

Fig. 1. Our benchmark scene. (left) The simulated environment. (right) The real setup.

- We bring simulation-based Bayesian inference methods to robotic grasping. By sequentially learning a model for the likelihood-to-evidence ratio and using an implicit neural representation, we derive an amortized and differentiable posterior for grasp poses.
- We make use of Riemannian manifold methods to sample from densities defined on smooth Riemannian manifolds and optimize the grasp poses.
- We validate the effectiveness of our method through both simulated and real experiments, demonstrating remarkable grasping performance.

II. Problem Statement

We consider the problem of planning 6-DoF hand configurations for a robotic gripper handling various unknown objects on a table, observed with a depth camera. A benchmark scene is shown in Fig. 1.

A. Description

The robot arm, which consists of 6 or 7 DoF, operates a gripper within a cubic workspace of size l, featuring a planar tabletop. The scene is observed with a depth camera mounted on the robot flange. Our objective is to identify the most plausible hand configuration conditioned by a grasp success and the observed point cloud. Then, a path planner computes a collision-free joint trajectory, enabling the robot to reach the desired grasp pose and safely remove the selected object from the tabletop.

Manuscript received 18 January 2024; accepted 31 May 2024. Date of publication 19 June 2024; date of current version 28 June 2024. This letter was recommended for publication by Associate Editor W. Yu and Editor A. Faust upon evaluation of the reviewers' comments. This work was supported in part by the Belgian Fund for Research training in Industry and Agriculture (FRIA) and in part by the Consortium des Équipements de Calcul Intensif (CÉCI) . *(Corresponding author: Norman Marlier.)*

The authors are with the Faculty of Applied Science, University of Liège, 4000 Liège, Belgium (e-mail: norman.marlier@uliege.be).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3416773, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3416773

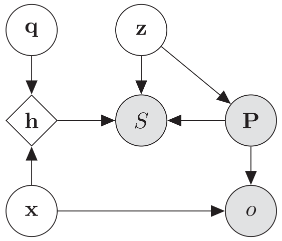

Fig. 2. Probabilistic graphical model of the environment. Gray nodes represent observed variables, white nodes represent unobserved variables, and diamond nodes represent deterministic functions.

B. Notations

Frames: We use several reference frames. The world frame $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{W}}$ and the workspace frame $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{S}}$ can be chosen freely and are not tied to a physical location. The world frame is used for the robot and the sensor, while the workspace frame is used for our inference system. $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{C}}$ and $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{E}}$ correspond respectively to the camera and the end-effector frames.

Hand configuration: The hand configuration $\mathbf{h} \in \mathcal{H} = \mathbb{R}^3 \times \mathbb{S}^3$ is defined as the pose $(\mathbf{x}, \mathbf{q}) \in \mathbb{R}^3 \times \mathbb{S}^3$ of the hand, where $\mathbf{x}$ is the vector $\overrightarrow{SE}$ expressed in $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{S}}$ and $\mathbf{q}$ is the 3D rotation from $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{S}}$ to $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{E}}$ represented using unit quaternions with scalar last format.

Binary metric: A binary variable $S \in \{0, 1\}$ indicates if the grasp fails ($S = 0$) or succeeds ($S = 1$).

Observation: Given the depth image I with its corresponding transformation camera to world $\mathbf{T}_{\mathrm{WC}}$ and camera intrinsic matrix K, we construct a point cloud $\mathbf{P} \in \mathbb{R}^{2048\times 3}$ expressed in $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{S}}$.

Occupancy: A binary variable $o \in \{0, 1\}$ indicates if a point $\mathbf{p} \in \mathbb{R}^3$ is occupied by any object of the scene.

Latent variables: Unobserved variables $\mathbf{z}$ capture uncertainties about the nonsmooth dynamics of contact, the sensor noise, as well as the number of objects and their geometry.

C. Probabilistic Modelling

We model the scene and the grasping task according to the Bayesian network shown in Fig. 2. This choice of structure is motivated by the dependencies observed in the system: S depends on $\mathbf{z}$, $\mathbf{P}$, and $\mathbf{h}$; o depends on $\mathbf{P}$ and $\mathbf{x}$; and $\mathbf{P}$ depends on $\mathbf{z}$. Such a structure facilitates straightforward generation procedures: $\mathbf{z}$ and $\mathbf{h}$ are sampled from their respective priors, while $\mathbf{P}$ and S are obtained through forward physical simulations.

D. Grasping as Inference

We formulate the problem of grasping as the Bayesian inference of the hand configuration $\mathbf{h}^*$ that is a posteriori the most likely given a successful grasp, an occupancy o at $\mathbf{x}$ and a point cloud $\mathbf{P}$. That is, we are seeking the maximum a posteriori (MAP) estimate

$$\mathbf{h}^* = \arg\max_{\mathbf{h}} \; p(\mathbf{h}|S = 1, o = 1, \mathbf{P}), \tag{1}$$

from which we then compute the joint trajectory

$$\tau_{1:m} = \Lambda(\tau_0, \mathrm{IK}(\mathbf{h}^*), \mathbf{P}), \tag{2}$$

where IK is an inverse kinematic solver, $\tau_{1:m}$ are waypoints in the joint space, $\tau_m = \mathrm{IK}(\mathbf{h}^*)$ with $\mathbf{h}^*$ expressed in $\underset{\rightarrow}{\mathcal{F}}_{\mathrm{W}}$ and Λ is a path planner.

III. Simulation-Based Inference

From the Bayes rule, the posterior of the hand configuration is

$$p(\mathbf{h}|S, o, \mathbf{P}) = \frac{p(S \mid \mathbf{h}, o, \mathbf{P})}{p(S \mid o, \mathbf{P})} p(\mathbf{h} \mid o, \mathbf{P}) \tag{3}$$

A. Priors

The prior $p(\mathbf{h} \mid o, \mathbf{P})$ represents our domain knowledge about the hand configuration without knowing if it leads to a successful grasp or not (S is not observed).

Position: Objects present in the workspace occupy a small volume of the whole workspace. To facilitate the exploration of potential grasp poses, we assume that, at a successful grasp pose, the position of the fingertips when the gripper is closed lies inside the object. This information is modeled by the occupancy variable o. Formally, we defined our prior as

$$p(\mathbf{x}|o = 1, \mathbf{P}) = \frac{p(o = 1|\mathbf{x}, \mathbf{P})}{p(o|\mathbf{P})} p(\mathbf{x}). \tag{4}$$

Here, $p(o|\mathbf{x}, \mathbf{P})$ represents the likelihood of occupancy o, $p(\mathbf{x})$ denotes a uniform distribution over the workspace, and $p(\mathbf{x}|\mathbf{P})$ simplifies to $p(\mathbf{x})$ due to independence.

We model the likelihood of occupancy $p(o|\mathbf{x}, \mathbf{P})$ with a convolutional occupancy network [2]. This network first generates an embedding on the three canonical planes $\mathbf{c}_{xy}(\mathbf{P}), \mathbf{c}_{xz}(\mathbf{P})$ and $\mathbf{c}_{yz}(\mathbf{P})$. Subsequently, a feature vector is obtained by summing point-wise features computed via bilinear interpolation at the desired position $\mathbf{x}_{\mathrm{D}}$, *i.e* $\psi(\mathbf{P}, \mathbf{x}_{\mathrm{D}}) = \mathbf{c}_{xy}(\mathbf{P})(\mathbf{x}_{\mathrm{D}}) + \mathbf{c}_{xz}(\mathbf{P})(\mathbf{x}_{\mathrm{D}}) + \mathbf{c}_{yz}(\mathbf{P})(\mathbf{x}_{\mathrm{D}})$. Finally, this embedding is passed through a fully connected network to produce the occupancy output.

We use a Markov chain Monte Carlo (MCMC) scheme to sample from the position prior because we have access to the target density $p(\mathbf{x}|o = 1, \mathbf{P}) \propto p(o = 1|\mathbf{x}, \mathbf{P})p(\mathbf{x})$ with the convolutional occupancy network. Furthermore, it is fully differentiable, making possible the use of the Hamiltonian Monte Carlo [3], a variant of MCMC using gradients to efficiently explore the position prior space.

Orientation: The prior of the orientation $\mathbf{q}$ is a uniform distribution over the hypersphere $\mathbb{S}^3$. This prior is *invariant* to any rotation $\mathbf{R} \in \mathbb{SO}(2)$ applied to individual objects. This is a desirable property as it does not assume any preferred orientation.

B. Neural Ratio Estimation

The likelihood function $p(S \mid \mathbf{h}, o, \mathbf{P})$ and the evidence $p(S \mid o, \mathbf{P})$ are both intractable because no closed-form relations exist to express the success of a grasp as it derives from nonsmooth contact dynamics. It makes standard Bayesian inference procedures such as Markov chain Monte Carlo unusable. However, drawing samples from forward models remains feasible with physical simulators, hence enabling simulation-based Bayesian inference algorithms [1], [4], [5].

First, we express the likelihood-to-evidence ratio as

$$r(S \mid \mathbf{h}, o, \mathbf{P}) = \frac{p(S \mid \mathbf{h}, o, \mathbf{P})}{p(S \mid o, \mathbf{P})} \tag{5}$$

$$= \frac{p(S, \mathbf{h} \mid o, \mathbf{P})}{p(S \mid o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})} \tag{6}$$

By adapting the approach described in [6] for likelihood ratio estimation, we train a neural network classifier d_ϕ which approximates $r(S|\mathbf{h}, o, \mathbf{P})$. This network is trained to distinguish tuples $(S, \mathbf{h}, o, \mathbf{P})$ (labeled $y = 1$) sampled from the joint distribution $p(S, \mathbf{h} \mid o, \mathbf{P})$ and tuples $(S, \mathbf{h}, o, \mathbf{P})$ (labeled $y = 0$) sampled from the product of the marginals $p(S \mid o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})$. The Bayes optimal classifier that minimizes the cross entropy is given by [6]

$$d^*(S, \mathbf{h}, o, \mathbf{P}) = \frac{p(S, \mathbf{h} \mid o, \mathbf{P})}{p(S, \mathbf{h} \mid o, \mathbf{P}) + p(S \mid o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})} \tag{7}$$

which recovers the likelihood-to-evidence ratio as

$$\frac{d^*}{1 - d^*} = \frac{p(S, \mathbf{h} \mid o, \mathbf{P})}{p(S \mid o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})} = \frac{p(S \mid \mathbf{h}, o, \mathbf{P})}{p(S \mid o, \mathbf{P})} \tag{8}$$

Therefore, by modeling the classifier with a neural network d_ϕ trained on the binary classification problem, we obtain an approximate but amortized and differentiable likelihood ratio

$$r_\phi(S \mid \mathbf{h}, o, \mathbf{P}) = \frac{d_\phi(S, \mathbf{h}, o, \mathbf{P})}{1 - d_\phi(S, \mathbf{h}, o, \mathbf{P})}. \tag{9}$$

Finally, the likelihood ratio is combined with the prior to approximate the posterior as

$$\hat{p}(\mathbf{h}|S, o, \mathbf{P}) = r_\phi(S \mid \mathbf{h}, o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P}), \tag{10}$$

which enables immediate posterior inference despite the initial intractability of the likelihood function $p(S \mid \mathbf{h}, o, \mathbf{P})$ and the evidence $p(S \mid o, \mathbf{P})$.

Instead of passing the raw point cloud into the ratio estimator, we use the point-wise features vector from the convolutional occupancy network $\psi(\mathbf{P}, \mathbf{x})$. Furthermore, we extract local features centered around the point $\mathbf{x}$, cropped from the features planes $\mathbf{c}_{xy}(\mathbf{P}), \mathbf{c}_{xz}(\mathbf{P})$ and $\mathbf{c}_{yz}(\mathbf{P})$ and scaled to be within the gripper's size. These local features $\Psi(\mathbf{P}, \mathbf{x})$ bring information about collisions and objects present in the area targeted by $\mathbf{x}$.

Using only a single neural network produces a posterior with very high frequencies. Ensembles tend to produce more conservative posteriors [7], making them suitable for optimization purposes. In our case, we take 5 models and compute the ratio as

$$\hat{r} = \frac{1}{5} \Sigma_{i=1}^{5} \hat{r}_i. \tag{11}$$

We train our models over 80 epochs with Adam optimizer [8] and a learning rate of 10^{-3}. The ratio network consists of three convolutional layers for extracting local features followed by two fully connected layers to finally output the binary occupancy logits. The convolutional occupancy network has the same architecture as described in [2]. The whole pipeline is illustrated in Fig. 3.

C. Sequential Neural Ratio Estimation

Despite using an informative position prior, the success rate a priori remains below 1%, resulting in an imbalanced dataset that favors failure grasps ($S = 0$). We iteratively refine our ratio estimator by using a sequential neural estimation scheme to address this issue. Starting with our prior $p_0(\mathbf{h} \mid o, \mathbf{P}) := p(\mathbf{h} \mid o, \mathbf{P})$, we refine the posterior from the previous iterate by setting it as the prior for the next iteration, $p_{t+1}(\mathbf{h} \mid o, \mathbf{P}) := \hat{p}_t(\mathbf{h}|S = 1, o, \mathbf{P})$. In each iteration, we repeat the training procedure described in [6].

Sampling from the new prior p_{t+1} poses significant challenges because $\mathbf{q}$ is defined on a manifold. The geodesic Monte Carlo scheme [9] generates samples similarly to Hamiltonian Monte Carlo for distributions defined on smooth manifolds. Furthermore, geodesic Monte Carlo is applicable to a product of manifolds $\mathcal{M}1 \times \mathcal{M}2 : (x_1, x_2) : x_1 \in \mathcal{M}1, x_2 \in \mathcal{M}_2$, which, in our specific case, corresponds to $\mathbb{R}^3 \times \mathbb{S}^3$. It requires to have access to closed-form expressions of the target density, orthogonal projection, and geodesics. While orthogonal projections and geodesics are available in closed form for $\mathbb{R}^3$ and $\mathbb{S}^3$, the target density is intractable here. As described in [6], a likelihood-free variant of Hamiltonian Monte Carlo exists, where the intractable likelihood is replaced by the ratio. Therefore, by replacing the likelihood with the ratio in the geodesic Monte Carlo scheme, we can draw samples from our posterior density $\hat{p}_t(\mathbf{h}|S = 1, o, \mathbf{P})$ defined as a product of smooth manifolds.

D. Maximum a Posteriori

Due to the intractability of the likelihood function and the evidence, Equation (1) cannot be solved analytically or numerically. We rely instead on the approximation given by the likelihood-to-evidence ratio $\hat{r}$ to find an approximation of the maximum a posteriori (MAP) estimate as

$$\hat{\mathbf{h}}^* = \arg\max_{\mathbf{h}} \hat{r}(S = 1 \mid \mathbf{h}, o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P}) \tag{12}$$

$$= \arg\min_{\mathbf{h}} -\log\big(\hat{r}(S = 1 \mid \mathbf{h}, o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})\big), \tag{13}$$

which we solve using gradient descent. The gradient of (13) decomposes as

$$\begin{aligned} &-\nabla_{(\mathbf{x},\mathbf{q})} \log\big(\hat{r}(S \mid \mathbf{h}, o, \mathbf{P}) p(\mathbf{h} \mid o, \mathbf{P})\big) \\ &= -\nabla_{(\mathbf{x},\mathbf{q})} \log \hat{r}(S \mid \mathbf{h}, o, \mathbf{P}) \\ &\quad - \nabla_{(\mathbf{x},\mathbf{q})} \log p(\mathbf{h} \mid o, \mathbf{P}). \end{aligned} \tag{14}$$

Since the likelihood-to-evidence ratio estimator $\hat{r}$ and our informative prior $p(\mathbf{h} \mid o, \mathbf{P})$ are modeled by a neural network, they are fully differentiable with respect to their inputs and their gradients can be computed by automatic differentiation. However, the orientation $\mathbf{q}$ belongs to a Riemannian manifold. Thus, performing gradient descent would violate our geometric assumptions (Fig. 4). Let us consider a variable $\mathcal{Z}$ on the

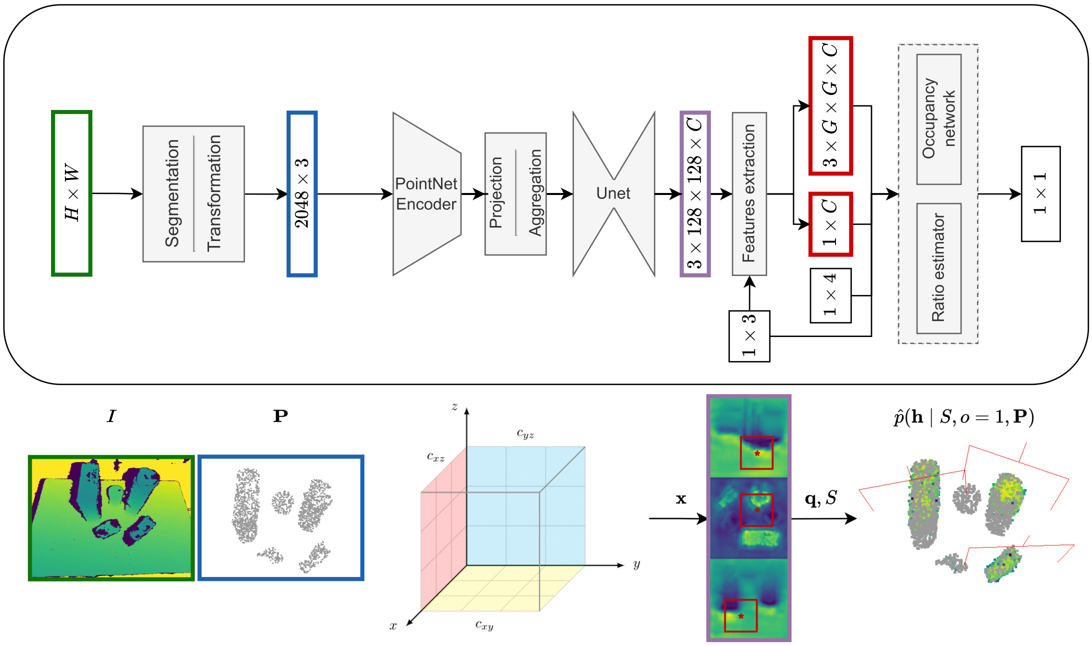

Fig. 3. Posterior inference. The depth image I (in green) passes through a segmentation model which extracts pixels belonging to objects. Then, these pixels are transformed into point cloud $\mathbf{P}$ (in blue) with intrinsic and extrinsic matrices of the depth camera. We then generate three canonical feature planes $\mathbf{c}_{xy}(\mathbf{P}), \mathbf{c}_{xz}(\mathbf{P})$ and $\mathbf{c}_{yz}(\mathbf{P})$ and extract for a given $\mathbf{h}$ point-wise $\psi(\mathbf{P}, \mathbf{x})$ and local $\Psi(\mathbf{P}, \mathbf{h})$ features. They are fed to the ratio and occupancy networks which output the posterior density $\hat{p}(\mathbf{h} \mid S, o, \mathbf{P})$.

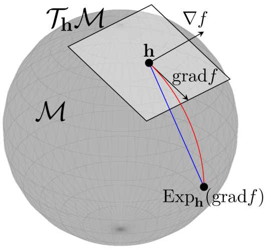
$\mathcal{T}_{\mathbf{h}}\mathcal{M}$
∇f
$\mathbf{h}$
$\text{grad} f$
$\mathcal{M}$
$\text{Exp}_{\mathbf{h}}(\text{grad} f)$

Fig. 4. Sphere manifold $\mathbb{S}^2$. $\mathbf{h}$ and $\text{Exp}_{\mathbf{h}}(\text{grad} f(\mathbf{h}))$ are points on the surface of the sphere. The red curve corresponds to the geodesic and the blue curve corresponds to the straight line in Euclidean space.

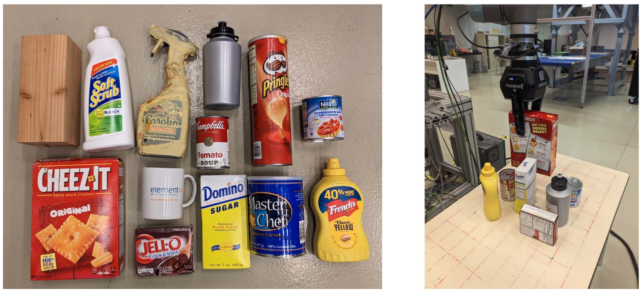

Fig. 5. (Left) Object assets used in the real setup. (Right) The first object removed is often the tallest one.

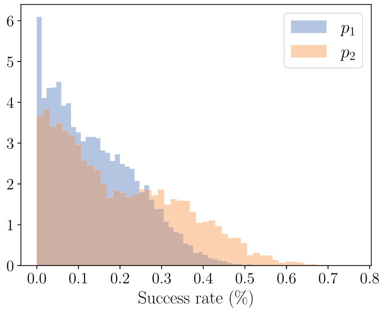

Fig. 6. Empirical distribution $p(S)$ evaluated on 10 000 scenes $\mathbf{z}$. Iterations of the sequential procedure shift the distribution from low grasping success rate regions to higher grasping success rate regions.

smooth Riemannian manifold $\mathcal{M} = \mathbb{R}^3 \times \mathbb{S}^3$ with tangent space $\mathcal{T}_{\mathcal{Z}}\mathcal{M}$ and a function $f : \mathcal{M} \to \mathbb{R}$. Since $\mathbb{S}^3$ is embedded in $\mathbb{R}^4$, f can be evaluated on $\mathbb{R}^3 \times \mathbb{R}^4$, leading to the definition of the Euclidean gradients $\nabla f(\mathcal{Z}) \in \mathbb{R}^3 \times \mathbb{R}^4$. In turn, these Euclidean gradients can be transformed into their Riemannian counterparts $\text{grad} f(\mathcal{Z})$ via orthogonal projection $\mathbf{P}_{\mathcal{Z}}$ into the tangent space $\mathcal{T}_{\mathcal{Z}}\mathcal{M}$. Therefore,

$$\text{grad} f(\mathcal{Z}) = \mathbf{P}_{\mathcal{Z}}(\nabla f(\mathcal{Z})) \tag{15}$$

where the orthogonal projection onto $\mathbb{R}^3$ is the identity $\mathbb{I}_3$ and the orthogonal projection onto $\mathbb{S}^3$ is $\mathbf{P}_{\xi}(\nabla f) = (\mathbb{I}_4 - \xi\xi^T)\nabla f$ at $\xi \in \mathbb{S}^3$. Thus, we can solve (13) by projecting Euclidean gradients of (14) to the tangent space $\mathcal{T}_{\mathcal{Z}}\mathcal{M}$ and use it in the following update rule [10]

$$\mathbf{h}_{k+1} = \text{Exp}_{\mathbf{h}_k}(-\alpha_k \text{grad} f(\mathbf{h}_k)) \tag{16}$$

with $\text{Exp}_x(v) : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$ is the exponential map and α_k is the step size.

IV. Experiments: Design and Setup

We assess our approach on a robotic grasping task in both simulation and real-world settings. Specifically, we investigate three questions: 1) Does the method transfer well from simulation to real-world? 2) How the sequential improvement impacts the grasping success rate? 3) What kind of posterior distribution are learned?

A. Data Generation

The data generating procedure is defined as follows:

$$\mathbf{z} \sim p(\mathbf{z}) \tag{17}$$

$$I \sim p(I \mid \mathbf{z}, \mathbf{T}_{\mathrm{WC}}) \tag{18}$$

$$\mathbf{P} = f(I, \mathbf{T}_{\mathrm{WC}}, K) \tag{19}$$

$$\{\mathbf{h} \sim p(\mathbf{h} \mid o = 1, \mathbf{P})\} \tag{20}$$

$$\{\tau_{1:m} \sim \Lambda(\tau_0, \mathrm{IK}(\mathbf{h}), \mathbf{P})\} \tag{21}$$

$$\{S \sim p(S \mid \tau_{1:m}, \mathbf{z})\} \tag{22}$$

We use Pybullet [11] for implementing these functions. We use the same object assets as VGN [12] for the training and testing and we placed the objects in a *packed* scenario, as defined in [12]. The latent variables $\mathbf{z}$ are described as follow:

Number of objects: We sample according to a Poisson law $N \sim Pois(4) + 1$ [12]

Object mesh: We sample uniformly an object mesh from an asset of objects.

Pose of the table: $\mathbf{T}_{\mathrm{ST}}$ We randomize the position $(x, y) \sim \mathcal{N}(0, 0.008)$ and the rotation $q_{\mathrm{T}} = (0., 0., \sin(\frac{\theta_{\mathrm{Table}}}{2}), \cos(\frac{\theta_{\mathrm{Table}}}{2}))$, $\theta_{\mathrm{Table}} \sim \mathcal{U}(-5, 5)$ of the table with respect to $\underrightarrow{\mathcal{F}}_{\mathrm{S}}$.

Pose of the object: $\mathbf{T}_{\mathrm{TO}}$ We randomize the position $(x, y) \sim \mathcal{U}(\frac{-l}{2}, \frac{l}{2})$ and the orientation $q_{\mathrm{O}} = (0., 0., \sin(\frac{\theta_{\mathrm{O}}}{2}), \cos(\frac{\theta_{\mathrm{O}}}{2}))$, $\theta_{\mathrm{O}} \sim \mathcal{U}(0, 2\pi)$ of the object with respect to $\underrightarrow{\mathcal{F}}_{\mathrm{T}}$.

Torque applied by the fingers: We randomize the final torque applied by the fingers $\tau \sim \mathcal{U}(35, 40)$.

Lateral friction coefficient: We randomize the lateral friction coefficient $\mu \sim \mathcal{U}(1, 2)$.

Spinning friction coefficient: We randomize the spinning friction coefficient $\gamma = \eta\mu, \eta \sim \mathcal{N}(0.002, 0.0001)$.

Depth images: We add noise to the rendered depth images in simulation using the additive noise model of [13] with the same parameters.

B. Robotic Setup

We carry out experiments with a Robotiq 3-finger gripper attached to a UR5 robotic arm, as shown in Fig. 1. An Intel Realsense D435i depth sensor is mounted to the flange of the arm and produces 848 × 480 depth images. The transformation $\mathbf{T}_{\mathrm{FC}}$ is calibrated using hand-eye calibration from OpenCV [14]. The objects used for testing are unseen during training, except for the bleach cleanser and the mustard, which are kept because of their mass and deformability, violating the rigid body assumption of Pybullet (Fig. 5).

TABLE I
Grasp Success Rate and Declutter Rate for Picking Experiments for the Packed Scenario With 5 Objects Over 100 Rounds

Method	GSR ↑	DR ↑
Simulation results		
GPD [15]	35.4 ± 1.9	30.7 ± 2.0
VGN [12]	74.5 ± 1.3	79.2 ± 2.3
GIGA [13]	87.9 ± 3	$\mathbf{86} \pm 3.2$
Ours (4DoF)	**91.1**	77
Ours (6DoF)	79.71 ± 1.7	58.56 ± 3.2
Real-world results		
VGN [12]	77.2	81.3
GIGA [13]	83.3	86.6
Ours (4DoF)	**95.6**	**88**
Ours (6DoF)	86.3	62.1

Real-world results are performed with 5 objects over 15 rounds. Results of GPD [15], VGN [12] and GIGA [13] are reported from [13].

V. Experiments: Results

A. Grasping Results

We compared our results with [13], as they also used an implicit representation and only one depth image. However, we benchmark only on *packed* scenario because our gripper is too big to fit small objects of the *pile* scenario, and our robotic arm has only 6 DoF and not 7, making it harder to find a valid joint trajectory.

For each iteration of the sequential approach, we generate 7500 different scenes $\mathbf{z}$ for each we sample 4000 tuples $(\mathbf{h}, S)$ from the prior p_t and train 5 models. We stop the procedure at the second iteration and use $p_2(\mathbf{h} \mid S = 1, o = 1, \mathbf{P})$ to compute the grasp poses. Because our modeling treats the rotation as a random variable belonging to the n-sphere manifold, we also train a neural ratio estimator to predict the 4 DoF pose (we enforce a top-down approach) with $\mathbf{q} \in \mathbb{S}^1$ but we only use the $\mathbf{c}_{xy}(\mathbf{P})$ features plane. We train 5 ratios on a unique dataset of 7500 different scenes $\mathbf{z}$ and 4000 tuples $(\mathbf{h}, S)$ per each scene.

We evaluate the grasp success rate (GSR), the ratio of successful grasp executions, and the declutter rate (DR), the average ratio of objects removed. Results are reported in Table I and are performed with 5 objects over 100 rounds for the simulation and 15 rounds for the real setup. To compute the MAP, we first take the 20 best candidates, that maximize the posterior density, among 1800 hand configurations sampled from the prior. Then, we perform 100 optimization steps and keep the best candidate, which takes approximately 20 seconds to compute. If our method fails three times in a row to generate a reachable hand configuration, we note it as a failure. Globally, our method performs similar to or better than other approaches. Our 4 DoF model performs better than the 6 DoF, mainly due to the lower dimensional space. While the DR of GIGA and VGN are similar to their GSR, our DR is significantly lower than our GSR, meaning that our model mostly fails to pick the first object. This is due to the probability of collision, which is high at the start and then decreases gradually when objects are removed from the table. The discrepancy between the simulation and real-world setup is overcome without any decrease in performance. In real-world settings, the majority of failure cases are due to insufficient friction forces, causing the objects to slip. We believe that these

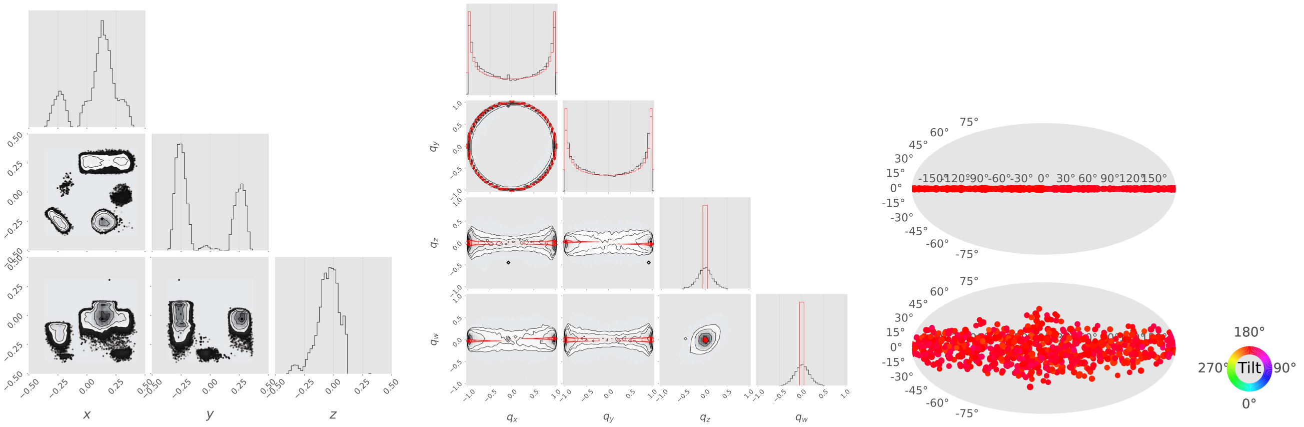

Fig. 7. Posterior distribution $\hat{p}(\mathbf{h} \mid S = 1, o = 1, \mathbf{P})$ of the grasp pose estimated by geodesic Hamiltonian Monte Carlo. The scene and thus the point cloud are the same as in Fig. 3. (left) The marginal distribution of the position $\hat{p}(\mathbf{x} \mid S = 1, o = 1, \mathbf{P})$. (middle) The marginal distribution of the orientation $\hat{p}(\mathbf{q} \mid S = 1, o = 1, \mathbf{P})$ in black compared to a top-down approach in red by components of the quaternion. (right top) Distribution $p(\mathbf{q})$ along a top-down approach shown with a Mollweide projection as in [16]. (right bottom) The marginal distribution of the orientation $\hat{p}(\mathbf{q} \mid S = 1, o = 1, \mathbf{P})$.

TABLE II
GSR of Sampling Strategies

Sampling strategy	GSR
Uniform prior	0.05
$p_0(\mathbf{h} \mid o, \mathbf{P})$	0.96
$p_1(\mathbf{h} \mid o, \mathbf{P})$	14
$p_2(\mathbf{h} \mid o, \mathbf{P})$	19.62

failure scenarios happen because the simulator used to generate the training set does not accurately model friction forces.

B. Sample Efficiency

We generate 2000 different scenes $(\mathbf{z})$ and for each scene, we sample 2000 tuples $(\mathbf{h}, S)$ from the prior and the sequential posteriors. We report the results in Table II. Unsurprisingly, sampling for the sequential posteriors leads to an increasing GSR. Using an occupancy network as a prior conditioned by observation for the position increases the GSR by a factor of 20. However, this rate is still far from an acceptable one, mainly due to the orientation prior, which is uniform. As shown in Fig. 6, the distribution $p(S)$ shifts from a concentrated distribution near 0% (meaning that for one scene $\mathbf{z}$, the sampled hand configurations $\mathbf{h}$ will lead to a very low grasping success rate in expectation) and spreads toward higher grasping success rate areas, meaning that each iteration of the sequential procedure generalizes better than the previous one.

C. Posteriors

Our method has the advantage of offering access to the full posterior distribution over $\mathcal{H}$. Thus, with our geodesic Hamiltonian Monte Carlo scheme, we can sample hand configurations from the posterior $\mathbf{h} \sim p(\mathbf{h} \mid S = 1, o, \mathbf{P})$, as depicted in Fig. 7. While our conditional prior distributes density uniformly across the objects, the posterior assigns maximal density to the top of objects and minimal density to the bottom of objects when multiple objects are present on the table. This occurs due to potential collisions between the gripper and the table, or the gripper and other objects. In practice, our method will first pick the tallest object present on the table and then move to the next tallest object or an isolated object (Fig. 5). Regarding orientation, the posterior converges to a top-down grasping approach while having some deviations from a strict 4 DoF setup. Top-down approaches seem optimal in our setup (small workspace size and large gripper) because they avoid many collisions between the gripper and the table or the other objects. However, side grasps may emerge from a different setup (unique object, parallel-jaw gripper, etc.).

VI. Related Work

Grasp sampling strategies that generate data for machine learning methods can be categorized based on their coverage of the hand configuration space $\mathcal{H}$ [17]. Uniform sampling provides direct density estimation but is highly inefficient. Heuristic methods [18], [19] are efficient but are not suitable for complex settings such as multi-fingered grippers. Our prior, based on occupancy networks, offers direct density estimation, efficient sampling and does not depend on specific objects or grippers.

Our work capitalizes on the latest advancements in neural implicit representations, which parameterize a 3D scene as a continuous function [20], [21], [22]. Implicit representations possess the advantage of offering 'infinite resolution' compared to discrete approaches. Furthermore, their rich latent space endows them with powerful feature extraction capabilities for diverse purposes. Additionally, they exhibit flexibility by being conditioned on different sensor inputs, such as point clouds or voxel grids, thus making them highly adaptable for robotic applications [13], [23], [24].

Probabilistic approaches for grasping problems typically rely on likelihood functions that model the probability of grasp success or a grasp quality metric given an observation and grasp pose. Various methods can be employed to determine the maximum likelihood estimate (MLE) corresponding to the final grasp pose, such as numerical optimization [25], direct regression [26], sampling and refinement [27] or a list of candidates [12]. Similar to us, [13] use an implicit representation for estimating the grasp quality with a surrogate of the likelihood but compute only point estimates for the orientation, losing information. We instead have access to the full posterior density for both the position and orientation. [28] approximates the posterior by learning the likelihood and the prior from data and computes the MAP by gradient descent. However, unlike our approach, they employ Euler angles, which can be susceptible to gimbal lock and singularities. Our method preserves the geometry of the rotation space by performing Riemannian gradient descent.

VII. Conclusion

We have shown that simulation-based Bayesian inference can be applied effectively to robotic grasping in complex and noisy environments. The proposed innovative method improves the sample efficiency of the inference pipeline. Our approach can manage tasks with escalating complexity and proves valuable for real-world robotic applications. Future research will focus on the real-time constraints for inferring the optimal grasp pose.

References

[1] K. Cranmer, J. Brehmer, and G. Louppe, “The frontier of simulation-based inference,” *Proc. Nat. Acad. Sci.*, vol. 117, no. 48, pp. 30055–30062, 2020.

[2] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, “Convolutional occupancy networks,” in *Proc. 16th Eur. Conf. Comput. Vis.*, 2020, pp. 523–540.

[3] R. M. Neal et al., “MCMC using Hamiltonian dynamics,” 2012, *arXiv:1206.1901*.

[4] G. Papamakarios, D. Sterratt, and I. Murray, “Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows,” in *Proc. 22nd Int. Conf. Artif. Intell. Statist.*, 2019, pp. 837–848.

[5] G. Papamakarios and I. Murray, “Fast ϵ-free inference of simulation models with Bayesian conditional density estimation,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, vol. 29, pp. 1036–1044.

[6] J. Hermans, V. Begy, and G. Louppe, “Likelihood-free MCMC with amortized approximate ratio estimators,” in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 4239–4248. [Online]. Available: http://proceedings.mlr.press/v119/hermans20a.html

[7] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, V. Begy, and G. Louppe, “A crisis in simulation-based inference? Beware, your posterior approximations can be unfaithful,” *Trans. Mach. Learn. Res.*, 2022.

[8] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, *arXiv:1412.6980*.

[9] S. Byrne and M. Girolami, “Geodesic Monte Carlo on embedded manifolds,” *Scand. J. Statist.*, vol. 40, no. 4, pp. 825–845, 2013.

[10] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[11] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” 2016–2020. [Online]. Available: http://pybullet.org,

[12] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, “Volumetric grasping network: Real-time 6 DoF grasp detection in clutter,” in *Proc. Conf. Robot Learn.*, 2020, pp. 1602–1611.

[13] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, “Synergies between affordance and geometry: 6-DOF grasp detection via implicit representations,” in *Robot.: Sci. Syst. XVII*, Virtual Event, D. A. Shell, M. Toussaint, and M. A. Hsieh, Eds., Jul. 2021. [Online]. Available: https://doi.org/10.15607/RSS.2021.XVII.024

[14] G. Bradski, “The OpenCV library,” *Dr Dobb’s J. Softw. Tools Professional Programmer*, vol. 25, pp. 120–123, 2000.

[15] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, “Grasp pose detection in point clouds,” *Int. J. Robot. Res.*, vol. 36, no. 13/14, pp. 1455–1473, 2017.

[16] K. A. Murphy, C. Esteves, V. Jampani, S. Ramalingam, and A. Makadia, “Implicit-pdf: Non-parametric representation of probability distributions on the rotation manifold,” in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7882–7893.

[17] C. Eppner, A. Mousavian, and D. Fox, “A billion ways to grasp: An evaluation of grasp sampling schemes on a dense, physics-based grasp data set,” in *Proc. 19th Int. Symp. Robot. Res.*, 2022, pp. 890–905.

[18] J. Mahler et al., “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” in *Robot.: Sci. Syst. XIII, Massachusetts Inst. Technol.*, N. M. Amato, S. S. Srinivasa, N. Ayanian, and S. Kuindersma, Eds., Cambridge, Massachusetts, USA, Jul. 2017. [Online]. Available: http://www.roboticsproceedings.org/rss13/p58.htm

[19] X. Yan et al., “Learning 6-DOF grasping interaction via deep geometry-aware 3D representations,” in *Proc. IEEE Int. Conf. Robot. Automat.* 2018, pp. 3766–3773.

[20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “Deepsdf: Learning continuous signed distance functions for shape representation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 165–174.

[21] 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,” *Commun. ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[22] A. Simeonov et al., “Neural descriptor fields: Se (3)-equivariant object representations for manipulation,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 6394–6400.

[23] K. Karunratanakul, J. Yang, Y. Zhang, M. J. Black, K. Muandet, and S. Tang, “Grasping field: Learning implicit representations for human grasps,” in *Proc. Int. Conf. 3D Vis.*, 2020, pp. 333–344.

[24] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake, “Learning models as functionals of signed-distance fields for manipulation planning,” in *Proc. Conf. Robot Learn.*, 2022, pp. 245–255.

[25] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, “Planning multi-fingered grasps as probabilistic inference in a learned deep network,” in *Proc. Robot. Res.*, 2020, pp. 455–472.

[26] J. Cai, J. Cen, H. Wang, and M. Y. Wang, “Real-time collision-free grasp pose detection with geometry-aware refinement using high-resolution volume,” *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1888–1895, Apr. 2022.

[27] A. Mousavian, C. Eppner, and D. Fox, “6-DOF GraspNet: Variational grasp generation for object manipulation,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 2901–2910.

[28] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, “Learning continuous 3D reconstructions for geometrically aware grasping,” in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 11516–11522.