Simulation-based Bayesian inference for robotic grasping

Norman Marlier[1] Olivier Brüls[2] Gilles Louppe[3]

***Abstract*— General robotic grippers are challenging to control because of their rich nonsmooth contact dynamics and the many sources of uncertainties due to the environment or sensor noise. In this work, we demonstrate how to compute 6-DoF grasp poses using simulation-based Bayesian inference through the full stochastic forward simulation of the robot in its environment while robustly accounting for many of the uncertainties in the system. A Riemannian manifold optimization procedure preserving the nonlinearity of the rotation space is used to compute the maximum a posteriori grasp pose. Simulation and physical benchmarks show the promising high success rate of the approach.**

I. Introduction

Industrial grasping works very well in highly structured environments with few uncertainties. However, complex applications requiring great flexibility have recently gained a lot of interest. For such tasks, dealing with uncertainties becomes key to robust performance.

While previous methods relied on simplified surrogates of the likelihood function, we bring a novel simulation-based approach for full Bayesian inference based on a deep neural network surrogate of the likelihood-to-evidence ratio. By framing robotic grasping as an inference task, we demonstrate the general applicability of simulation-based inference algorithms to complex robotic tasks and their usefulness to deal with uncertainties.

We summarize our contributions as follow:

- We bring simulation-based Bayesian inference methods [1] to robotic grasping.
- We make use of Riemannian manifold optimization to deal with the nonlinearity of the rotation space.
- We validate our method on simulated and real experiments. Results show promising grasping performances.

II. Problem statement

We consider the problem of planning 6-DoF hand configurations of a general robotic gripper for unknown rigid objects placed on a table and observed through multi-view depth images (Fig. 1).

A. Description

The robot arm (6 or 7 DoF) evolves in a cubic workspace with a planar tabletop. It is equipped with a robotic gripper and observes the scene with a depth camera mounted on its flange. Depth images, captured along a predefined trajectory, are fused into a Truncated Signed Distance Function (TSDF) voxel grid [2]. Then, we search for the most plausible hand configuration given a successful grasp and the TSDF voxel grid. Finally, a joint trajectory is computed by a path planner based on the TSDF to reach the hand pose and grasp the object in order to remove it from the table.

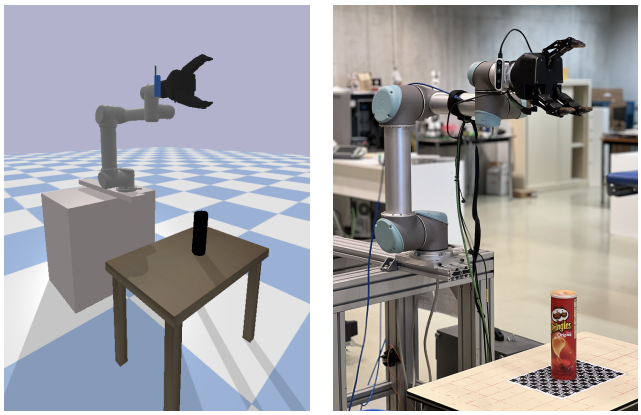

Fig. 1: Our benchmark scene. (left) The simulated environment. (right) The real setup.

B. Notations

Frames We use several reference frames in our work. The world frame $\underrightarrow{\mathcal{F}}_{\mathrm{W}}$ and the workspace frame $\underrightarrow{\mathcal{F}}_{\mathrm{S}}$ can be choosen freely and are not tied to a physical location. $\underrightarrow{\mathcal{F}}_{\mathrm{B}}$, $\underrightarrow{\mathcal{F}}_{\mathrm{C}}$, $\underrightarrow{\mathcal{F}}_{\mathrm{F}}$, $\underrightarrow{\mathcal{F}}_{\mathrm{E}}$ correspond respectively to the robot base, the camera, the flange and the tool center point (TCP).

Hand configuration The hand configuration $\mathbf{h} \in \mathcal{H} = \mathbb{R}^3 \times \mathrm{SO}(3)$ is defined as the combination of the pose $\mathbf{T}_{\mathrm{SE}} = ({}_{\mathrm{S}}\mathbf{t}_{\mathrm{SE}}, \mathbf{R}_{\mathrm{SE}}) \in \mathbb{R}^3 \times \mathrm{SO}(3)$ of the hand, where ${}_{\mathrm{S}}\mathbf{t}_{\mathrm{SE}}$ is the vector $\vec{SE}$ expressed in $\underrightarrow{\mathcal{F}}_{\mathrm{S}}$. We parametrize the rotation $\mathbf{R}_{\mathrm{SE}}$ with quaternions.

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

Observation Given the depth images $\mathcal{I}_k = \{I_0, ..., I_k\}$ with their corresponding transformations camera to world $\Gamma_k = \{\mathbf{T}^0_{\mathrm{WC}}, ..., \mathbf{T}^k_{\mathrm{WC}}\}$ and camera intrinsic matrix K, we construct a TSDF voxel grid $\mathbf{V}$ with N^3 voxels, representing the workspace of size l.

Latent variables Unobserved variables $\mathbf{z}$ capture uncertainties about the nonsmooth dynamics of contact, the sensor noise, as well as the geometry of the object (see Section.V-A).

C. Probabilistic modeling

We model the scene and the grasping task according to the Bayesian network shown in Fig. 2. The variables $S, \mathbf{V}$ and

*This work was not supported by any organization

[1]norman.marlier@uliege.be
[2]o.bruls@uliege.be
[3]g.louppe@uliege.be

$\mathbf{h}$ are modelled as random variables to capture the noise in sensors, uncertainties in the dynamics, as well as our prior beliefs about the hand configuration. The structure of the Bayesian network is motivated by the fact that S is dependent on $\mathbf{h}$, $\mathbf{V}$ and $\mathbf{z}$, $\mathbf{h}$ is dependent of $\mathbf{V}$ and $\mathbf{V}$ is dependent on $\mathbf{z}$. This structure also enables a direct and straightforward way to generate data: $\mathbf{h}$ and $\mathbf{z}$ are sampled from their respective prior distributions while S and $\mathbf{V}$ can be generated using forward physical simulators.

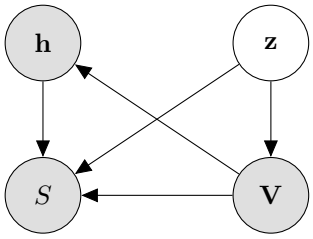

Fig. 2: Probabilistic graphical model of the environment. Gray nodes correspond to observed variables and white nodes to unobserved variables.

D. Objectives

Given our probabilistic graphical model, 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 and a TSDF voxel grid $\mathbf{V}$. That is, we are seeking for the maximum a posteriori (MAP) estimate

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

from which we then compute the joint trajectory

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

where IK is an inverse kinematic solver, $\tau_{1:m}$ are waypoints in the joint space, $\tau_m = \text{IK}(\mathbf{h}^*)$ and Λ is a path planner.

III. Related work

Probabilistic approaches for grasping problems are usually based on likelihood functions which model the probability of success or a grasp quality metric with respect to an observation and a grasp pose. Then, different methods can be used to find the maximum likelihood estimate (MLE) which corresponds to the final grasp pose. Numerical optimization can be used when the likelihood is modeled by differentiable models [3]. Direct regression of the MLE with a learnt model generates quick output but without capturing the full distribution [4]. Other approaches identify the maximum likelihood estimate based on a list of candidates computed through a grasp map on the sensor space [5], [6]. Similar to our work, [7] learn models respectively for the likelihood and the prior. Then, they can optimize via gradient descent the posterior density. Contrary to our work, they use Euler angles which can lead to gimbal lock and singularities. Our method preserves the topology by using Riemannian gradient descent.

From a statistical perspective, several Bayesian likelihood-free inference algorithms [8], [9], [10], [11], [12], [13], [14] have been developed to carry out inference when the likelihood function is implicit and intractable. These methods operate by approximating the posterior through rejection sampling or by learning parts of the Bayes' rule, such as the likelihood function, the likelihood-to-evidence ratio, or the posterior itself. These algorithms have been used across a wide range of scientific disciplines such as particle physics, neuroscience, biology, or cosmology [1]. To the best of our knowledge, our work is one of the first to apply one of those for the direct planning successful grasps. More specifically, we rely here on amortized neural ratio estimation [14] to carry out inference within seconds for any new observation $\mathbf{V}$. In contrast, an approach such as ABC [8], [9] could take up to hours to determine a single hand configuration $\mathbf{h}$ since data would need to be simulated on-the-fly for each observation $\mathbf{V}$ due to the lack of amortization of ABC. Neural posterior estimation [13] is also amortizable but would have required new methodological developments to be applicable on distributions defined on manifolds, such as those needed here for the rotational part of the pose.

IV. Likelihood-free Bayesian inference for multi-fingered grasping

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

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

A. Priors

Position The prior over the position ${}_{\text{S}}\mathbf{t}_{\text{SE}} \coloneqq \mathbf{x}_{\text{E}}$ is a uniform distribution over all the dimensions. We first use a uniform distribution over the cube of length $[-1,1]^3$, called $p(\mathbf{u})$ and then use the bijection $\text{B}(\mathbf{u};\mathbf{V}) : [-1,1]^3 \rightarrow [x_{\text{low}}, x_{\text{high}}] \times [y_{\text{low}}, y_{\text{high}}] \times [z_{\text{low}}, z_{\text{high}}]$ to compute $\mathbf{x}_{\text{E}}$, where the bounds are chosen to be the dimensions of the object voxel axis aligned bounding box. Then, $p(\mathbf{x}_{\text{E}} \mid \mathbf{V}) = (\text{B}(\mathbf{V}) \circ p)(\mathbf{u})$. It ensures that the position and orientation are within the same numerical values for estimating the density and the bijection emphasizes our ignorance about interesting regions of space for grasping.

Orientation The prior over the orientation $\mathbf{R}_{\text{SE}} \coloneqq \mathbf{q}_{\text{E}}$ is defined as a mixture of *power-spherical* (PS) distributions [15] with 20 modes ν_i (Fig. 3). Each mode is itself a mixture that satisfies $p(\mathbf{q}_{\text{E}};\cdot) = p(-\mathbf{q}_{\text{E}};\cdot)$. In total, we have

$$p(\mathbf{q}_{\text{E}}) = \frac{1}{20}\sum_{i=1}^{20} \frac{\text{PS}(\mathbf{q}_{\text{E}};\nu_i,\kappa)}{2} + \frac{\text{PS}(\mathbf{q}_{\text{E}};-\nu_i,\kappa)}{2}. \tag{4}$$

This prior encodes a top-down approach as well as side approaches by its 5 main modes ν_i. The 4 additional modes, rotated by $\frac{\pi}{2}$, allows us to explore various orientations. We set the concentration factor $\kappa = 8$ for all modes, which keeps the prior gradients low and not hightly regularizes the MAP. In this way, our prior covers a large part of the rotation space

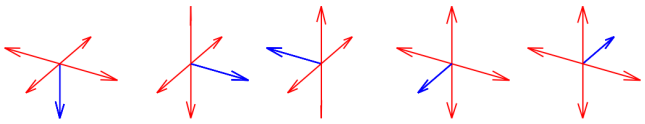

Fig. 3: The modes of the orientation distribution. (left) Encode a top-down approach. (others) Encode side approach.

and is sufficiently informative by contrast to a uniform prior over the unit sphere $\mathbb{S}^3$.

Finally, $p(\mathbf{h} \mid \mathbf{V}) = p(\mathbf{x}_{\mathrm{E}} \mid \mathbf{V})p(\mathbf{q}_{\mathrm{E}})$.

B. Density ratio estimation

The likelihood function $p(S \mid \mathbf{h}, \mathbf{V})$ and the evidence $p(S \mid \mathbf{V})$ are both intractable, which 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 likelihood-free Bayesian inference algorithms.

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

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

By adapting the approach described in [14] for likelihood ratio estimation, we train a neural network classifier d_ϕ that we will use to approximate $r(S|\mathbf{h}, \mathbf{V})$. The network d_ϕ is trained to distinguish positive tuples $(S, \mathbf{h}, \mathbf{V})$ (labeled $y = 1$) sampled from the joint distribution $p(S, \mathbf{h} \mid \mathbf{V})$ against negative tuples (labeled $y = 0$) sampled from the product of marginals $p(S \mid \mathbf{V})p(\mathbf{h} \mid \mathbf{V})$. The Bayes optimal classifier $d^*(S, \mathbf{h}, \mathbf{V})$ that minimizes the cross-entropy loss is given by

$$d^*(S, \mathbf{h}, \mathbf{V}) = \frac{p(S, \mathbf{h} \mid \mathbf{V})}{p(S \mid \mathbf{V})p(\mathbf{h} \mid \mathbf{V}) + p(S, \mathbf{h} \mid \mathbf{V})}, \tag{6}$$

which recovers the likelihood ratio $r(S|\mathbf{h})$ as

$$\frac{d^*(S, \mathbf{h}, \mathbf{V})}{1 - d^*(S, \mathbf{h}, \mathbf{V})} = \frac{p(S, \mathbf{h} \mid \mathbf{V})}{p(S \mid \mathbf{V})p(\mathbf{h} \mid \mathbf{V})} = \frac{p(S|\mathbf{h}, \mathbf{V})}{p(S \mid \mathbf{V})}. \tag{7}$$

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

$$\hat{r}(S \mid \mathbf{h}, \mathbf{V}) = \frac{d_\phi(S, \mathbf{h}, \mathbf{V})}{1 - d_\phi(S, \mathbf{h}, \mathbf{V})}. \tag{8}$$

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

$$\hat{p}(\mathbf{h}|S = 1, \mathbf{V}) = \hat{r}(S = 1 \mid \mathbf{h}, \mathbf{V})p(\mathbf{h} \mid \mathbf{V}), \tag{9}$$

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

Ensembles tend to produce more conservative posteriors [16]. In our case, we take 4 models and compute the ratio as

$$\log \hat{r} = \log \frac{1}{4} \Sigma_{i=1}^{4} \exp \log \hat{r}_i \tag{10}$$

The neural network classifiers d_ϕ is architectured as follows. The hand configuration $\mathbf{h}$ enters the neural network as a tuple of $(\mathrm{N_B} \times 3, \mathrm{N_B} \times 4)$ vector where $\mathrm{N_B}$ is the batch size. The position is rescaled into a cube of $[-1, 1]$ thanks to a bijection. In d_ϕ, $\mathbf{V}$ is fed to a 3D convolutional network made of four convolutional layers followed by a fully connected layer, as in [5], and which goal is to produce a vector embedding of the voxel grid. The voxel embedding, the 4D pose (position and 2D rotation) of the object point cloud $\mathbf{p} = f(\mathbf{V})$ obtained via the TSDF, S and $\mathbf{h}$ are then fed to a subsequent network made of 2 fully connected layers of 256 neurons. The parameters ϕ are optimized using Adam as optimizer.

C. Maximum a posteriori estimation

Due to the intractability of the likelihood function and of the evidence, Eq. (1) cannot be solved analytically nor 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

$$\hat{\mathbf{h}}^* = \arg\max_{\mathbf{h}} \hat{r}(S = 1 \mid \mathbf{h}, \mathbf{V})p(\mathbf{h} \mid \mathbf{V}) \tag{11}$$

$$= \arg\min_{\mathbf{h}} -\log \hat{r}(S = 1 \mid \mathbf{h}, \mathbf{V})p(\mathbf{h} \mid \mathbf{V}), \tag{12}$$

which we solve using gradient descent. The gradient of Eq. (12) decomposes as

$$-\nabla_{(\mathbf{x},\mathbf{q})} \log \hat{r}(S \mid \mathbf{h}, \mathbf{V})p(\mathbf{h} \mid \mathbf{V}) = -\nabla_{(\mathbf{x},\mathbf{q})} \log \hat{r}(S \mid \mathbf{h}, \mathbf{V}) - \nabla_{(\mathbf{x},\mathbf{q})} \log p(\mathbf{h} \mid \mathbf{V}). \tag{13}$$

Our prior $p(\mathbf{h} \mid \mathbf{V})$ has analytical gradients. In fact, uniform distributions are set to have null gradient everywhere in the domain. Therefore, $\nabla_{\mathbf{x}} p(\mathbf{h}) = \mathbf{0}$. By contrast, $p(\mathbf{q}_{\mathrm{E}})$ is a weakly informative prior and has a non null gradient from the power spherical distribution. Its derivative with respect to $\mathbf{q}$ is

$$\begin{aligned} \nabla_{\mathbf{q}} p(\mathbf{q}; \nu, \kappa) &= C(\kappa)\kappa(1 + \nu^T \mathbf{q})^{\kappa-1} \nabla_{\mathbf{q}}(1 + \nu^T \mathbf{q}) \\ &= C(\kappa)\kappa\nu(1 + \nu^T \mathbf{q})^{\kappa-1}, \end{aligned} \tag{14}$$

where $C(\kappa)$ is the normalization term. Since the likelihood-to-evidence ratio estimator $\hat{r}$ is modelled by a neural network, it is fully differentiable with respect to its inputs and its gradients can be computed by automatic differentiation. However, not all variables of the problem are Euclidean variables and naively performing gradient descent would violate our geometric assumptions. Let us consider a variable $\mathcal{Z}$ on the 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 $\mathrm{grad} f(\mathcal{Z})$ via orthogonal projection $\mathbf{P}_{\mathcal{Z}}$ into the tangent space $\mathcal{T}_{\mathcal{Z}}\mathcal{M}$. Therefore,

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

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 Eq. (12) by projecting Euclidean gradients of Eq. (13) to the tangent space $\mathcal{T}_{\mathcal{Z}}\mathcal{M}$ and use it in the following update rule

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

with $\exp_x(v) : \mathcal{T}_x\mathcal{M} \to \mathcal{M}$ is the exponential map.

V. Experiments

To validate our approach, we perform a series of experiments in simulation as well as in the real setup. We evaluate the performance of our method and determine the transfer capabilities of our network without any fine-tuning.

A. Data generation

The data generating procedure is defined as follow:

$$\mathbf{z} \sim p(\mathbf{z}) \tag{17}$$
$$I_k \sim p(I \mid \mathbf{z}, \mathbf{T}^k_{\text{WC}}) \tag{18}$$
$$\mathbf{V} = f(\mathcal{I}_k, \Gamma_k) \tag{19}$$
$$\{\mathbf{h} \sim p(\mathbf{h} \mid \mathbf{V})\} \tag{20}$$
$$\{\tau_{1:m} \sim \Lambda(\tau_0, \text{IK}(\mathbf{h}), \mathbf{V})\} \tag{21}$$
$$\{S \sim p(S \mid \tau_{1:m}, \mathbf{z})\} \tag{22}$$

We use Pybullet [17] for implementing these functions. We use the same object assets than VGN [5] for the training and testing. The latent variables $\mathbf{z}$ are described as follow:

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

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

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

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 [18] with the same parameters.

B. Simulated Experiments

We evaluate the performance of our method with the success rate (%). For one round, procedures from (17) to (19) are done. We find the MAP or the MLE by sampling 1000 initial hand configurations from the prior and we take the best one. Then, we perform 300 optimization steps with a step size of 0.005 for the orientation and 0.008 for the position. Because of the stochastic nature of our MAP estimate, we recompute the MLE/MAP at a maximum of 3 times if the path planner fails to find a valid path. Our method reaches a success rate of nearly $\mathbf{91}\%$ with the MAP, demonstrating the capabilities to adapt to new objects and correctly lift object. Moreover, the MLE performs slightly lower (87.3%) than the MAP. Our weakly informative prior explains the difference in success rates and motivates the use of a Bayesian approach.

Fig. 4: (left) Object assets used in the real setup. (right) Example of side grasp.

C. Real Robot Experiments

We carry out experiments with a Robotiq 3-finger gripper attached to a UR5 robotic arm, as shown in Fig. 1. A Intel Realsense D435i depth sensor is mounted to the flange of the robotic arm. It produces 848×480 depth images which are integrated into a TSDF with a resolution of $N = 40$ for the network and a resolution of $N = 120$ for collision detection using Open3D [19]. The transformation $\mathbf{T}_{\text{FC}}$ is calibrated using hand-eye calibration from OpenCV [20]. All the devices are handled within the ROS framework. We performs 100 rounds with a protocol similar to the simulation experiments. We randomly select 1 object from the 10 test objects and put it randomly on the table by hand. The objects are chosen between seen and unseen objects during training and for their availability in the lab. Our success rate of $\mathbf{90}\%$ is similar than in simulation, which indicates that the simulation-to-reality transfer works well. Our approximate ratio learnt successfully several modes to grasp an object and can switch most of the time between them if the path planner fails (Fig .4).

In simulation as well as in the real setup, half of the failure cases are due to the path planner and half are due to wrong hand configurations making the object slip. We leave the improvement of these parts as future work.

VI. Conclusion

We demonstrate the usefulness and applicability of simulation-based Bayesian inference to robotic grasping. Our results show promising performance for determining 6 DoF grasp poses. Nevertheless, our task is rather simple compared to others benchmarks. In the next step, we plan to challenge our method to more complex tasks such as grasping in cluttered environments.

REFERENCES

[1] K. Cranmer, J. Brehmer, and G. Louppe, "The frontier of simulation-based inference," *Proceedings of the National Academy of Sciences*, 2020.

[2] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312.

[3] Q. Lu, K. Chenna, B. Sundaralingam, and T. Hermans, "Planning multi-fingered grasps as probabilistic inference in a learned deep network," in *Robotics Research*. Springer, 2020, pp. 455–472.

[4] 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 Robotics and Automation Letters*, vol. 7, no. 2, pp. 1888–1895, 2022.

[5] M. Breyer, J. J. Chung, L. Ott, S. Roland, and N. Juan, "Volumetric grasping network: Real-time 6 dof grasp detection in clutter," in *Conference on Robot Learning*, 2020.

[6] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International Journal of Robotics Research*, vol. 39, p. 027836491985906, 06 2019.

[7] 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)*. IEEE, 2020, pp. 11 516–11 522.

[8] J.-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder, "Approximate bayesian computational methods," *Statistics and Computing*, vol. 22, no. 6, pp. 1167–1180, 2012.

[9] M. A. Beaumont, W. Zhang, and D. J. Balding, "Approximate bayesian computation in population genetics," *Genetics*, vol. 162, no. 4, pp. 2025–2035, 2002.

[10] G. Papamakarios, D. Sterratt, and I. Murray, "Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 837–848.

[11] G. Papamakarios and I. Murray, "Fast ϵ-free inference of simulation models with bayesian conditional density estimation," in *Advances in Neural Information Processing Systems*, 2016.

[12] J.-M. Lueckmann, P. J. Goncalves, G. Bassetto, K. Öcal, M. Nonnenmacher, and J. H. Macke, "Flexible statistical inference for mechanistic models of neural dynamics," in *Advances in Neural Information Processing Systems*, 2017.

[13] D. Greenberg, M. Nonnenmacher, and J. Macke, "Automatic posterior transformation for likelihood-free inference," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2404–2414.

[14] J. Hermans, V. Begy, and G. Louppe, "Likelihood-free MCMC with amortized approximate ratio estimators," in *Proceedings of the 37th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, H. D. III and A. Singh, Eds., vol. 119. PMLR, 13–18 Jul 2020, pp. 4239–4248. [Online]. Available: http://proceedings.mlr.press/v119/hermans20a.html

[15] N. De Cao and W. Aziz, "The power spherical distribution," *arXiv preprint arXiv:2006.04437*, 2020.

[16] J. Hermans, A. Delaunoy, F. Rozet, A. Wehenkel, and G. Louppe, "Averting a crisis in simulation-based inference," *arXiv preprint arXiv:2110.06581*, 2021.

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

[18] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, Y. Zhu, J. Tremblay, S. Birchfield, G. Shi, F. Ramos, A. Anandkumar, *et al.*, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[19] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

[20] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.