Learning Diffusion Models in SE(3) for 6DoF Grasp Pose Generation

Julen Urain*[1], Niklas Funk*[1], Jan Peters[1,2,3,4], Georgia Chalvatzaki[1]

***Abstract*— In this work, our objective is to investigate the potential of employing diffusion models for point cloud-based 6 DoF grasp pose generation. Recently, diffusion models have demonstrated remarkable success as generative models for images, surpassing earlier models such as Variational Auto-Encoders and Generative Adversarial Networks. Unlike their predecessors, diffusion models do not directly generate samples from the learned model. Instead, they determine the energy or score of the data distribution, and during inference, utilize it in an inverse diffusion process.**

In the field of robotics, generative models have been especially common for grasp pose selection. Due to the inherent complexity of devising grasp poses for a wide range of objects, it is common to develop 6 DoF grasp pose generators using data-driven approaches. However, unlike images that typically exist in Euclidean space, grasp poses reside in the Lie group SE(3), necessitating modifications in the generative models to accommodate to this space.

In our research, we modified diffusion models to facilitate learning generative models within the Lie group SE(3). As SE(3) is not a Euclidean space, we implemented several adjustments to both the training and sampling algorithms to accurately generate samples in SE(3). Subsequently, we trained a point cloud-dependent 6 DoF grasp generative model. Our findings revealed that SE(3) diffusion models surpassed previous methods for 6 DoF grasp generation in terms of successful grasping and distribution coverage. Videos, code, and additional details are available at: `https://sites.google.com/view/se3dif`

I. INTRODUCTION

Grasp selection is a crucial problem in robotic manipulation. When presented with an arbitrary object, the task is to choose a 6 DoF pose (3D position + orientation) that enables successful gripping of the object. Proper grasp selection requires consideration of factors such as the object's geometry, gripper's geometry, and object's mass distribution. Classical approaches to grasp selection relied on geometric-based heuristic methods to select grasp points on objects. However, these models are heavily dependent on the object's mesh, limiting their applicability to realistic scenarios where only camera observations might be available.

Data-driven models have gained popularity for handling unstructured scenarios. Using a dataset of suitable grasp poses for a diverse set of objects, a generative model learns the underlying data distribution. Then, given an observation (usually a point cloud) of the scene or the object to be grasped, the learned model can generate a set of candidate grasp poses. The community has explored various methods for 6-DoF grasp generation. In [1]–[3], an array of grasps is generated through heuristics, and a learned classifier evaluates the quality of these generated grasps, selecting the satisfactory ones. In [4], a two-step generative model is proposed: a Variational Autoencoders (VAE) generates grasp pose candidates, which are then refined by a trained classifier. Nevertheless, learning the classifier remains crucial, as current generative models struggle to generate accurate grasp poses.

Our work is inspired by recent advances in generative modeling. Diffusion models [5]–[9] have achieved remarkable results in text-to-image generation and surpassed previous generative models, such as Generative Adversarial Networks (GAN) [10] or VAE [11], in terms of image quality. Contrary to prior methods that learn explicit sampling models, diffusion models aim to learn a vector field corresponding to the score function of a noisy data distribution. Samples are then generated through a diffusion process based on the learned vector field. We hypothesize that similar improvements in quality observed for images can be achieved for 6 DoF grasp pose generation. Additionally, due to their implicit nature, diffusion models can be incorporated into robotics in innovative ways. While VAE or GAN have been primarily used for direct sampling, diffusion models can also serve as cost functions and be integrated into multi-objective problems, such as motion optimization problems.

This work has two primary contributions. First, we investigate the application of diffusion models for non-Euclidean spaces, such as SE(3). Unlike images, which are usually represented in Euclidean space, grasp poses reside in the Lie group SE(3). This necessitates several modifications to both the training and sampling algorithms. Second, we demonstrate how to learn diffusion models for 6DoF grasping by leveraging widely annotated open-source 6DoF grasp pose datasets like Acronym [12]. SE(3) diffusion models enable the transformation of initially random samples into low-cost regions (regions with suitable grasping poses on objects) through an inverse diffusion process [13] (cf. Fig. 1). SE(3) diffusion models offer two benefits: First, they better represent and cover multimodal distributions, such as those in 6DoF grasp generation scenarios, leading to wider variety of grasp poses. Second, due to their implicit nature, they can be exploited as costs or cost gradients in motion optimization problems in contrast with other generative models.

∗ Authors contributed equally.

This work received funding by the DFG Emmy Noether Programme (CH 2676/1-1), by the AICO grant by the Nexplore/Hochtief Collaboration with TU Darmstadt, and the EU project ShareWork.

[1] Technische Universität Darmstadt (Germany), [2] German Research Center for AI (DFKI), [3] Hessian.AI, [4] Centre for Cognitive Science {julen.urain, niklas.funk, jan.peters, georgia.chalvatzaki} @tu-darmstadt.de

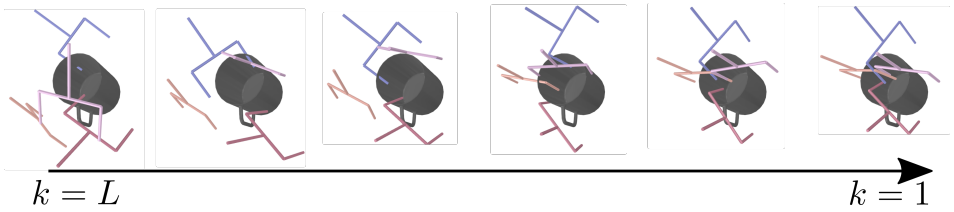

Fig. 1: Generating high quality SE(3) grasp poses by iteratively refining random initial samples (k=L) with an inverse Langevin diffusion process over SE(3) elements (Eq. (6)).

II. Preliminaries

Diffusion Models. Unlike common deep generative models (VAE, GAN) that explicitly generate a sample from a noise signal, Diffusion models learn to generate samples by iteratively moving noisy random samples towards a learned distribution [5], [14]. A common approach to train diffusion models is by *Denoising Score Matching (DSM)* [15], [16]. To apply DSM [14], [17], we first perturb the data distribution $\rho_{\mathcal{D}}(\boldsymbol{x})$ with Gaussian noise on L noise scales $\mathcal{N}(\mathbf{0}, \sigma_k \boldsymbol{I})$ with $\sigma_1 < \sigma_2 < \cdots < \sigma_L$, to obtain a noise perturbed distribution $q_{\sigma_k}(\hat{\boldsymbol{x}}) = \int_{\boldsymbol{x}} \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{x}, \sigma_k \boldsymbol{I}) \rho_{\mathcal{D}}(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}$. To sample from the perturbed distribution, $q_{\sigma_k}(\hat{\boldsymbol{x}})$ we first sample from the data distribution $\boldsymbol{x} \sim \rho_{\mathcal{D}}(\boldsymbol{x})$ and then add white noise $\hat{\boldsymbol{x}} = \boldsymbol{x} + \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_k \boldsymbol{I})$. Next, we estimate the score function of each noise perturbed distribution $\nabla_{\boldsymbol{x}} \log q_{\sigma_k}(\boldsymbol{x})$ by training a noise-conditioned vector field $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, k)$, by score matching $\boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}, k) \approx \nabla_{\boldsymbol{x}} \log q_{\sigma_k}(\boldsymbol{x})$ for all $k = 1, \ldots, L$. The training objective of DSM [16] is

$$\mathcal{L}_{\text{dsm}} = \frac{1}{L} \sum_{k=0}^{L} \mathbb{E}_{\boldsymbol{x}, \hat{\boldsymbol{x}}} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}, k) - \nabla_{\hat{\boldsymbol{x}}} \log \mathcal{N}(\hat{\boldsymbol{x}}|\boldsymbol{x}, \sigma_k^2 \boldsymbol{I}) \right\|_2^2 \right], \quad (1)$$

with $\boldsymbol{x} \sim \rho_{\mathcal{D}}(\boldsymbol{x})$ and $\hat{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}, \sigma_k \boldsymbol{I})$. To generate samples from the trained model, we apply Annealed Langevin Markov Chain Monte Carlo (MCMC) [18]. We first draw an initial set of samples from a distribution $\boldsymbol{x}_L \sim \rho_L(\boldsymbol{x})$ and then, simulate an inverse Langevin diffusion process for L steps, from $k = L$ to $k = 1$

$$\boldsymbol{x}_{k-1} = \boldsymbol{x}_k + \frac{\alpha_k^2}{2} \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{x}_k, k) + \alpha_k \boldsymbol{\epsilon}\,, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I}), \quad (2)$$

with $\alpha_k > 0$ a step dependent coefficient. Overall, DSM Eq. (1) learns vector fields that point towards the samples of the training dataset $\rho_{\mathcal{D}}(\boldsymbol{x})$ [13].

SE(3) Lie group. The SE(3) Lie group is prevalent in robotics. A point $\boldsymbol{H} = \left[\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0} & 1 \end{smallmatrix}\right] \in$ SE(3) represents the full pose (position and orientation) of an object or robot link with $\boldsymbol{R} \in$ SO(3) the rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ the 3D position. A Lie group encompasses the concepts of group and smooth manifold in a unique body. Lie groups are smooth manifolds whose elements have to fulfil certain constraints. Moving along the constrained manifold is achieved by selecting any velocity withing the space tangent to the manifold at $\boldsymbol{H}$ (i.e., the so-called tangent space). The tangent space at the identity is called *Lie algebra* and noted $\mathfrak{se}(3)$. The Lie algebra has a non-trivial structure, but is isomorphic to the vector space $\mathbb{R}^6$ in which we can apply linear algebra. As in [19], we work in the vector space $\mathbb{R}^6$ instead of the Lie algebra $\mathfrak{se}(3)$. We can move the elements between the Lie group and the vector space with the logarithmic and exponential maps, Logmap : SE(3) $\to \mathbb{R}^6$ and Expmap : $\mathbb{R}^6 \to$ SE(3) respectively [19]. A Gaussian distribution on Lie groups can be defined as

$$q(\boldsymbol{H}|\boldsymbol{H}_\mu, \boldsymbol{\Sigma}) \propto \exp\left(-0.5 \ \left\|\text{Logmap}(\boldsymbol{H}_\mu^{-1}\boldsymbol{H})\right\|_{\boldsymbol{\Sigma}^{-1}}^2\right), \quad (3)$$

with $\boldsymbol{H}_\mu \in SE(3)$ the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{6\times6}$ the covariance matrix [20]. This special form is required as the distance between two Lie group elements is not represented in Euclidean space. Following the notation of [19], given a function f : SE(3) $\to \mathbb{R}$, the derivative w.r.t. a SE(3) element, $Df(\boldsymbol{H})/D\boldsymbol{H} \in \mathbb{R}^6$ is a vector of dimension 6. We refer the reader to [19] and the Appendix in project site for an extended presentation of the SE(3) Lie group.

III. SE(3) Diffusion Models

In this section, we show how to adapt diffusion models to the Lie group SE(3) [19], as it is a crucial space for robot manipulation. The SE(3) space is not Euclidean, hence, multiple design choices need to be considered for adapting Euclidean diffusion models. In the following, we explain the required modifications and show how they can be applied in practise.

A. From Euclidean diffusion to diffusion in SE(3)

A Diffusion Model in SE(3), is a *vector field* that outputs a 6 dimensional vector $\boldsymbol{v} \in \mathbb{R}^6$ for an arbitrary query point $\boldsymbol{H} \in$ SE(3), i.e., $\boldsymbol{v} = \boldsymbol{s}_{\boldsymbol{\theta}}(\boldsymbol{H}, k)$ with a scalar conditioning variable k determining the current noise scale [14].

Denoising Score Matching in SE(3). Similar to the Euclidean space version (cf. Sec. II), DSM is applied in two phases. We first generate a perturbed data point in SE(3), i.e., sample from the Gaussian on Lie groups Eq. (3), $\hat{\boldsymbol{H}} \sim q(\hat{\boldsymbol{H}}|\boldsymbol{H}, \sigma_k \boldsymbol{I})$ with mean $\boldsymbol{H} \in \rho_{\mathcal{D}}(\boldsymbol{H})$ and standard deviation σ_k for noise scale k. Practically, we sample from this distribution using a white noise vector $\boldsymbol{\epsilon} \in \mathbb{R}^6$,

$$\hat{\boldsymbol{H}} = \boldsymbol{H}\text{Expmap}(\boldsymbol{\epsilon})\,, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_k^2 \boldsymbol{I}). \quad (4)$$

Following the idea of DSM, the model is trained to match the score of the perturbed training data distribution. Thus, DSM in SE(3) requires computing the derivatives of the perturbed distribution w.r.t. a Lie group element. Hence, the new DSM loss function on Lie groups equates to

$$\mathcal{L}_{\text{dsm}} = \frac{1}{L} \sum_{k=0}^{L} \mathbb{E}_{\boldsymbol{H}, \hat{\boldsymbol{H}}} \left[\left\| \boldsymbol{s}_{\boldsymbol{\theta}}(\hat{\boldsymbol{H}}, k) - \frac{D \log q(\hat{\boldsymbol{H}}|\boldsymbol{H}, \sigma_k \boldsymbol{I})}{D\hat{\boldsymbol{H}}} \right\|_2^2 \right], \quad (5)$$

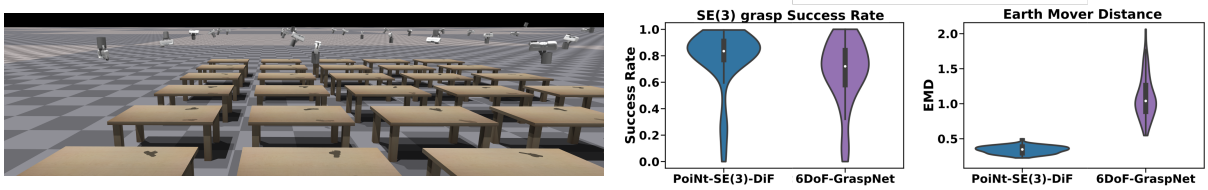

Fig. 2: Left: The success of the generated grasp poses is evaluated in Isaac Gym. Right: Success Rate and EMD evaluation.

with $\boldsymbol{H} \sim \rho_{\mathcal{D}}(\boldsymbol{H})$ and $\hat{\boldsymbol{H}} \sim q(\hat{\boldsymbol{H}}|\boldsymbol{H}, \sigma_k \boldsymbol{I})$. Note that, as introduced in Sec. II, the derivatives w.r.t. a SE(3) element $\hat{\boldsymbol{H}}$ outputs a vector on $\mathbb{R}^6$. In practice, we compute this derivative by automatic differentiation using Theseus [21] library along with PyTorch. We present in Algorithm 1 the training pipeline.

Algorithm 1: SE(3) Diffusion Model Training

Given: $\boldsymbol{\theta}_0$: initial params for $\boldsymbol{s_\theta}$;
Datasets: $\mathcal{D}_g^m : \{\boldsymbol{H}\}$ succesful grasp poses for object m;

for $s \leftarrow 0$ **to** $S-1$ **do**
  $k, \sigma_k \leftarrow [0, \dots, L]$; // sample noise scale
  $\boldsymbol{H} \sim \mathcal{D}_g^m$; // Sample success grasp poses for obj. m
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_k \boldsymbol{I})$; // sample white noise on k scale
  $\hat{\boldsymbol{H}} = \boldsymbol{H}\text{Expmap}(\boldsymbol{\epsilon})$; // perturb grasp pose Eq. (4)
  $\boldsymbol{v}_{\text{T}} = \text{get\_grad}(\log q(\hat{\boldsymbol{H}}|\boldsymbol{H}, \sigma_k \boldsymbol{I}))$; // Compute grad with Theseus
  $\boldsymbol{v} = \boldsymbol{s_\theta}(\hat{\boldsymbol{H}}, k)$; // compute vector
  $L_{\text{dsm}} = \text{MSE}(\boldsymbol{v}, \boldsymbol{v}_T)$; // Compute dsm loss Eq. (5)
  Parameter update
  $\boldsymbol{\theta}_{s+1} = \text{ADAM}(\boldsymbol{\theta}_s, \alpha, L_{\text{dsm}})$; // Update parameters
return $\boldsymbol{\theta}^*$;

Sampling with Langevin MCMC in SE(3). Evolving the inverse Langevin diffusion process for SE(3) elements (cf. Fig. 1 for visualization) requires adapting the previously presented Euclidean Langevin MCMC approach Eq. (2). In particular, we have to ensure staying on the SE(3) manifold throughout the inverse diffusion process. Thus, we adapt the inverse diffusion to SE(3) as

$$\boldsymbol{H}_{k-1} = \boldsymbol{H}_k \text{Expmap}\left(-\frac{\alpha_k^2}{2}\boldsymbol{s_\theta}(\boldsymbol{H}_k, k) + \alpha_k \boldsymbol{\epsilon}\right), \tag{6}$$

with $\boldsymbol{\epsilon} \in \mathbb{R}^6$ sampled from $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and the step dependent coefficient $\alpha_k > 0$. By iteratively applying Eq. (6), we move a set of randomly sampled SE(3) poses to the low energy regions of $E_{\boldsymbol{\theta}}$, i.e. good grasp pose regions (See Fig. 1). We present the sampling pipeline in Algorithm 2.

Algorithm 2: SE(3) Diffusion Model Sampling

Given: $\boldsymbol{s_\theta}$: trained model, $\{\sigma_k\}_k^L$ noise scales, β scale, T steps time;

$\boldsymbol{H}_L \sim q(\boldsymbol{I}, \sigma_L \boldsymbol{I})$
for $k \leftarrow L$ **to** 0 **do**
  $\alpha_k = \beta \sigma_k / \sigma_0$
  for $t \leftarrow 0$ **to** T **do**
    $\epsilon \sim \mathcal{N}(0, \sigma_k \boldsymbol{I})$
    $\boldsymbol{H}_{k-1} = \boldsymbol{H}_k \text{Expmap}\left(-\frac{\alpha_k^2}{2}\boldsymbol{s_\theta}(\boldsymbol{H}_k, k) + \alpha_k \boldsymbol{\epsilon}\right)$
return $\boldsymbol{H}_0$;

IV. EXPERIMENTAL EVALUATION

In the following, we evaluate SE(3) Diffusion Models as a 6 DoF grasp pose generative model. First, we train a SE(3) Diffusion Model using the Acronym dataset [12]. The dataset contains successful 6 DoF grasp poses for a variety of objects from ShapeNet [22]. We focus on the collection of successful grasp poses for 90 different mugs (approximately 90K 6DoF grasp poses). The model is trained conditioned on the object's point cloud. In our experiment, we obtain the point cloud directly from the meshes from Shapenet.

We evaluate grasp poses generated with our trained model in terms of the success rate, and the Earth Mover Distance (EMD) between the generated grasps and the training data distribution. We consider 90 different mugs and evaluate 200 generated grasps per mug. We evaluate the grasp success on Nvidia Isaac Gym [23] (Fig. 2). The EMD measures the divergence between two empirical probability distributions [24], providing a metric on how similar the generated samples are to the training dataset. We evaluate the performance of our model with respect to 6 DoF-GraspNet [4].

We present the results in Fig. 2. We name our method PoiNt-SE(3)-DiF. In terms of success rate, our model outperforms 6 DoF GraspNet slightly (especially yielding lower variance). We highlight that in contrast with 6 DoF GraspNet, our model considers a single network, while 6 DoF GraspNet requires both a generator and a classifier. Considering grasp diversity, i.e., EMD metric (lower is better), our model outperforms the baseline significantly. A reason for the difference, might be that 6 DoF-GraspNet tend to overfit to specific overrepresented modes of the data distribution. In contrast, our model's samples capture the data distribution more properly. We, therefore, conclude that our method is indeed generating high-quality and diverse grasp poses. We add an extended presentation of the experiment in the Appendix in our project site.

V. CONCLUSION

We proposed Diffusion Models in SE(3) for learning data-driven generative models for 6 DoF grasp poses. We have shown that our model outperformed in terms both success rate and data distribution covering to previous state-of-the-art methods. Given their inherent implicit behavior, in the future, we want to explore diffusion models for reactive motion control, exploiting them for gradient-based MPC. Also, we would like to explore learnin SE(3) Diffusion Models that could adapt to multiple conditioning inputs such as language and also apply them to represent target pose distributions of arbitrary objects.

REFERENCES

[1] X. Lou, Y. Yang, and C. Choi, "Collision-aware target-driven object grasping in constrained environments," in *IEEE International Conference on Robotics and Automation*, 2021.

[2] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, 2017.

[3] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, "Pointnetgpd: Detecting grasp configurations from point sets," in *International Conference on Robotics and Automation*, 2019.

[4] A. Mousavian, C. Eppner, and D. Fox, "6-DoF graspnet: Variational grasp generation for object manipulation," in *International Conference on Computer Vision*, 2019.

[5] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *International Conference on Learning Representations*, 2020.

[6] C. Luo, "Understanding diffusion models: A unified perspective," 2022. [Online]. Available: https://arxiv.org/abs/2208.11970

[7] C.-W. Huang, M. Aghajohari, A. J. Bose, P. Panangaden, and A. Courville, "Riemannian diffusion models," *arXiv preprint arXiv:2208.07949*, 2022.

[8] V. D. Bortoli, E. Mathieu, M. J. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet, "Riemannian score-based generative modelling," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=oDRQGo8I7P

[9] D. Gnaneshwar, B. Ramsundar, D. Gandhi, R. Kurchin, and V. Viswanathan, "Score-based generative models for molecule generation," *arXiv preprint arXiv:2203.04698*, 2022.

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, 2014.

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

[12] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *IEEE International Conference on Robotics and Automation*, 2021.

[13] Y. Song and D. P. Kingma, "How to train your energy-based models," *arXiv preprint arXiv:2101.03288*, 2021.

[14] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," *Advances in Neural Information Processing Systems*, 2019.

[15] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural computation*, 2011.

[16] S. Saremi, A. Mehrjou, B. Schölkopf, and A. Hyvärinen, "Deep energy estimator networks," *arXiv preprint arXiv:1805.08306*, 2018.

[17] Y. Song and S. Ermon, "Improved techniques for training score-based generative models," *Advances in Neural Information Processing Systems*, 2020.

[18] R. M. Neal *et al.*, "Mcmc using hamiltonian dynamics," *Handbook of markov chain monte carlo*, 2011.

[19] J. Sola, J. Deray, and D. Atchuthan, "A micro lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018.

[20] G. Chirikjian and M. Kobilarov, "Gaussian approximation of non-linear measurement models on lie groups," in *IEEE Conference on Decision and Control*, 2014.

[21] L. Pineda, T. Fan, M. Monge, S. Venkataraman, P. Sodhi, R. Chen, J. Ortiz, D. DeTone, A. Wang, S. Anderson *et al.*, "Theseus: A library for differentiable nonlinear optimization," *arXiv preprint arXiv:2207.09442*, 2022.

[22] 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.

[23] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[24] A. Tanaka, "Discriminator optimal transport," *Advances in Neural Information Processing Systems*, 2019.