SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion

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

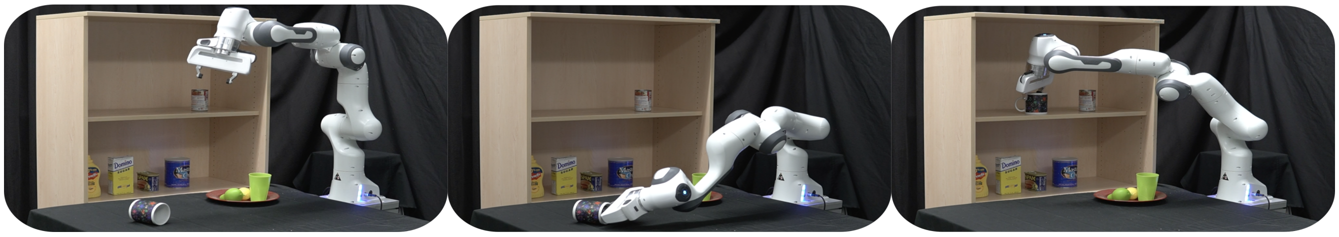

Fig. 1: Pick and place task in which the robot has to pick a mug and move it to the target pose (in the shelves) without colliding. We exploit diffusion models for jointly optimizing both grasp and motion and show the successful trajectory from left to right.

***Abstract*— Multi-objective optimization problems are ubiquitous in robotics, e.g., the optimization of a robot manipulation task requires a joint consideration of grasp pose configurations, collisions and joint limits. While some demands can be easily hand-designed, e.g., the smoothness of a trajectory, several task-specific objectives need to be learned from data. This work introduces a method for learning data-driven SE(3) cost functions as diffusion models. Diffusion models can represent highly-expressive multimodal distributions and exhibit proper gradients over the entire space due to their score-matching training objective. Learning costs as diffusion models allows their seamless integration with other costs into a single differentiable objective function, enabling joint gradient-based motion optimization. In this work, we focus on learning SE(3) diffusion models for 6DoF grasping, giving rise to a novel framework for joint grasp and motion optimization without needing to decouple grasp selection from trajectory generation. We evaluate the representation power of our SE(3) diffusion models w.r.t. classical generative models, and we showcase the superior performance of our proposed optimization framework in a series of simulated and real-world robotic manipulation tasks against representative baselines. Videos, code and additional details are available at: `https://sites.google.com/view/se3dif`**

I. INTRODUCTION

Autonomous robot manipulation tasks usually involve complex actions requiring a set of sequential or recurring subtasks to be achieved while satisfying certain constraints, thus, casting robot manipulation into a multi-objective motion optimization problem [1]–[3]. Let us consider the pick-and-place task in Fig. 1, for which the motion optimization should consider the possible set of grasping and placing poses, the trajectories' smoothness, collision avoidance with the environment, and the robot's joint limits. While some objectives are easy to model (e.g., joint limits, smoothness), others (e.g., collision avoidance, grasp pose selection) are more expensive to model and are therefore commonly approximated by learning-based approaches [4]–[8].

Data-driven models are usually integrated into motion optimization either as sampling functions (explicit generators) [6], [9], or cost functions (scalar fields) [4], [10]. When facing multi-objective optimization scenarios, the explicit generators do not allow a direct composition with other objectives, requiring two or even more separate phases during optimization [11]. Looking back at the example of Fig. 1, a common practice is to learn a grasp generator as an explicit model, sample top-k grasps, and then find the trajectory that, initialized by a grasp candidate, solves the task with a minimum cost. Given the grasp sampling is decoupled from the trajectory planning, it might happen the sampled grasps to be unfeasible for the problem, leading to an unsolvable trajectory optimization problem. On the other hand, learned scalar fields represent task-specific costs that can be combined with other learned or heuristic cost functions to form a single objective function for a joint optimization process. However, these cost functions are often learned through *cross-entropy optimization* [6], [12] *or contrastive divergence* [10], [13], creating hard discriminative regions in the learned model that *lead to large plateaus in the learned field with zero or noisy slope regions* [14], [15], thereby making them unsuitable for pure gradient-based optimization. Thus, it is a common strategy to rely on task-specific samplers that first generate samples close to low-cost regions before optimizing [6], [12].

In this work, we propose learning *smooth* data-driven cost functions, drawing inspiration from state-of-the-art diffusion generative models [16], [17]. By *smoothness*, we refer to the cost function exposing informative gradients in the entire space. We propose learning these smooth cost functions in the SE(3) robot's workspace, thus defining task-specific SE(3) cost functions. In particular, in this work, we show how to learn diffusion models for 6DoF grasping, leveraging open-source vastly annotated 6DoF grasp pose datasets like Acronym [18]. SE(3) diffusion models allow moving initially

∗ 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`

random samples to low-cost regions (regions of good grasping poses on objects) by evolving a gradient-based inverse diffusion process [19] (cf. Fig. 2). SE(3) diffusion models come with two benefits. First, we get smooth cost functions in SE(3) that can be directly used in motion optimization. Second, they better cover and represent multimodal distributions, like in a 6DoF grasp generation scenario, leading to better and more sample efficient performance of the subsequent robot planning.

Consequently, we propose a joint grasp and motion optimization framework using the learned 6DoF grasp diffusion model as cost function and combining it with other differentiable costs (trajectory smoothness, collision avoidance, etc.). All costs combined (learned and hand-designed) form a single, smooth objective function that optimizing it enables the generation of good robot trajectories for complex robot manipulation tasks. This work shows how our framework enables facing grasp generation and classical trajectory optimization as a joint gradient-based optimization loop.

Our contributions are threefold: (**1**) we show how to **learn smooth cost functions in SE(3) as diffusion models**. As SE(3) is a non-Euclidean space, we show a practical approach for adapting the training and sampling processes of diffusion models for learning 6DoF costs. Then, (**2**) **we use the SE(3) diffusion models to learn 6DoF grasp pose distributions as cost functions**. In our experiments, we show that our learned models generate more diverse and successful grasp poses w.r.t. state-of-the-art grasp generative models. Once the model is trained, (**3**) we introduce **a gradient-based optimization framework for jointly resolving grasp and motion generation**, in which, we integrate our learned 6DoF grasp diffusion model with additional task related cost terms. In contrast with previous methods that decouple the grasp pose selection and the motion planning, our framework resolves the grasp and motion planning problem by iteratively improving the trajectory to jointly minimize the learned object-grasp cost term and the task related costs. We remark that this joint optimization is only possible thanks to the smoothness of our learned diffusion model and using instead a grasp classifier, trained with cross-entropy loss, as cost won't resolve the problem due to its lack of smoothness. Our quantitative and qualitative results in simulation and the real-world robotic manipulation experiments suggest that our proposed method for learning costs as SE(3) diffusion models enables efficiently finding good grasp and motion solutions against baseline approaches and resolves complex pick-and-place tasks as in Fig. 1.

II. Preliminaries

Diffusion Models. Unlike common deep generative models (Variational Autoencoders (VAE), generative adversarial networks (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 [16], [20]. A common approach to train diffusion models is by *Denoising Score Matching (DSM)* [21], [22]. To apply DSM [20], [23], 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 Energy Based Models (EBM) $E_{\boldsymbol{\theta}}(\boldsymbol{x}, k)$, by score matching $\nabla_{\boldsymbol{x}} E(\boldsymbol{x}, k) \approx -\nabla_{\boldsymbol{x}} \log q_{\sigma_k}(\boldsymbol{x})$ for all $k = 1, \dots, L$. The training objective of DSM [22] is

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

with $\boldsymbol{x} \sim \rho_{\mathcal{D}}(\boldsymbol{x})$ and $\hat{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}, \sigma_k \boldsymbol{I})$[1]. To generate samples from the trained model, we apply Annealed Langevin Markov Chain Monte Carlo (MCMC) [24]. 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 $k = 1$

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

with $\alpha_k > 0$ a step dependent coefficient. Overall, DSM Eq. (1) learns models whose gradients point towards the samples of the training dataset $\rho_{\mathcal{D}}(\boldsymbol{x})$ [19].

SE(3) Lie group. The SE(3) Lie group is prevalent in robotics. A point $\boldsymbol{H} = \left[\begin{smallmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \mathbf{0} & 1 \end{smallmatrix}\right] \in \mathrm{SE}(3)$ represents the full pose (position and orientation) of an object or robot link with $\boldsymbol{R} \in \mathrm{SO}(3)$ 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 [25], 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, $\mathrm{Logmap} : \mathrm{SE}(3) \to \mathbb{R}^6$ and $\mathrm{Expmap} : \mathbb{R}^6 \to \mathrm{SE}(3)$ respectively [25]. A Gaussian distribution on Lie groups can be defined as

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

with $\boldsymbol{H}_\mu \in SE(3)$ the mean and $\boldsymbol{\Sigma} \in \mathbb{R}^{6 \times 6}$ the covariance matrix [26]. This special form is required as the distance between two Lie group elements is not represented in Euclidean space. Following the notation of [25], given a function $f : \mathrm{SE}(3) \to \mathbb{R}$, 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 [25] and the Appendix in project site for an extended presentation of the SE(3) Lie group.

[1]Note that we learn the energy $E_{\boldsymbol{\theta}}$ instead of the score, as its common in the literature [16], [20], to use it as a cost function and evaluate the quality of our samples in an optimization problem

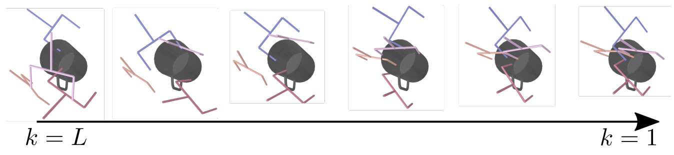

Fig. 2: Generating high quality SE(3) grasp poses by iteratively refining noisy initial samples (k=L) with an inverse Langevin diffusion process over SE(3) elements (Eq. (6)).

III. SE(3)-DIFFUSION FIELDS

In this section, we show how to adapt diffusion models to the Lie group SE(3) [25], 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 first explain the required modifications (Section III-A). Then, we propose a neural network architecture for learning SE(3) diffusion models that represent 6DoF grasp pose distributions and show how we train it (Sec. III-B). Finally, we show how to integrate the learned diffusion models into a grasp and motion optimization problem and show how to optimize it jointly considering the grasp and the motion (Sec. III-C).

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

We call SE(3)-DiffusionFields (SE(3)-DiF) a *scalar field* that provides a scalar value $e \in \mathbb{R}$ for an arbitrary query point $\boldsymbol{H} \in \mathrm{SE}(3)$, i.e., $e = E_{\boldsymbol{\theta}}(\boldsymbol{H}, k)$ with a scalar conditioning variable k determining the current noise scale [20].

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 $\boldsymbol{\epsilon} \in \mathbb{R}^6$,

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

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 model and the perturbed distribution w.r.t. a Lie group element. Hence, the new DSM loss function on Lie groups equates to

$$\mathcal{L}_{\mathrm{dsm}} = \frac{1}{L}\sum_{k=0}^{L}\mathbb{E}_{\boldsymbol{H},\hat{\boldsymbol{H}}}\left[\left\|\frac{DE_{\boldsymbol{\theta}}(\hat{\boldsymbol{H}}, k)}{D\hat{\boldsymbol{H}}} + \frac{D\log q(\hat{\boldsymbol{H}}|\boldsymbol{H}, \sigma_k \boldsymbol{I})}{D\hat{\boldsymbol{H}}}\right\|\right], \tag{5}$$

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 [27] library along with PyTorch.

Sampling with Langevin MCMC in SE(3). Evolving the inverse Langevin diffusion process for SE(3) elements (cf. Fig. 2 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 in SE(3) as

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

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 noisy SE(3) poses to the low energy regions of $E_{\boldsymbol{\theta}}$, i.e. good grasp pose regions (See Fig. 2).

B. Architecture & training of Grasp SE(3)-DiffusionFields

Even though we can represent any data-driven cost in SE(3) with SE(3)-DiF, in this work, we focus on cost functions that capture 6DoF grasp pose distributions conditioned on the object we aim to grasp. In this work, we assume to have access to the object pose, a reasonable assumption thanks to the impressive results in 6DoF object pose estimation and segmentation [28]. We defer studying the perception aspect of encoding point clouds into object pose and shape as in [6], [29] for a future work. We illustrate the architecture for our grasp SE(3)-DiF model in Fig. 3 and the training pipeline in Algorithm 1. The proposed model maps an object (represented by its id and pose) and a 6DoF grasp pose $\boldsymbol{H} \in \mathrm{SE}(3)$ to an energy $e \in \mathbb{R}$, that measures the grasp quality for the particular object.

We train the model to jointly match the Signed Distance Field (SDF) of the object we aim to grasp and predict the grasp energy level by the DSM loss Eq. (5). Learning jointly the SDF of the object and the grasp pose improves the quality of the grasp generation [29], [30]. During the training, we assume the object's id m and pose $\boldsymbol{H}_w^o \in \mathrm{SE}(3)$ are available, and we retrieve a learnable object shape code $\boldsymbol{z}_m$ given the index m as in [31]. For training the SDF loss, we apply a supervised learning pipeline. Given a dataset of 3D points $\boldsymbol{x}_w \in \mathbb{R}^3$ and $\mathrm{sdf} \in \mathbb{R}$ for a particular object m, $\mathcal{D}_{sdf}^m : (\boldsymbol{x}_w, \mathrm{sdf})$, we first map the points to the object's reference frame $\boldsymbol{x}_o = \boldsymbol{H}_w^o \boldsymbol{x}_w$ and then predict the SDF given the feature encoder $F_{\boldsymbol{\theta}}$ (See Algorithm 1).

As previously introduced in Eq. (5), to apply the DSM loss, we compute the energy $e \in \mathbb{R}$ over the grasp poses $\hat{\boldsymbol{H}}$. These grasp poses have been previously obtained by perturbing grasp poses from the dataset $\boldsymbol{H} \in \rho_{\mathcal{D}}(\boldsymbol{H})$ with a noise level k Eq. (4). In our problem, we consider $\rho_{\mathcal{D}}(\boldsymbol{H})$ to be a distribution of successful grasp poses for a particular object, and learn the energy to approximate the log-probability of this distribution under noise. We compute the energy e given a grasp pose $\hat{\boldsymbol{H}}$ in three steps. **(I)** We transform the grasp pose to a fixed set of N 3D-points around the gripper $\boldsymbol{x}_g \in \mathbb{R}^{N\times 3}$ in the world frame $\boldsymbol{x}_w = \boldsymbol{H}\boldsymbol{x}_g$. We thereby express the grasp pose through a set of 3D points' positions, similar to [30]. Then, we move the points to the object's local frame, $\boldsymbol{x}_{o_m} = \boldsymbol{H}_w^{o_m}\boldsymbol{x}_w$. **(II)** We apply the feature encoding network $F_{\boldsymbol{\theta}}$ which is also conditioned on $\boldsymbol{z}_m$ and k to inform about the object shape and noise level, respectively. The encoding network outputs both the SDF predictions for the query points, $\mathrm{sdf} \in \mathbb{R}^{N\times 1}$, and a set of additional features $\boldsymbol{\psi} \in \mathbb{R}^{N\times\psi}$. Thus, the feature encoder's output is of size $N \times (1 + \psi)$. **(III)**

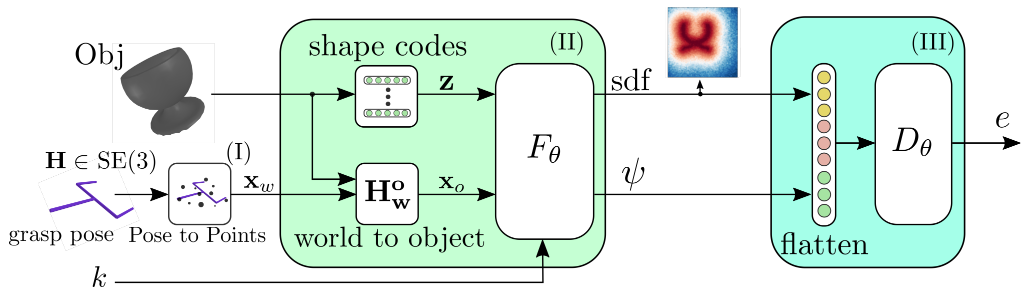

Fig. 3: SE(3)-DiF's architecture for learning 6D grasp pose distributions. We train the model to jointly learn the objects' sdf and to minimize the denoising loss. Given grasp pose $\boldsymbol{H} \in \mathrm{SE}(3)$ we transform it to a set of 3D points $\boldsymbol{x}_w \in \mathbb{R}^{N\times 3}$ (**I**). Next, we transform the points into the object's local frame, using the object's pose $\boldsymbol{H}_w^o$. Given the resulting points $\boldsymbol{x}_o$ and the object's shape code $\boldsymbol{z}$ we apply the feature encoder F_θ (**II**) to obtain a object and grasp-related features (sdf, ψ)$\in \mathbb{R}^{N\times(\psi+1)}$. Finally, (**III**) we flatten the features and compute the energy e through the decoder D_θ.

We flatten the features and pass them through the decoder $D_{\boldsymbol{\theta}}$ to obtain the scalar energy value e. Given the energy, we compute the DSM loss Eq. (5). During training, we jointly learn the objects' latent codes $\boldsymbol{z}_m$, and the parameters $\boldsymbol{\theta}$ of the feature encoder $F_{\boldsymbol{\theta}}$ and decoder $D_{\boldsymbol{\theta}}$.

Algorithm 1: Grasp SE(3)-DiF Training

```
Given: $\boldsymbol{\theta}_0$: initial params for $\boldsymbol{z}$, $F_{\boldsymbol{\theta}}$, $D_{\boldsymbol{\theta}}$;
Datasets: $\mathcal{D}_o : \{m, \boldsymbol{H}_w^o\}$, object ids and poses,
$\mathcal{D}_{sdf}^m : \{\boldsymbol{x}, \mathrm{sdf}\}$, 3D positions $\boldsymbol{x}$ and sdf for object $m$,
$\mathcal{D}_g^m : \{\boldsymbol{H}\}$ succesful grasp poses for object $m$;
for $s \leftarrow 0$ to $S-1$ do
    $k, \sigma_k \leftarrow [0, \dots, L]$;                              // sample noise scale
    $m, \boldsymbol{H}_w^o \in \mathcal{D}_o$;                          // sample objects ids and poses
    $\boldsymbol{z} = \text{shape codes}(m)$;                           // get shape codes
    SDF train
    $\boldsymbol{x}, \mathrm{sdf} \in \mathcal{D}_{sdf}^m$;             // get 3D points and sdf for obj. $m$
    $\hat{sdf}, \_ = F_{\boldsymbol{\theta}}(\boldsymbol{H}_w^o \boldsymbol{x}, \boldsymbol{z}, k)$;   // get predicted sdf
    $L_{\mathrm{sdf}} = \mathcal{L}_{\mathrm{mse}}(\hat{sdf}, sdf)$;     // compute sdf error
    Grasp diffusion train
    $\boldsymbol{H} \sim \mathcal{D}_g^m$;                              // Sample success grasp poses for obj. $m$
    $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_k \boldsymbol{I})$;   // sample white noise on $k$ scale
    $\hat{\boldsymbol{H}} = \boldsymbol{H}\,\mathrm{Expmap}(\boldsymbol{\epsilon})$;   // perturb grasp pose Eq. (4)
    $\boldsymbol{x}_n^o = \hat{\boldsymbol{H}} \boldsymbol{x}_n$;       // Transform to N 3d points (see Figure 3)
    $\hat{sdf}_n, \boldsymbol{\psi}_n = F_{\boldsymbol{\theta}}(\boldsymbol{x}_n^o, \boldsymbol{z}_b, k)$;   // get features
    $\Psi = \mathrm{Flatten}(\hat{sdf}_n, \boldsymbol{\psi}_n)$;        // Flatten the features
    $e = D_{\boldsymbol{\theta}}(\Psi)$;                                // compute energy
    $L_{\mathrm{dsm}} = \mathcal{L}_{\mathrm{dsm}}(e, \hat{\boldsymbol{H}}, \boldsymbol{H}, \sigma_k)$;   // Compute dsm loss Eq. (5)
    Parameter update
    $L = L_{\mathrm{dsm}} + L_{\mathrm{sdf}}$;                          // Sum losses
    $\boldsymbol{\theta}_{s+1} = \boldsymbol{\theta}_s - \alpha \nabla_{\boldsymbol{\theta}} L$;   // Update parameters
return $\boldsymbol{\theta}^*$;
```

C. Grasp and motion optimization with diffusion models

Given a trajectory $\boldsymbol{\tau} : \{\boldsymbol{q}_t\}_{t=1}^T$, consisting of T waypoints, with $\boldsymbol{q}_t \in \mathbb{R}^{d_q}$ the robot's joint positions at time instant t; in motion optimization, we aim to find the minimum cost trajectory $\boldsymbol{\tau}^* = \arg\min_{\boldsymbol{\tau}} \mathcal{J}(\boldsymbol{\tau}) = \arg\min_{\boldsymbol{\tau}} \sum_j \omega_j c_j(\boldsymbol{\tau})$, where the objective function $\mathcal{J}$ is a weighted sum of costs c_j, with weights $\omega_j > 0$. Herein, we integrate the learned SE(3)-DiF for grasp generation as one cost term of the objective function. It is, thus, combined with other heuristic costs, e.g., collision avoidance or trajectory smoothness. Optimizing over the whole set of costs enables obtaining optimal trajectories jointly taking into account grasping, as well as motion-related objectives. This differs from classic grasp and motion planning approaches in which the grasp pose sampling and trajectory planning are treated separately [32], by first sampling the grasp pose, and, then, searching for a trajectory that satisfies the selected grasp. In classic approaches, given the grasp sampling is decoupled from the trajectory planning, it might happen the sampled grasps to be unfeasible for the problem, leading to an unsolvable trajectory planning problem. We hypothesize that jointly optimizing over both the grasp pose and the trajectory allows us to be more sample efficient w.r.t. decoupled approaches.

Given that the learned function is in SE(3) while the optimization is w.r.t. the robot's joint space, we redefine the cost as $c(\boldsymbol{q}_t, k) = E_{\boldsymbol{\theta}}(\phi_{ee}(\boldsymbol{q}_t), k)$, with the forward kinematics $\phi_{ee} : \mathbb{R}^{d_q} \to \mathrm{SE}(3)$ mapping from robot configuration to the robot's end-effectors task space. This cost function provides low cost to those robot configurations that lead to good grasps. To obtain minimum cost trajectories, *we frame the motion generation problem as an inverse diffusion process*. Using a planning-as-inference view [33]–[36], we define a desired target distribution as $q(\boldsymbol{\tau}|k) \propto \exp(-\mathcal{J}(\boldsymbol{\tau}, k))$. This allows us to set an inverse Langevin diffusion process that evolves a set of random initial particles drawn from a distribution $\boldsymbol{\tau}_L \sim p_L(\boldsymbol{\tau})$ towards the target distribution $q(\boldsymbol{\tau}|k)$

$$\boldsymbol{\tau}_{k-1} = \boldsymbol{\tau}_k + 0.5\ \alpha_k^2 \nabla_{\boldsymbol{\tau}_k} \log q(\boldsymbol{\tau}|k) + \alpha_k \boldsymbol{\epsilon}\,,\ \boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}), \quad (7)$$

with step dependent coefficient $\alpha_k > 0$, noise level moving from $k = L$ to $k = 1$, and one particle corresponding to an entire trajectory. If we evolve the particles by this inverse diffusion process for sufficient steps, the particles at $k = 1$, $\boldsymbol{\tau}_1$ can be considered as particles sampled from $q(\boldsymbol{\tau}|k = 1)$. To obtain the optimal trajectory, we evaluate the samples on $\mathcal{J}(\boldsymbol{\tau}, 1)$ and pick the one with the lowest cost.

IV. EXPERIMENTAL EVALUATION

The experimental section is divided in three parts. First, we evaluate our trained model for 6DoF grasp pose generation (Sec. IV-A). We train a SE(3)-DiF as a 6DoF grasp pose generative model using the Acronym dataset [18]. This simulation-based dataset contains successful 6DoF grasp poses for a variety of objects from ShapeNet [37]. We focus on the collection of successful grasp poses for 90 different mugs (approximately 90K 6DoF grasp poses). We obtain the mugs' meshes from ShapeNet, and train the model as described in Algorithm 1. We generate a set of grasp poses from the learned models and evaluate on successful grasping and diversity. Second, we evaluate the quality of our trained model when used as an additional cost term for grasp and motion optimization (Sec. IV-B). We compare the performance of solving a grasp and motion optimization problem

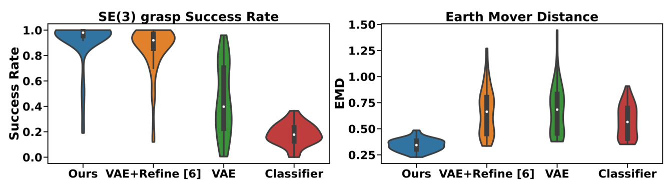

Fig. 4: 6D grasp pose generation experiment. Left: Success rate evaluation. Right: Earth Mover Distance (EMD) evaluation metrics (lower is better).

jointly (using the learned model as cost function), w.r.t. the state-of-the-art approaches that decouple the grasp selection and motion planning, or heuristically combine them. Finally, we validate the performance of our method in a set of real robot experiments (Sec. IV-C).

A. Evaluation of 6DoF grasp pose generation

We evaluate grasp poses generated from our trained grasp SE(3)-DiF model in terms of the success rate, and the 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 [38]. The EMD measures the divergence between two empirical probability distributions [39], providing a metric on how similar the generated samples are to the training dataset. To eliminate any other influence, we only consider the gripper and assume that we can set it to any arbitrary pose. We generate 6DoF grasp poses from SE(3)-DiF by an inverse diffusion process, following Eq. (6).
We compare against three baselines. First, based on [6], [40], we consider generating grasp poses by first sampling from a decoder of a trained VAE and subsequently running MCMC over a trained classifier for pose refinement (VAE+Refine). Second, we consider sampling from the VAE (without any further refinement). Third, we consider running MCMC over the classifier starting from random initial pose [41]. In this experiment, we assume the object's pose and id/shape to be known, and purely focus on evaluating the models' generative capabilities. For ensuring a fair comparison, all the baselines consider a shape code z_m to encode the object information as presented in Fig. 3. We add a pointcloud-conditioned experiment in the Appendix.
We present the results in Fig. 4. In terms of success rate, SE(3)-DiF outperforms VAE+Refine slightly (especially yielding lower variance), and VAE or classifier on their own significantly. The VAE alone generates noisy grasp poses that are often in collision with the mug. In the case of classifier only, the success rate is low. We hypothesize that this might be related with the classifier's gradient, as specifically in regions far from good samples, the field has a large plateau with close to zero slopes [14]. This leads to not being able to improve the initial samples. Considering grasp diversity, i.e., EMD metric (lower is better), SE(3)-DiF outperforms all baselines significantly. A reason for the difference, might be that VAE+Refine overfits to specific overrepresented modes of the data distribution. In contrast, SE(3)-DiF's samples capture the data distribution more properly. We, therefore, conclude that SE(3)-DiF is indeed generating high-quality and diverse grasp poses. We add an extended presentation of the experiment in the Appendix in our project site.

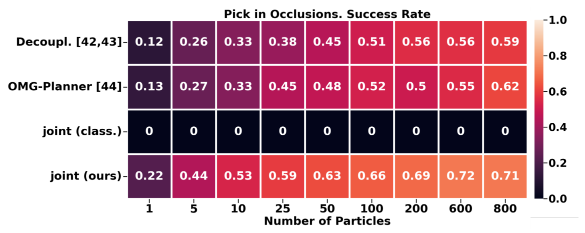

Fig. 5: Evaluation Pick in occlusion. We measure the success rate of 4 different methods based on different number of initializations.

B. Performance on grasp and motion optimization

We evaluate the performance of our learned grasp SE(3)-DiF as a cost term into multi-objective grasp and motion optimization problems. We consider the task of picking amidst clutter (see Fig. 6) and measure the success rate on solving it. The success is measured based on the robot being able to grasp the object at the end of the execution. In the Appendix in our project site, we provide additional details on the chosen cost functions for the task. As introduced in Sec. III-C, we generate the trajectories by integrating our learned grasp SE(3)-DiF as an additional cost function to the motion optimization objective function. Then, given a set of initial trajectory samples, obtained from a Gaussian distribution with a block diagonal matrix as in [2], we apply gradient descent methods Eq. (7) to iteratively improve the trajectories on the objective function. We evaluate the success rate of the trajectory optimization given a different number of initial samples. As gradient-based trajectory optimization methods are inherently local optimization methods, multiple initializations might lead to better results. We consider three baselines (see Fig. 5). **Decoupl.**: we adopt the common routing to solve grasp and motion optimization problems in a decoupled way [11], [42], [43]. We first sample a set of 6DoF grasp poses from a generative model and then plan a trajectory that satisfies the selected grasp pose with CHOMP [1]. Second, we consider the **OMG-Planner** [44], that applies an online grasp selection and planning approach. Finally, **joint (class.)**: we consider applying a joint optimization as in our approach, but using a 6DoF grasp classifier as cost function rather than a grasp SE(3)-DiF.
The results in Fig. 5 present a clear benefit from the joint optimization w.r.t. the decoupled approach and the OMG-Planner. In particular, our proposed joint optimization only requires 25 particles to match the success rate of the decoupled approach with 800 particles. The reason for this significant gap in efficiency is that the decoupled approach generates SE(3) grasp poses that are not feasible given the environment constraints, such as clutter or joint limits. However, when optimizing jointly, we can find trajectories that satisfy all the costs by iteratively improving entire trajectories w.r.t. all objectives. We also observe the importance of using grasp SE(3)-DiF as cost term instead of a grasp classifier. The classifier model lacks proper gradient information to inform how to move the trajectories to grasp the object due to its lack of smoothness in the whole space. Thus, the motion optimization problem is unable to find solutions.

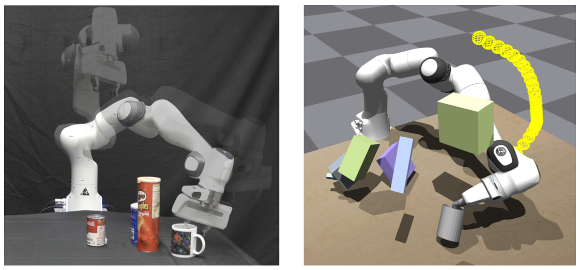
Fig. 6: Simulated and real robot environments for picking amidst clutter.

C. Grasp and motion optimization on real robots

We conducted a thorough real-world evaluation of our joint grasp and motion optimization framework driven by our 6DoF grasp diffusion model, using it as an additional cost function, similarly to our simulated robot manipulation tasks. Fig. 1 depicts a sequence of a real-world pick-mug and place-on-shelf scenario. Overall, the experiments aim at assessing the method's capabilities in realistic conditions that include, i) non-perfect state information, as the mugs pose is retrieved from an external system (Optitrack) which induces small calibration errors, ii) variations in the mug's shape, as we use a mug that is slightly different from the one we specify for SE(3)-DiF, and iii) real-world trajectory execution. For optimization, we initialize 800 particles (trajectories), and only execute the one with lowest cost.
In the simplest testing scenario, where the robot has to pick up a mug from various poses in a scene without any clutter, we achieve **100%** (20 successes / 20 trials) pickup-success. We also find that our method transfers well to the more difficult scenarios of picking up mugs that are initially placed upside down with **90%** (18/20) success, picking in occluded scenes with **95%** (19/20) success, and having to pick and place the mug in a desired pose inside the shelf of Fig. 1 with **100%** (20/20) success. Our real-world results underline the effectiveness of our joint optimization approach. Videos of the experiments also showcase that our method still comes up with very versatile solutions[2]. Note that we attribute the increased real-world performance w.r.t. the simulated one to the simpler designed experimental scene, i.e., in simulation we considered flying obstacles that were not realizable in the real scene (Fig. 6). Nevertheless, our results confirm that our proposed approach is highly performant in real settings, without suffering sim2real discrepancies.
Limitations In our experiments, we focused on evaluating our diffusion model's performance in grasp generation, besides full trajectory optimization, assuming full object state knowledge, without relying on complex perception systems. Potential sim2real gaps w.r.t. the real environment could potentially arise from imperfect perception, and hand-designed cost terms that may not capture well the relevant task description in more complex scenarios. Moreover, a limitation comes with increasing number of cost terms, as it becomes more difficult to weight them. In the future, we consider adding automatic tuning of these hyperparameters (i.e., Bayesian Optimization) to make the method scale better.

[2]Videos in `https://sites.google.com/view/se3dif`

V. RELATED WORK

Learning cost functions. Learned cost functions are prevalent in robotics research [45]. Particularly when it comes to constructing SDFs for motion planning [46], or using SDFs for forming optimization costs [47]. Learning implicit representations of objects can be leveraged as costs for motion optimization [10], [48] and control [49]. Highly related to our work is the field of Inverse Reinforcement Learning [13], [50]–[52], where the goal is to learn a cost function from demonstrations.
6D grasp generation. 6D grasp pose generation is solved with a myriad of methods from classifiers to explicit samplers. [41], [53], [54] sample candidate grasps and score them with learned classifiers. [55] predicts grasping outcomes using a geometry-aware representation. Contrary to methods classifying grasps, generative models can be trained to generate grasp poses from data [6] but might require additional sample refinement. While the generator in [40] considers possible collisions in the scene, [56] proposes to learn a grasp distribution over the object's manifold. [29] uses scene representation learning to learn grasp qualities and explicitly predict 3D rotations.
Integrated grasp and motion planning. Due to the interdependence of the selected grasp pose with the robot motion, multiple efforts have tried to integrate both variables into a single planning problem. In [57], [58], goal sets representing grasp poses are integrated as constraints in a motion optimization problem. In [59], [60], Rapidly-exploring Random Trees [61] is combined with a TCP attractor to bias the tree towards good grasps. [44] proposes an iterative procedure to optimize both the grasp pose and the motion. Our work differs from previous methods as we propose to set the grasp pose objective as a learned cost function in a gradient-based motion optimization problem.

VI. CONCLUSION

We proposed SE(3)-DiffusionFields (SE(3)-DiF) for learning task-space, data-driven cost functions to enable robotic motion generation through joint gradient-based optimization over a set of combined cost functions. At the core of SE(3)-DiFs is a diffusion model that provides informative gradients across the entire space and enables data generation through an inverse Langevin dynamics diffusion process. Besides having demonstrated that SE(3)-DiF generates diverse and high-quality 6DoF grasp poses, we also drew a connection between motion generation and inverse diffusion. Thus, we presented a joint gradient-based grasp and motion optimization framework, which outperforms traditional decoupled optimization approaches. Our extensive experimental evaluations reveal the superior performance of the proposed method w.r.t. efficiency, adaptiveness, and success rates. In the future, we want to explore diffusion models for reactive motion control and the composition of multiple diffusion models to solve complex manipulation tasks in which multiple hard-to-model objectives might arise.

References

[1] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation*, 2009.

[2] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *IEEE International Conference on Robotics and Automation*, 2011.

[3] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, 2014.

[4] D. Rakita, B. Mutlu, and M. Gleicher, "RelaxedIK: Real-time synthesis of accurate and feasible robot arm motion." in *Robotics: Science and Systems*, 2018.

[5] T. Osa, "Motion planning by learning the solution manifold in trajectory optimization," *The International Journal of Robotics Research*, 2022.

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

[7] J. Urain, M. Ginesi, D. Tateo, and J. Peters, "Imitationflows: Learning deep stable stochastic dynamic systems by normalizing flows," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[8] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *International Conference on Robotics and Automation*. IEEE, 2022.

[9] D. Koert, G. Maeda, R. Lioutikov, G. Neumann, and J. Peters, "Demonstration based trajectory optimization for generalizable robot motions," in *IEEE-RAS International Conference on Humanoid Robots*, 2016.

[10] A. Lambert, A. T. Le, J. Urain, G. Chalvatzaki, B. Boots, and J. Peters, "Learning implicit priors for motion optimization," *IEEE International Conference on Intelligent Robots and Systems*, 2022.

[11] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, "6-DoF grasping for target-driven object manipulation in clutter," in *IEEE International Conference on Robotics and Automation*, 2020.

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

[13] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," in *International Conference on Machine Learning*, 2016.

[14] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," *arXiv preprint arXiv:1701.04862*, 2017.

[15] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," in *International Conference on Learning Representations*, 2018.

[16] 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.

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

[18] 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.

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

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

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

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

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

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

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

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

[27] 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.

[28] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 10 367–10 373.

[29] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations," in *Robotics: Science and Systems*, 2021.

[30] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se(3)-equivariant object representations for manipulation," in *International Conference on Robotics and Automation*, 2022.

[31] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[32] F. Lagriffoul, D. Dimitrov, J. Bidot, A. Saffiotti, and L. Karlsson, "Efficiently combining task and motion planning using geometric constraints," *The International Journal of Robotics Research*, 2014.

[33] M. Botvinick and M. Toussaint, "Planning as inference," *Trends in cognitive sciences*, 2012.

[34] S. Levine, "Reinforcement learning and control as probabilistic inference: Tutorial and review," *arXiv preprint arXiv:1805.00909*, 2018.

[35] J. Urain, P. Liu, A. Li, C. D'Eramo, and J. Peters, "Composable Energy Policies for Reactive Motion Generation and Reinforcement Learning ," in *Robotics: Science and Systems*, 2021.

[36] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[37] 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.

[38] 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.

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

[40] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-DoF grasp generation in cluttered scenes," in *IEEE International Conference on Robotics and Automation*, 2021.

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

[42] K. Rahardja and A. Kosaka, "Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues," in *IEEE/RSJ International Conference on Intelligent Robots and Systems.*, 1996.

[43] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[44] L. Wang, Y. Xiang, and D. Fox, "Manipulation Trajectory Optimization with Online Grasp Synthesis and Selection," in *Robotics: Science and Systems*, 2020.

[45] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *The International Journal of Robotics Research*, 2019.

[46] J. Ortiz, A. Clegg, J. Dong, E. Sucar, D. Novotny, M. Zollhoefer, and M. Mukadam, "iSDF: Real-Time Neural Signed Distance Fields for Robot Perception," in *Robotics: Science and Systems*, 2022.

[47] D. Driess, J.-S. Ha, M. Toussaint, and R. Tedrake, "Learning models as functionals of signed-distance fields for manipulation planning," in *Conference on Robot Learning*, 2022.

[48] M. Danielczuk, A. Mousavian, C. Eppner, and D. Fox, "Object rearrangement using learned implicit collision functions," in *IEEE International Conference on Robotics and Automation*, 2021.

[49] P. Liu, K. Zhang, D. Tateo, S. Jauhri, J. Peters, and G. Chalvatzaki, "Regularized deep signed distance fields for reactive motion genera-

tion," *IEEE International Conference on Intelligent Robots and Systems*, 2022.

[50] B. D. Ziebart, A. L. Maas, J. A. Bagnell, A. K. Dey *et al.*, "Maximum entropy inverse reinforcement learning." in *Association for the Advancement of Artificial Intelligence*, 2008.

[51] A. Boularias, J. Kober, and J. Peters, "Relative entropy inverse reinforcement learning," in *International Conference on Artificial Intelligence and Statistics*, 2011.

[52] M. Kalakrishnan, P. Pastor, L. Righetti, and S. Schaal, "Learning objective functions for manipulation," in *IEEE International Conference on Robotics and Automation*, 2013.

[53] 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.

[54] 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.

[55] X. Yan, J. Hsu, M. Khansari, Y. Bai, A. Pathak, A. Gupta, J. Davidson, and H. Lee, "Learning 6-DoF grasping interaction via deep geometry-aware 3d representations," in *IEEE International Conference on Robotics and Automation*, 2018.

[56] J. Hager, R. Bauer, M. Toussaint, and J. Mainprice, "Graspme-grasp manifold estimator," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication*, 2021.

[57] A. Dragan, G. J. Gordon, and S. Srinivasa, "Learning from experience in manipulation planning: Setting the right goals," *Robotics Research*, 2017.

[58] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *The International Journal of Robotics Research*, 2011.

[59] N. Vahrenkamp, M. Do, T. Asfour, and R. Dillmann, "Integrated grasp and motion planning," in *IEEE International Conference on Robotics and Automation*. IEEE, 2010.

[60] J. Fontanals, B.-A. Dang-Vu, O. Porges, J. Rosell, and M. A. Roa, "Integrated grasp and motion planning using independent contact regions," in *IEEE-RAS International Conference on Humanoid Robots*, 2014.

[61] S. M. LaValle *et al.*, "Rapidly-exploring random trees: A new tool for path planning," *The annual research report*, 1998.