Learning to Plan with Pointcloud Affordances for General-Purpose Dexterous Manipulation

Anthony Simeonov[1], Yilun Du[1], Beomjoon Kim[1], Francois R. Hogan[2], Pulkit Agrawal[1], and Alberto Rodriguez[1]

***Abstract*—We aim to enable a robot to solve arbitrary manipulation tasks using dexterous manipulation primitives. This suggests the use of techniques from task-and-motion planning, which can sequence primitives by performing multistep reasoning and forward simulation. At the same time, we aim to solve such tasks when presented with objects of different shapes and sizes, suggesting the use of learned perceptual representation which enable generalization across geometries. However, it has remained challenging to incorporate these representations into systems that perform accurate forward simulation, a necessary component for reasoning toward long-horizon goals. We propose a framework, that utilizes deep generative models and segmented object pointclouds, that enables multistep planning using dexterous primitive manipulation skills in tasks involving a variety of object shapes and sizes. Our learned models, taking the form of biased sampling distributions, provide gains in planning efficiency over a manually designed baseline when integrated with a sampling-based planner. We also contribute a set of novel design choices in this framework which provide benefits in generalization and sample quality.**

I. Introduction

Consider the task depicted in Figure 1. A two-arm robot must use its palms to move the red box from its initial configuration to the green goal pose, which is on the opposite side of the table and on a different face of the object. This can be imagined as a proxy for the real-world task of moving, for example, a book from an arbitrary initial tabletop pose into a specified upright pose in the corner of a shelf.

We assume that the robot has access to a variety of parameterized primitive skill behaviors that can be combined to solve the task, such as pulling, to translate and change the yaw angle of the object, and grasping, to flip it onto a different face. Parameter values for each skill must be chosen that determine *how* the skill is executed. Finding a task solution amounts to searching for a suitable sequence of skill types and the values of the parameters of the skills (i.e. *where* to grasp and *how* to flip) for a particular start and goal pose of the object.

One way to tackle this problem is by using a task-and-motion planning (TAMP) algorithm [5, 2, 3, 10], which usually incorporate construction of a search tree, sampling from distributions over skill parameters, and iterative simulation of many skill sequences until one that reaches the goal is found. One of the major drawbacks of most of these TAMP algorithms is that they assume access to the pose and shape of the objects in the scene, and use manually-designed samplers that use this information for skill parameters.

[1]Massachusetts Institute of Technology. [2]Samsung Electronics. Correspondence to: Anthony Simeonov <asimeono@mit.edu>

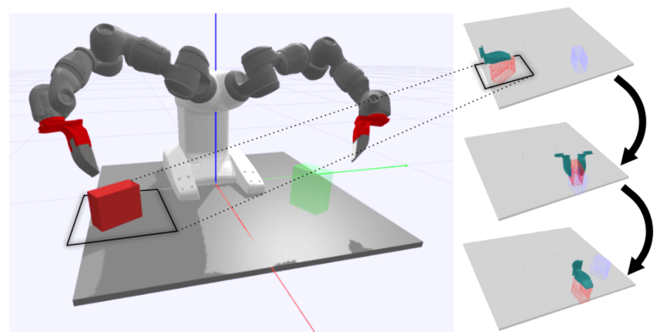


Figure 1. Our framework uses conditional generative models and dexterous primitives to imagine and execute multi-step manipulation plans for solving arbitrary start/goal manipulation tasks, when presented with only a pointcloud observation of the object.

For robots to operate in unforeseen environments however, they must be able to manipulate objects whose shape and pose are not known apriori. For example, consider again the problem in Fig. 1. The robot may need to put away many different sized books, but not know in advance their specific size and shape. This requires a sampler that can directly operate on sensory input, which is difficult to hand-design.

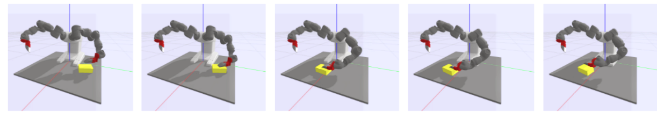
Based on this observation, we propose a framework that learns skill parameter samplers from prior experience using the primitive skills [6], that directly operate on sensory observations. While our framework can be used within any TAMP algorithm, in this work, we assume that a *plan skeleton* [7], which is a sequence of primitive types, such as pull-grasp-push, is given. Our objective is to find the continuous parameters of the plan skeleton using the learned samplers.

One key challenge in designing such a framework is learning the representation of an object represented by unstructured sensor data that enables both long-horizon reasoning and generalization across object shapes. There has been much work on learning representations of RGB-D images that are tailored to distinct action types, such as top-down grasping [13, 12, 14], suction [13] and pushing [1, 12]. These learning-based techniques for encoding sensory observations demonstrate strong generalization to diverse real world objects and are thus applicable for our similar goals of object generalization.

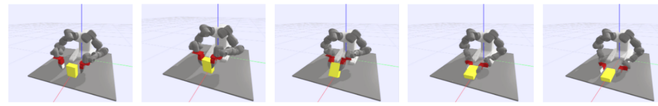
However, it has remained difficult to integrate these representations into systems that can solve problems involving longer planning horizons, in part because directly learning highly accurate forward models in this space of sensory data is extremely difficult. We instead propose to use segmented pointclouds as an object representation that provides more structure than pixel-based representations but is still flexible enough to afford generalization across object shapes.

To support long-horizon planning, we propose a system design where we learn two different samplers: a subgoal sampler for predicting a reachable rigid body transform that can be used to forward simulate and imagine future pointcloud observations, and a contact sampler that generates end-effector poses suitable for achieving the predicted subgoal. The samplers are designed to learn and exploit the correlation between contacts and subgoals. To support generalization across objects, we use recent advancements in neural network architectures that can operate on pointcloud data, such as PointNet++ [8] and Graph Attention Networks (GAT) [11], in a conditional generative modeling scheme where the samplers are represented as neural networks.

We validate our approach in a simulated domain where a dual arm system, equipped with end effector palms, is tasked with solving a large distribution of object manipulation tasks. Our method provides significant advantages in planning efficiency and prediction quality over a manually designed baseline that utilizes privileged knowledge about the shape of the objects and the task.



(a)



(b)

Figure 2. Snapshots during the execution of a (a) pull skill and (b) grasp skill on the simulated YuMi robot in PyBullet.

II. Problem Definition

A. Problem Setup

Throughout this work, we define a primitive *skill* π as a function that takes as an input a set of parameters, Θ, and outputs a robot joint trajectory. Specifically, we consider primitive which take as parameters an initial world frame pose of one or both robot palms $T^p_{R,L} \in SE(3)$ and a desired transformation $T^o \in SE(3)$ to apply to the object,

$$\pi : T^p_R \times T^p_L \times T^o \longrightarrow \mathcal{Q}^*_{R,L}$$

where $\mathcal{Q}^*_{R,L}$ denotes the space of sequence of configurations of left and right arms for the primitive motion. In particular, we utilize the primitive planning scheme developed in [4], although other types of primitives which are expressed based on an initial contact configuration and a desired motion to apply are equally applicable to the presented framework.

B. Planning Problem Definition

Assume we are given a desired rigid body transformation to be applied to the object, $T^o_{des} \in SE(3)$ along with a *plan skeleton* PS that defines the high level sequence of primitives that should be used to complete the task:

$$PS = \pi_0(\Theta_0) \longrightarrow \pi_1(\Theta_1) \longrightarrow \cdots \longrightarrow \pi_K(\Theta_K)$$

For plan skeleton PS, we denote its space of parameters as Θ_{PS}. The robot observes the world using a set of RGB-D sensors which provide a segmented point cloud observation of the environment $X \in \mathbb{R}^{N\times 3}$. Given a set of primitive skills, a plan skeleton PS, a desired object transformation T^o_{des}, and a pointcloud observation X of the object at some initial configuration, our objective is to find the parameters of the given plan skeleton, $\theta_{PS} \in \Theta_{PS}$, that achieves T^o_{des}.

C. Learning Problem Definition

Given a distribution from which skill parameters can be sampled, an RRT-style algorithm (described in detail in Section III-D) can be used to solve the planning problem described in Section II-B. This involves sampling intermediate values of T^o, referred to as *subgoals*, and corresponding end effector poses T^p that can reach these subgoals. We are given an experience dataset D consisting of successful single-step skill executions. These can be considered as a set of demonstrations of solving planning problems with PS of length *one*, where the subgoal parameter T^o equals T^o_{des}.

$$D = \{(X^{(i)}, PS^{(i)}, T^{o\ (i)}_{des}, \theta^{(i)}_{PS})\}_{i=1}^{n}$$

Using this data, we aim to train *conditional generative skill models*, $p_{\phi,\textsc{Skill}}(\cdot|X)$, taking the form of a deep neural network parameterized by ϕ, that at test time produces a distribution over high likelihood skill parameters when provided with a new pointcloud observation.

III. Methodology

A. Learning Approach

We leverage conditional variational auto-encoders (CVAE) [9] to learn $p_{\phi,\textsc{Skill}}$ using dataset D. This approach raises multiple open questions in the design of the system architecture and skill parameter representation, and these questions are discussed in the following sections.

B. Learning the joint subgoal-contact distribution

The most straightforward application of this framework to our problem of sampling T^p and T^o is to train separate CVAEs, effectively modeling T^o and T^p as independent random variables. Another option is to model their dependence via a conditional distribution, i.e. learn one CVAE that predicts subgoals conditioned on pointclouds, and learn a second CVAE that predicts contact configurations conditioned on pointclouds *and* subgoals. Instead, our approach is to directly try to model the *joint* distribution via a single latent variable, i.e. *simultaneously* predict both subgoals and contact configurations conditioned on pointclouds.

C. Generalizable $SE(3)$ Subgoal Representation

Some skills such as $\pi_{\text{GraspReorient}}$ operate in full $SE(3)$. For the purposes of generalization to novel geometries and novel object states, it is unclear whether directly predicting an $SE(3)$ transformation is the most useful approach. Our apporach is to instead have the model predict a binary segmentation mask directly with respect to the observed pointcloud, representing the set of points that *should end in contact with the table*, and then use these points in a downstream registration routine to solve for T^o.

D. Generalizable Multistep Planning

We integrate our learned generative models with a sampling-based planning framework inspired by RRT. For the first step in PS, the initial pointcloud X_0 is used to sample parameters $\theta_0 \sim p_{\phi,\text{Skill}}(\cdot|X_0)$ for that corresponding skill. If the instantiated skill $\pi_0(\theta_0)$ is feasible, θ_0 are saved for that position in PS. Additionally, the initial pointcloud X_0 is transformed via the sampled T^o_0 into a new configuration X_1 that can be used as the conditioning variable when sampling the *next* skill in the skeleton, and this process is repeated. For the final step in PS, the required unknown T^o_K can be solved for based on the required transformation-to-go to solve the task as $T^o_K = T^o_{des}\Pi_{i=0}^{K-1}T^o_i$, since we know the determined returned subgoals $T^o_0, ..., T^o_K$ should follow $T^o_{des} = \Pi_{i=0}^{K}T^o_{K-i}$.

IV. Experiments and Results

Our experiments are designed to validate our primary claim that our learned sampler performs better than a hand-designed sampler when integrated with our multistep planning framework. We have also conducted ablation studies to confirm that our choice of joint subgoal-contact distribution modeling and segmentation mask subgoal representation lead to better generated sample quality and generalization to unseen pointclouds, but these results are left separate from this paper. In the current work we focus on the primitive $\pi_{\text{GraspReorient}}$ (Fig. 2b).

A. Experimental Setup

The framework is implemented on an ABB YuMi robot simulated in PyBullet. GelSlim end effectors are used, similarly to as in [4], without the tactile sensing simulated. The robot operates in a table-top environment with RGB-D cameras located at the four corners all pointing at the same focal point in the front of the robot. We utilize the built-in segmentation mask capability in PyBullet to segment the object from the environment to obtain pointclouds of the object from the simulated depth images.

B. Training Details and Network Architecture

We compare both PointNet++ [8] and GAT [11] for the encoder and decoder in our CVAE. The encoder is trained to map the data in D to a latent conditional distribution, constrained by a KL-divergence loss to resemble a unit Gaussian. Sampled latents are concatenated as an additional feature to the points in X, which the decoder uses to reconstruct the data. We generate a distribution of cuboids of varying dimensions. The model is trained on a small subset of them and tested on cuboids with dimensions never seen during training. The data generation procedure relies on the simulation of the primitive skills in a scenario where 3D object models are available, allowing computation of stable poses, collision checking between the robot and the environment, and using a rejection sampling scheme to randomly sample skill parameters to find ones that are feasible. When feasible skills are found, they are executed and the parameters are added to the dataset.

Table I
Multistep Planning Results

Planning Success Rate	PG	GP	PGP
Learned	0.78	0.83	0.79
Uniform	0.19	0.19	0.05
Sticking Contact Success Rate	PG	GP	PGP
Learned	0.79	0.93	0.74
Uniform	0.75	0.71	0.50
Mean Planning Time (s)	PG	GP	PGP
Learned	96	27	87
Uniform	72	90	68

Multistep planning with a 5-minute timeout using our learned samplers and a manually designed uniform sampler. Planning success rate: fraction of problems where the planner found a solution before timing out. Sticking contact success rate: fraction of plans that were executed where contact was maintained with the object for the duration of each step. Mean planning time: average planning time in seconds for finding a feasible plan, not counting time-outs. `PG: Pull → Grasp`, `GP: Grasp → Pull`, `PGP: Pull → Grasp → Pull`

C. Evaluation: Multi-step Manipulation

To validate the benefits provided by our learned sampler, we conduct multi-step planning experiments using unseen objects and measure planning success rate and returned plan quality for a variety of plan skeletons. The sample-based framework described in III-D is implemented using $p_{\phi,\text{GraspReorient}}$ and a set of planning problems are set up using 20 unseen cuboids. We compare to a baseline uniform sampler that uses the pointcloud to sample potential values for $\theta_{\text{GraspReorient}}$ based on the prior knowledge that the objects are cuboids. A similar heuristic sampler for π_{PullRH} (Fig. 2a) is used in both cases.

Planning success rate is used to quantify the planning efficiency afforded by the two sampling schemes. We additionally report the fraction of plans where the robot maintains contact throughout open loop plan execution to quantify the plan quality, along with mean planning times. The results in Table I, indicate that the learned sampler provides a substantial benefit in planning efficiency and plan quality.

V. Conclusion

We presented a method based on deep generative modeling applied to segmented pointclouds to enable general-purpose multistep sampling-based planning using dexterous primitive manipulation skills. Our method enables planning efficiency gains over a manually designed baseline sampler, while simultaneously allowing generalization to unseen objects by utilizing a perception-driven object representation.

References

[1] Pulkit Agrawal, Ashvin V Nair, Pieter Abbeel, Jitendra Malik, and Sergey Levine. Learning to poke by poking: Experiential learning of intuitive physics. In *Advances in neural information processing systems*, pages 5074–5082, 2016.

[2] S. Cambon, R. Alami, and F. Gravot. A hybrid approach to intricate motion, manipulation, and task planning. *International Journal of Robotics Research*, 2009.

[3] C. R. Garrett, T. Lozano-Peréz, and L P. Kaelbling. Ffrob: Leveraging symbolic planning for efficient task and motion planning. *International Journal of Robotics Research*, 2014.

[4] Francois R. Hogan, Jose Ballester, Siyuan Dong, and Alberto Rodriguez. Tactile dexterity: Manipulation primitives with tactile feedback, 2020.

[5] Leslie Pack Kaelbling and Tomas Lozano-Perez. Hierarchical task and motion planning in the now. In *IEEE Conference on Robotics and Automation (ICRA)*, 2011. URL http://people.csail.mit.edu/lpk/papers/hpnICRA11Final.pdf. Finalist, Best Manipulation Paper Award.

[6] Beomjoon Kim, Leslie Pack Kaelbling, and Tomas Lozano-Perez. Guiding search in continuous state-action spaces by learning an action sampler from off-target search experience. In *Proceedings of the 32th AAAI Conference on Artificial Intelligence (AAAI). To appear*. AAAI Press, 2018. URL http://lis.csail.mit.edu/pubs/kim-aaai18.pdf.

[7] Tomás Lozano-Pérez and Leslie Pack Kaelbling. A constraint-based method for solving sequential manipulation planning problems. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3684–3691. IEEE, 2014.

[8] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[9] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In *Advances in neural information processing systems*, pages 3483–3491, 2015.

[10] Marc Toussaint. Logic-geometric programming: An optimization-based approach to combined task and motion planning. *International Joint Conference on Artificial Intelligence*, 2015.

[11] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[12] Andy Zeng, Shuran Song, Stefan Welker, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Learning synergies between pushing and grasping with self-supervised deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4238–4245. IEEE, 2018.

[13] Andy Zeng, Shuran Song, Kuan-Ting Yu, Elliott Donlon, Francois R Hogan, Maria Bauza, Daolin Ma, Orion Taylor, Melody Liu, Eudald Romo, et al. Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 1–8. IEEE, 2018.

[14] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. Tossingbot: Learning to throw arbitrary objects with residual physics. *arXiv preprint arXiv:1903.11239*, 2019.