ABC PROBLEM: AN INVESTIGATION OF OFFLINE RL FOR VISION-BASED DYNAMIC MANIPULATION

Seyed Kamyar Seyed Ghasemipour*, **Igor Mordatch & Shixiang Shane Gu**
Google Research
Mountain View, CA, USA
{kamyar,imordatch,shanegu}@google.com

ABSTRACT

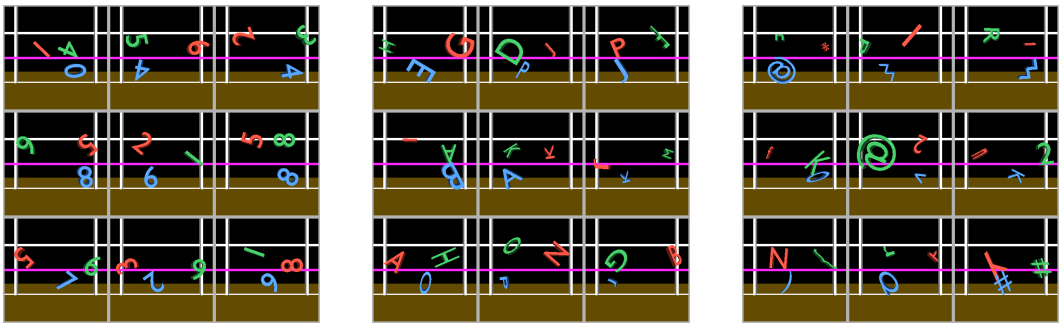
In recent years, reinforcement learning has had significant successes in the domain of robotics. However, most such successes have either been in robotic locomotion domains or robotic manipulation settings such as pick-and-place, block-stacking, and other quasi-static tasks, both of which lack a requirement of fine-grained geometric reasoning. In this work, we ask the following question: Given current established methodologies in RL, can we obtain effective *vision-based* policies that solve tasks requiring significant geometric reasoning, and how well do such policies *generalize*? As a secondary question, we pursue our investigations under the offline/batch RL setting. We study these questions in a simplified simulated rendition of the "ABC Problem" proposed by Prof. Tenenbaum. In the ABC problem, in each episode an agent is initialized with two random objects to use as its hands (A and B), and the objective is to lift a third randomly selected object (C) from the ground. Due to the varying geometries of the sampled objects, a trained agent must learn to reason about the most effective procedure for lifting the objects. Our empirical results demonstrate that indeed, by training on a limited subset of available objects, vision-based policies obtained through offline RL can significantly improve upon the policies generating the offline datasets, and can transfer to a diversity of objects outside the training distribution. Additionally, we demonstrate that learned policies exhibit novel characteristics not seen in the offline datasets, and we provide evidence that points towards investing efforts in attention architectures for vision-based control policies. Videos can be found in supplementary materials.

1 INTRODUCTION

From learning strong vision-based pick-and-place of diverse objects (Kalashnikov et al., 2018) to sample-efficient learning of robust locomotion policies (Song et al., 2020), reinforcement learning (RL) has resulted in many major advances in the field of robotics. In this work seek to understand how well current established methodologies can be applied to robotics manipulation tasks characterized by vision-based observations requiring highly dynamic policies and geometric reasoning. Additionally, we seek to evaluate how well such policies *generalize* to unseen geometries. Inspired by the "ABC Problem" proposed by Prof. Tenenbaum at RSS 2020, we design a simplified simulated environment that enables us to study the above questions.

In the ABC problem, in each episode an agent is initialized with two random objects to use as its hands (A and B), and the objective is to lift a third randomly selected object (C) from the ground. Due to the varying geometries of the sampled objects, an agent must be able to reason about object geometries in order to execute an effective procedure for lifting the objects. In contrast to most typically studied quasi-static robotic manipulation domains in the reinforcement learning literature (e.g. block-stacking, Meta-World (Yu et al., 2020), etc.), the ABC challenge requires more agile policies and significantly more fine-grained geometric reasoning. Indeed, while tasks such as opening door and stacking blocks can be formulated to use low-level state information, in the ABC problem it is a *necessity* to use higher dimensional structured input such as vision.

*Work done while intern at Google



(a) Training Tasks (b) Alphabet Subset Tasks (c) Diverse Characters Tasks

Figure 1: Samples of training and different evaluation tasks

2 SETUP

In it's original form, the ABC problem introduces a number of orthogonal complexities to our main questions of interest (geometric reasoning and generalization ability). Such complexities include, partial observability due to occlusions in 3D physics, the need for teaching a robot to grasp the A and B objects before using them for manipulation, and a very challenging RL exploration landscape[1]. In this section we describe our setup which mitigates such additional complexities.

2.1 ENVIRONMENT & TASK SETUP

Figure 1 presents visualizations of our training and evaluation task setups. In each image, the green object is object A, the red object is B, and the blue object is the one to pick up, C. In order to easily generate a wide diversity of geometries, we decided to use numbers and characters as to obtain a wide distribution of meshes. To remove partial observability, we restrict all objects to only move and rotate along the $x - z$ axis. And to remove the need for learning grasping policies, agents are enabled to directly actuate objects A and B. The white bars represent the boundaries of the working space (i.e. the center of the objects cannot move outside this region), and the magenta bar represents the height above which the C object must be lifted to be considered a successful lift.

The training tasks – the tasks on which the offline dataset (discussed below) is generated and the agents are trained on – use meshes sampled from numbers 0-9 *at a fixed scale* (Figure 1a). The Alphabet Subset evaluation tasks sample meshes from capital letters A-P that are *randomly scaled* (Figure 1b), and the Diverse Characters evaluation tasks sample meshes from the entire alphabet (captial and lower case) as well as numbers and characters such as @,#,$,%,&,(,),{,},[,],* that are all *randomly scaled* (Figure 1c). This enables us to evaluate generalization at varying levels of difficulty. Our environments are implemented in Mujoco(Todorov et al., 2012), and we plan to release these domains for open-source usage.

2.2 OVERCOMING EXPLORATION WITH OFFLINE RL

To overcome the challenge of exploration, we perform our investigations under the offline RL setting (Levine et al., 2020). We use MPPI (Williams et al., 2017) – a model-based control method which uses access to the simulator – and reward-shaping to generate offline datasets. In short, at a given state, MPPI proceeds by sampling N sets of action sequences $\{(a_0^i, ..., a_t^i)\}_{i=1}^N$ from an initial distribution, executing each sequence from the current state, computing their respective returns, and using the returns to re-weight the samples and obtain a new sampling distribution. This process is repeated for a number of iterations. With simple reward shaping[2], MPPI can solve the problem with non-neglible success rate. After finding some initial reasonable parameters for MPPI, we only mod-

[1]By smoother exploration landscapes, we mean that simpler heuristics such as random actions and boltzmann exploration can result in sufficient exploratory behavior

[2]The shaped reward is the negative distance of A and B from C plus the height of object C

Setting	Numbers (Train Domain)		Scaled Alphabet Subset		Scaled Diverse Characters	
Data MPPI 16	0.51	-17.03	—		—	
Data MPPI 128	0.81	12.88	—		—	
BC MPPI 16	0.08	-35.76	0.03	-37.29	0.02	-39.91
BC MPPI 128	0.24	-31.25	0.10	-35.57	0.06	-38.95
BC MPPI 16+128	0.11	-34.92	0.05	-36.87	0.03	-39.62
CQL MPPI 16	0.76	-3.45	0.64	-13.85	0.52	-22.90
CQL MPPI 128	0.72	-9.92	0.55	-18.13	0.43	-26.56
CQL MPPI 16+128	0.71	-9.35	0.59	-17.89	0.49	-25.32
CQL MPPI 16 (No-Crop)	0.12	-36.15	0.12	-35.98	0.08	-38.85
CQL MPPI 16 (Bin Rews)	0.76	-3.32	0.65	-13.21	0.50	-23.36

Table 1: Evaluation results: To evaluate each setting, we trained policies with 2 random seeds for each hyperparameter setting, chose the best hyperparameter, and evaluated the 2 corresponding policies on 2000 sampled tasks. We are reporting the average success rate and average return value for each setting (returns computed under the "C height reward"). BC: Behavioral Cloning, No-Crop: Setting where image observation is uncropped, Bin Rews: Setting where reward is 0-1 for whether the height of object C is above the magenta line.

ified N for obtaining offline datasets of varying quality. We plan to release our generated datasets for open-source usage.

2.3 Method

As our RL training algorithm, we use Conservative Q-Learning (CQL) (Kumar et al., 2020), a state-of-the-art method for offline RL. To handle vision-based inputs, we use a small Resnet model(He et al., 2016) that is shared across the policy and the Q-functions. The policy and Q-functions each apply separate additional convolution and fully connected layers on top of the Resnet output. Some additional low-level state information such as position and rotation of the objects are also provided to the models. We have validated that by solely using this low-level information the tasks are unsolvable by CQL. In our experience, training policies using only image information and no low-level state information results in similar performance as when using the low-level state. In all experiments, the images had dimension 64x64 and the images from past 3 frames were stacked across the channel axis to allow the extraction of temporal information such as velocity.

3 Experiments

3.1 Choice of Reward Function and Type of Image Observation

We begin by evaluating important choices of reward function and the type of image observation used. We evaluate these choices on an offline dataset of 1 million timesteps generated using MPPI with 16 samples ($N = 16$). For rewards, we compare using the height of object C as the reward, versus, using a 0-1 indicator reward for whether object C is above the magenta line. Table 1 shows that the choice between these two rewards results in no noticeable difference (CQL MPPI 16 vs. CQL MPPI 16 (Bin Rews)).

For the choice of image observations, intuitively we can see that most regions of the images are uninformative to the task. For this reason we were interested in whether incorporating attention(Vaswani et al., 2017) would be valuable. To avoid the computational overhead of using attention architectures for the visual domain and for a more direct comparison to the baseline, as a proxy, we decided to crop the image observations to a tight bounding box around the three objects. Table 1 demonstrates the very significant effect that this choice has (CQL MPPI 16 vs. CQL MPPI 16 (No-Crop))[3]. *This finding provides significant motivation for investing in attention-based architectures or other effective proxies towards vision-based control.*

[3]In experiments not included in this work, we also trained with non-cropped image observations of size 128x128, however cropped observations of size 64x64 resulted in better policies.

Method	First 30 Steps	Last 30 Steps	Full Trajectory
CQL on MPPI 16 Data	12.46 ± 6.68	10.39 ± 7.69	34.98 ± 19.40
MPPI 16 Data	3.96 ± 3.07	4.95 ± 4.12	10.86 ± 8.08
MPPI 128 Data	5.56 ± 4.86	4.79 ± 3.93	10.57 ± 7.98

Table 2: Average amount of rotation action done by hand A in different time-slices of the trajectory. For a given trajectory, the amount of rotation is calculated as $|\sum_{t_0}^{t_1} \texttt{rot}(A)|$, where (t_0, t_1) mark the beginning and end of the timeslice and $\texttt{rot}(A)$ is the amount of rotation action. Since the rotation can be positive of negative, this evaluates the consistency of rotation in a particular direction.

Moving forward, for all experiments we use the height of object C as the reward, and cropped images for the observations.

3.2 Generalization

In this section we evaluate the generalization abilities of the learned policies. We train policies under 3 types of offline datasets: 1) 1 million timesteps of MPPI with $N = 16$, 2) 1 million timesteps of MPPI with $N = 128$, and 3) the combination of the prior two datasets. Our empirical results are shown in Table 1.

The evaluation results on the training domain demonstrate that while technically MPPI 128 is a significantly better policy than MPPI 16, perhaps paradoxically, training offline RL policies with MPPI results in better policies (CQL MPPI 16 vs. CQL MPPI 128). Evaluating on the unseen meshes, first we observe that trained policies *transfer intriguing well to very different meshes of very different scales*. Second, we observe that MPPI 16 trained policies generalize better as well and that the gap between them and the MPPI 128 trained policies widens.

We believe that this gap is due to MPPI being applied in a noise-free simulated domain, and due to its access to an exact model; as the number of samples in MPPI (N) increases, it can exploit the simulation better and exhibit bespoke behaviors for the meshes at hand and the specific situation. As a result, these behaviors are harder to learn and do not transfer well to new meshes.

3.3 Emergent Behavior

Visualizing the MPPI trajectories, we observe the policies exhibiting a general "hug-and-lift" behavior, particularly when $N = 16$. We were interested in whether the offline RL regime had learned novel types of behavior from this data. Visualizing the learned policies – reliably across random seeds – we observe the following behavior: 1) objects A and B move to either side of object C, 2) they both then rotate in opposite directions until object C lifts from the ground by a small amount, 3) they then push into C and lift it in the air, 4) once in the air, they rotate until object C is in a more stable position in their "grip" and they try to maintain it in the air.

To provide quantitative results for this qualitative behavior, we log the average amount of rotation action done by object A in the first 30 timesteps, last 30 timesteps, and the full trajectory. Table 2 clearly supports our qualitative observation. Interestingly, this also demonstrates that although MPPI 128 has a similar (but a bit higher) success rate than CQL with MPPI 16 data, they are accomplishing this success rate through very different means.

4 Discussion

In this work, we sought to study the efficacy of RL for vision-based manipulation problems requiring geometric reasoning. To this end we designed a simplified rendition of the "ABC Problem". Our empirical investigations demonstrated that using offline RL we were able to obtain policies that exhibited novel behaviors and generalized to a wide range of unseen objects. Additionally, our results presented a significant motivation for investing in attention architectures for visual manipulation. However, a number of questions remain unanswered. First, although it is clear that geometric reasoning is required to solve our tasks, it is less clear the extent to which they test this ability, since at least in simulation, a general hug-and-lift behavior can be very successful. How can we improve our task setup? Second, analyzing the videos, we see that obtained policies are not very good at maintaining object in the air. Where does this problem stem from, hardness of the tasks or limit of the offline data? Lastly, how can other modalities such as touch be levergaged for geometric reasoning? For example, could they help guide visual attention?

REFERENCES

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, et al. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *arXiv preprint arXiv:1806.10293*, 2018.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *arXiv preprint arXiv:2006.04779*, 2020.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Xingyou Song, Yuxiang Yang, Krzysztof Choromanski, Ken Caluwaerts, Wenbo Gao, Chelsea Finn, and Jie Tan. Rapidly adaptable legged robots via evolutionary meta-learning. *arXiv preprint arXiv:2003.01239*, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020.