Visuomotor Mechanical Search: Learning to Retrieve Target Objects in Clutter

Andrey Kurenkov*[1], Joseph Taglic*[1], Rohun Kulkarni[1], Marcus Dominguez-Kuhne[2],
Animesh Garg[3,4], Roberto Martín-Martín[1], Silvio Savarese[1]

***Abstract*— When searching for objects in cluttered environments, it is often necessary to perform complex interactions in order to move occluding objects out of the way and fully reveal the object of interest and make it graspable. Due to the complexity of the physics involved and the lack of accurate models of the clutter, planning and controlling precise predefined interactions with accurate outcome is extremely hard, when not impossible. In problems where accurate (forward) models are lacking, Deep Reinforcement Learning (RL) has shown to be a viable solution to map observations (e.g. images) to good interactions in the form of close-loop visuomotor policies. However, Deep RL is sample inefficient and fails when applied directly to the problem of unoccluding objects based on images. In this work we present a novel Deep RL procedure that combines i) teacher-aided exploration, ii) a critic with privileged information, and iii) mid-level representations, resulting in sample efficient and effective learning for the problem of uncovering a target object occluded by a heap of unknown objects. Our experiments show that our approach trains faster and converges to more efficient uncovering solutions than baselines and ablations, and that our uncovering policies lead to an average improvement in the graspability of the target object, facilitating downstream retrieval applications.**

I. Introduction

It is challenging for a robot to retrieve a known target object from a pile of cluttered elements, even when the target is partially visible. The occluding objects make it or impossible to grasp the desired object, requiring the robot to interact first with the unknown clutter to improve the target's *graspability*. Such situations appear frequently in domains such as home robotics or even logistic centers, and is considered an instance of the Mechanical Search problem [1].

Previous approaches proposed carefully-coded heuristics [1, 2] or learned [3, 4] sequences of actions that try to discover and retrieve the desired object. In both cases, the problem is simplified by choosing the action space to be a set of linear pushes parameterized as a point on the clutter and a direction to push, and a retracting motion after each action. The simplified pushing strategy leads to longer execution times and undesired clutter motion due to the retraction motion. Further, the goal of the pushing motions is sometimes primarily to singulate the target object, not to uncover it from underneath the covering clutter. A more natural solution for uncovering is to use a closed-loop and continuous pushing strategy based on the current visual signal, allowing the policy to adapt to the unforeseeable reactions of the interactions with the clutter.

*These authors contributed equally.
[1]Stanford University, [2]Caltech, [3]University of Toronto & Vector Institute [4]Nvidia

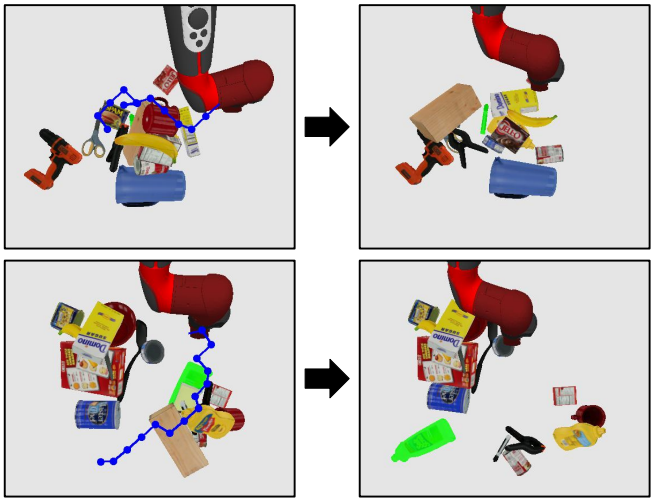


Fig. 1: An overview of the problem we address. On the left, the start state with the visible part of the target object highlighted green and the trajectory our policy took shown in blue. On the right, the resulting final state, with the target object now more visible and graspable.

In this work, we propose to address the problem of uncovering a partially visible target object to improve graspability by learning a visuomotor policy that maps the current image of the cluttered pile to continuous robot actions. We propose to use deep Reinforcement Learning (RL) to learn such a policy, given the recent successes of deep RL in image-based sequential decision making problems with unknown or complex environment dynamics [5–7]. However, existing RL approaches are data hungry and brittle: they require a large number of environment interactions to learn a mapping from high dimensional images to successful continuous robot actions, and often the algorithms fail to find a solution. A common strategy to avoid having to collect many interactions in the real world, which can be slow and dangerous, is to use a simulator. Nevertheless, for hard interactive problems, existing RL algorithms for continuous control may still struggle to learn a successful strategy in simulation. Therefore, in this work we present a deep RL solution combining in a novel manner three algorithmic strategies that allow our method to learn to uncover the target object based on images.

A first strategy to improve the efficiency of visuomotor learning in simulation is to leverage the information about the state of the environment from the simulator. Such *privileged information* is used only during training, while the component that maps inputs to actions at test time (the actor) is trained to use only images. While promising, this strategy alone is not sufficient to learn complex multi-object

continuous tasks such as pushing to uncover a target because of the large state space for exploration [8]

A second algorithmic strategy to improve RL training is to guide the exploration using teachers [9]. Teachers are expert policies that provide suggestions for suboptimal actions, which can guide the exploration of the RL agent to the relevant areas of the state space. While this helps with exploration, the challenge of learning a policy directly from images that can be used on a real robot is still significant.

For this last challenge, a third algorithmic strategy that has been shown demonstrated is to provide the agent with inputs in a mid-level representation instead of directly the raw RGB pixel inputs. Learning in the mid-level representation is more effective and facilitates transfer from simulation onto a real robot [6]. We make use of this concept by leveraging the segmentation mask of the target object, similarly to [3, 10], and the known extrinsics and intrinsics of the camera to get the position of these pixels relative to the end effector and provide those as input to the agent.

In summary, the main contributions in this work are:

1) a novel learning procedure that combines an asymmetric architecture to leverage privileged information, guidance from suboptimal teachers, and mid-level representations to train deep RL agents for visuomotor continuous control tasks,
2) the application and instantiation of this learning procedure to solve the problem of uncovering a target object to improve its graspability.

We conducted extensive experiments in simulation to evaluate the performance of our learned agents. The results indicate that our combination of privileged information, teacher guidance, and mid-level representation greatly improves sample efficiency and final performance. The method, applied to our Mechanical Search problem, learns to uncover a target object under clutter and improves its graspability.

II. Related Work

When operating in unstructured environments, robots often encounter cluttered environments that need to be interacted with. This may occur, for example, during sorting or retrieving a specific object. This problem has been tackled from different perspectives. Interactive perception approaches [11] have considered the perceptual problem of segmenting images of a pile of objects into coherent components that move together [12–15]. Several works in this area applied pushing actions to facilitate segmentation [2, 16, 17] but not to uncover a target object.

The grasping community considered the problem of planning and executing grasps on a pile of cluttered objects until all objects have been cleared. Depending on the assumed prior knowledge these methods can be considered model-based [18–20] or model-free approaches [1, 3, 4, 21–25]. The latter use only images to decide on the best grasping action for a pile of objects. A recent work [26] presented an instance of this problem to show how human demonstrations for such a task can be crowdsourced. Differently, our goal is not to clear a pile of objects, but to uncover and facilitate grasping of a given target object.

The two works that are the closest to ours are [1, 3]. Danielczuk et al. [1] defined the problem of Mechanical Search, searching for a known object among unknown cluttering objects with interactions, and proposed a method that chooses among discrete pre-specified action policies (e.g. pushing, grasping with suction, grasping with parallel-jaw gripper) based on the acquired RGB-D images. While their method includes a heuristically pre-specified action to push clutter, we go further in this work and use RL to optimize continuous visuomotor policies to more optimally push objects in the environment to uncover a known target. We consider our method complementary to the one by Danielczuk et al. [1]: our learned pushing policies could be integrated into their framework as a more robust action policy.

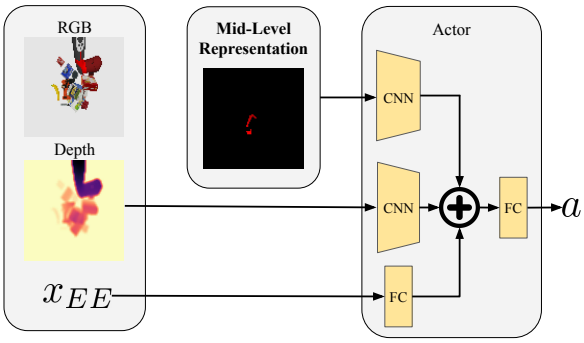
Yang et al. [3] proposed a Bayesian exploration policy to search for a target object with pushing actions. They pose the action selection problem as a Q-learning problem in the image domain. In contrast to their approach (and also to the one by Danielczuk et al. [1]), we aim to learn continuous controlled actions without retracting the arm after each discrete push, so that the manipulation is more reactive to the outcome of the interaction.

Learning continuous interaction with a complex environment using RL is a hard exploration problem. The exploration can be simplified using teachers, black box expert policies that can be queried during training time. This paradigm has been shown successful for manipulation [9]; in this work we show, for the first time, its applicability to high-dimensional inputs, i.e. images.

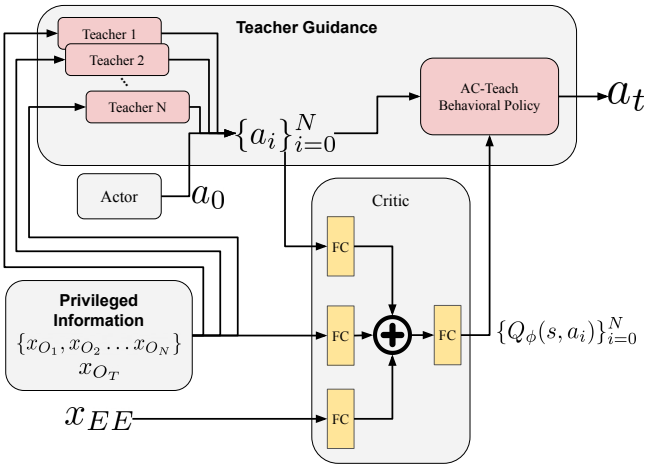
Although teacher guidance provides a better exploration strategy for the task, it does not address the additional challenge of learning from images. This can be alleviated using privileged information during training. We use privileged in our teachers an in the critic of our actor-critic solution, similar to Pinto et al. [8]. Additionally, recent work has studied how to use pretrained representations for robot learning [28–30], and in particular [3] showed how to use a pre-trained segmentation module to direct the policy towards the target object. We take inspiration from these methods and propose the use of a position image (similar to [27] use of a pixel coordinate channel), masked with the segmentation image, as an intermediate representation to facilitate policy learning.

III. Method

In the following, we first explain how we pose the problem of uncovering a known object with continuous pushing interactions based on visual images as a reinforcement learning problem. Then, we explain the novel policy architecture we proposed to solve this problem, followed by the special training procedure that leads us to a trained policy in simulation that we can transfer to the real robot.



(a) Actor of our actor-critic architecture using our mid-level representation. Inputs to our algorithm are RGB-D images, a segmentation image of the target object, and the pose of robot's end-effector. We transform the input into a mid-level representation consisting of a position image (each pixel has the values of the 3D offset to it from the end effector, similarly to [27] pixel coordinate channel) masked by the segmentation of the target object as in [3, 10]. The image in this representation, the depth image and the end-effector pose are featurized by separate network heads, concatenated and used to generate a pusing action.



(b) Training architecture leveraging teacher guidance and privileged information. Both the teachers and the critic of our actor-critic RL architecture leverage privileged information: the pose of all objects in the cluttered pile, including the target. The different teachers provide strategic pushing motions (straight, spiral, ...). The teachers and the critic with privileged information act only during training to reduce the samples necessary to train the actor.

Fig. 2: An overview of our method. x_{EE} denotes the end effect position of the robot, x_T denotes the position of the target object, $\{x_{O_1}, x_{O_2} \dots x_{O_N}\}$ denotes that set of the non target objects, and actions in the form of end effector position control commands are denoted with a_i. Our method is the result of combining the actor policy in (a) and the training procedure in (b).

A. Problem Statement

In our setup, a stationary robot is tasked with uncovering a known object of interest from the unknown occluding objects on top using pushing actions based on the images acquired by an RGB-D sensor. We pose this problem as a reinforcement learning problem on a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{Y}, \gamma, \rho)$. In the tuple, $\mathcal{S}$ is a continuous state space, $\mathcal{A}$ is a continuous action space, $\mathcal{Y}$ is the space of observations, $\mathcal{R}$ is a reward function $\mathcal{R}(s, a) = r \in \mathbb{R}$, $\gamma \in [0, 1)$ is a discount factor (for infinite horizon problems) and ρ is the initial state distribution. The goal is to learn a policy $\pi(a|s) = p(a|s)$ that selects actions based on current observations so as to maximize the expected reward [31].

The instantiation of our problem is depicted in Fig. 2. In our problem, the observations are RGB-D images and the position of the robot's end-effector in Cartesian space, and the actions of the agent are small end-effector changes in position (offsets relative to the current position).

Reward Function. The reward function that represents our task provides positive feedback when the object of interest becomes less occluded, and negative feedback when objects are moved. These penalties deter the agent from learning to push the entire pile to spread all objects, a strategy that can be dangerous and break the objects or the robot in the real world. Concretely, the reward function is a sum of the following terms:

1) **Target Uncovering Reward** of $2.5 * c$, where c is change in object visibility computed as $c = (occlusion_{t-1} - occlusion_t)/occlusion_{t-1}$. The target object to uncover is not static, but is rather a function of the state and changes every episode.
2) **Heap Movement Penalty** If $c < 0.05$, we penalize moving other objects needlessly. For each object, if its change in position is $m_o = \|(pos_t - pos_{t-1})\|$, this term is $75.0/N * m_o$. m_o is represented in meters, and therefore typically $m_o < 0.05$.
3) **Target Movement Penalty** If m_t is change in object visibility computed as $m_t = \|(target\_pos_t - target\_pos_{t-1})\|$, 75.0 * m_t. m_t is represented in meters, and therefore typically $m_t < 0.05$.
4) **Workspace Limits Penalty** A -0.5 reward is provided to encourage the agent to avoid the bounds of the workspace.
5) **Idleness Penalty** A -0.5 reward is provided to encourage the agent to seek positive rewards despite of the risk of other negative rewards.

An episode is considered finished when the target object is completely visible, or a specified limit of actions is reached.

B. Closed-Loop Visuomotor Control Policy

In addition to the complexity of the above reward function, this RL problem is challenging due to the difficulty of having to adapt to uncovering different target objects, due to having to explore a large state space, and due to having to learn from high dimensional visual inputs. We design our policy and training algorithm with a novel combination of features designed to address these challenges.

Though we could train our policy directly from the RGB-D observations of the scene, this would present the policy with the challenge of having to detect every kind of object it may have to uncover just from the signal of rewards. Therefore, instead of having the agent learn from RGB images, we

provide it with a mid-level representation that encodes the approximate position of the target object.

To make this representation, we first mask the pixels of the depth image to only those belonging to the target object, as in [3, 10]. Then, we use the known camera intrinsics and extrinsics to back-project the pixels into their 3D positions relative to the position of the robot's end effector. We base this second step on previous work that showed that CNNs are not well suited regress the coordinates of non-zero pixels in an image (something the agent will need to do to move towards the target object), and that explicitly providing the position of these pixels in the input is an effective solution [27]. The result is a 3 channel 'image', in which the 3 channels of the non zero pixels are the X, Y, and Z offsets from the end effector to visible pixels of the target object in meters.

For implementing this mid-level representation, we use the ground truth segmentation mask from rendering in the simulator. For eventual transfer to real world settings, it should also be possible to instead use a model trained for segmenting the objects of interest, such as the encoder-decoder segmentation network from previous work [32].

The deterministic actor $\pi_\theta(s)$ then takes this mid-level representation as input, as well as a depth image of the tabletop and heap of objects. Additionally, the actor accepts the position of the end effector as an input. Each image input is separately processed by a 3-layer convolutional neural net with the same structure as is used in [33]. The end effector position is passed through a fully connected layer of size 32 and ReLU activations. The outputs of the last convolution layers are then flattened and concatenated with each other and the end effector position output, and these features are processed by two fully connected layers of size 256. Lastly, a final fully connected layer with a tanh activation produces the action output scaled to the appropriate range.

C. Teacher-Guided Actor-Critic Policy Learning

By its nature as a continuous control problem, our task requires many actions to be executed to achieve the goal, which makes effective exploration challenging. For this challenge, we adopt the idea of agent exploration being guided by a set of provided black box policies (teachers) that suboptimally address part of the task and can suggest possible actions to take in any state. Early on in training the teachers' action suggestions are still expected to be superior to exploration only guided by noise added to the actor's output, which enables the agent to train faster while also optimizing for disturbing the heap of objects less than the teachers. Pushing is a particularly good fit for this approach, since it is easy to come up with several heuristic solutions that would be expected to be suboptimal but may be better than random. Specifically, we utilize the following teachers:

Straight Line Push: Executes a random straight line push to execute above the target object.

Zig Zag Push: Same as above, but the end effector moves in a zig-zag pattern while moving along the straight line.

Spiral Push: The end effector is placed near the target object, and the arm spirals out from that location for a specified amount of distance.

To have access to the target object's location as is necessary to implement these teachers, and to be able to compute the task's reward function, we perform training in the PyBullet simulator [34], with the renderer from the Gibson v2 simulator [35].

Because of our setting of continuous control and having access to multiple teachers, we base our approach on that of [9]. Thus, as in that work the agent to be trained is based on a probabilistic variant of the Deep Deterministic Policy Gradient (DDPG) algorithm [36], though other actor-critic algorithms could in principle be used. We make the critic probabilistic so that it can be used to select between the policy's chosen action or the actions output by hand-coded suboptimal teachers to the task for any given state. For more details including the exact formulations of the losses, refer to [9].

While teacher guidance helps lessen the exploration challenge, it does not alleviate the challenge of learning from high dimensional image inputs, and so is not enough to enable our method to learn effectively and efficiently. To deal with this last challenge, we again make use of the 'privileged information' afforded to us by training in a simulator by making the critic only depend on this low-dimensional information for its input, as in [8]. This privileged information is made up of two vectors, the position of the target object and a concatenation of the positions of all the other objects in the environment. All positions are provided relative to the end effector's position.

Both privileged information vectors and the end effector position are each processed with a separate fully connected network of size 32 and ReLU activations – unlike the actor network, there is no need to train convolutional neural nets for the critic. The outputs of these layers are concatenated, and then processed with two fully connected layers of size 256 and ReLU activations and a final fully connected layer that outputs the Q value estimate.

To summarize, on a given training step the policy's actor makes use of depth inputs and a mid-level representation to output its action, and our set of teachers also each output their respective actions. Then, the critic makes use of privileged information to evaluate each action so that the best one may be selected. Once an action is executed, its transition (s, a, s', r) is put in a replay buffer. Transitions from the replay buffer are intermittently sampled as training data for the actor and critic. The critic is trained via the Bellman residual loss $\mathcal{L}_{\text{critic}} = (r + \gamma Q_{\phi'}(s', \pi_{\theta'}(s')) - Q_\phi(s, a))^2$ and the actor is trained with a deterministic policy gradient update to choose actions that maximize the critic $\mathcal{L}_{\text{actor}} = -Q_\phi(s, \pi_\theta(s))$ where ϕ' and θ' denote the use of target critic and actor networks.

IV. Experiments

In our experiments, we aim to answer two main questions: First, we evaluate whether the proposed combination of

Fig. 3: Quantitative results in simulation for experiments with single-heap (top) and dual-heap (bottom) conditions showing evaluation results throughout training for the episode rewards (a), change in target visibility (b), number of steps taken until the episode nished (c), and change in target object graspability (d) respectively. Each plot represents values attained by the agent without added noise on held out evaluation heaps, plotted with respect to number of actions taken in the environment as part of exploration for training. The plots are limited to end on the step when the trained agent reaches its peak performance. Teacher evaluations are averages evaluated based on 100 rollouts. As shown by columns (a)-(c), our method consistently learns to improve these metrics and gets close to matching the teachers. And as shown in column (d), our method attains improvement in graspability that are close to on par with the teachers, despite not having access to the priviliged information they rely on.

asymmetric, teacher-guided visuomotor learning with a mid-level representation achieves better sample ef ciency and task performance in terms of rewards than an approach not utilizing these ideas. And second, we evaluate if the proposed approach is bene cial for continuous visuomotor control to uncover known target objects and increase their graspability. Concretely, we will answer the following questions:

1) Does our method results in policies that consistently uncover the object and increase the expected graspability?
2) How quickly objects are uncovered, how much are objects moved as a result, and what is the trade off between time to uncover the object and amount by which objects are moved?

To answer these questions, we perform a series of tests in simulation, with comparisons to several ablation versions of our solution.

A. Simulation Experiments

Heap Generation: We create two sets of experimental conditions: single-heap and dual-heap. The former has just a single heap of objects near the center of the workspace, while the latter has two heaps at some distance from each other. Intuitively, the latter condition better tests that the policy pays attention to the target object and not just the closest heap of objects.

For the single-heap setting, we programmatically generate 3000 heaps composed of 5, 10, and 15 objects. For the dual-heap setting, we generate 1200 heaps having 5,6,7,8,9, and 10 objects per heap. All heaps are made up of objects from the YCB object set [37]. To generate a heap, the Bullet Physics Engine [34] is used to simulate dropping random distinct objects onto the table workspace from a xed height. After all objects come to rest (their velocity nears zero), the modal (accounting for occlusions) and amodal (disregarding occlusions) segmentation masks of each object are used to check the degree to which they are occluded. The rst object with a valid amount of occlusion is made to be the target object of that heap. If no object is least 10% and no more than 90% occluded then we discard this heap and do not keep it in the nal set. These heaps are kept constant throughout our experiments, so that there is a fair comparison between our approach and baselines.

Policy Evaluation: Once heaps are generated, they are used to train and evaluate the deep RL agent. Training is done on either single-heap or dual-heap settings, with random sampling of a heap each episode, and with half of the heaps being held out during training to be used during evaluation. Evaluation is done after every N interaction steps with the environment by running the agent's policy (without additive noise or teacher guidance) on an evaluation heap. The agent is allowed to move its end effector at most 5cm in a given step. In both training and evaluation, the agent begins each episode in a random position at a minimum distance of 0.2 meters from the target object, and is given at most 50 actions to complete the task. To evaluate whether the trained agent can make objects more graspable and not just more visible, we measure the change in mean grasp quality score as measured by the fully convolutional grasp quality network [38] of the best 10 grasps for the image at the beginning and the end of the episode. We report results

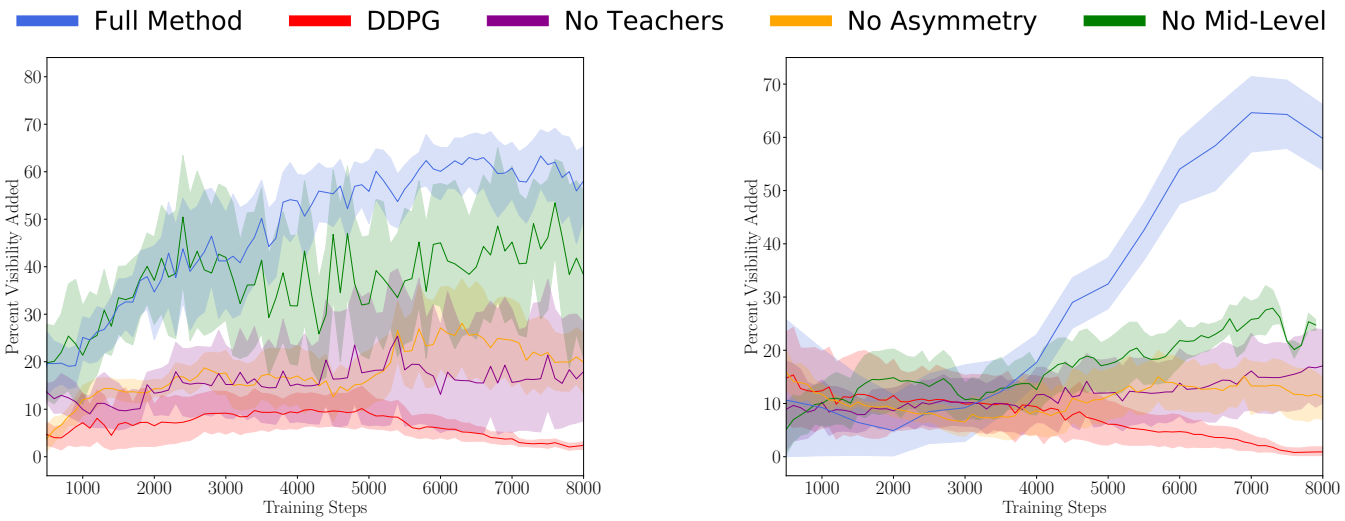


Fig. 4: Ablation results, for (left) single heap condition, and (right) dual heap condition. For the 'No Pose Input' ablation, we provide the agent with an RGB image and the label of the object instead of the approximate target position input. In the single heap setting, the agent can often perform well without the position input by just de-occluding the objects in the heap, but in the two-heap setting this is more important. In both conditions, teacher guidance and asymmetric learning contribute to effective learning.

using 5 seeds in all conditions.

B. Simulation Results

As shown in figure 3, our method is able to converge to high improvements in graspability within only 8,000 action executions in the environment in both the single-heap and the dual heap conditions, with the latter being more challenging. Furthermore, this level of performance corresponds to attaining higher environment rewards, the object being significantly more uncovered, and the object uncovered in less time. Lastly, our method does so with less disturbance to the object heap than any of the teachers, as measured by the mean distance that the non target objects in the heap move.

We compare our method to the baseline DDPG algorithm without teacher guidance or asymmetry but with the same inputs as our agent, and find that it does not learn to improve any of these metrics at all within the same time. We restrict our comparison to just the DDPG algorithm so the baseline RL algorithm is the same as in our method, since it could be implemented using newer state of the art algorithms as well. While results can be expected to be better if using superior algorithms, we report results with DDPG due to ease of implementation and results already being positive.

As shown in figure 4, the components of asymmetry and teacher guidance are both essential to the method being able to learn this efficiently and effectively. While training with RGB input works on par with the alternative of having the intermediate in the single-heap setting, that is not the case in the dual-heap setting, showing that the intermediate representation of where the target object is matters when there is more uncertainty about it.

C. Qualitative Results

As shown in figure 1, our policy learns to execute complex continuous control that is different to the behavior of the teachers. The policy learns a behavior to approach the object and "nudge" the occluding clutter gently, rather than executing a continuous push like the teachers do. We hypothesize this is an effect of the negative reward for moving the non-target object and the lack of ground truth about object poses used by our teachers with privileged information. The agent learns to uncover the target object moving the occluding objects that occlude it as little as possible.

The agents do not achieve full uncovering of the target object and improvement of graspability in all cases due to several failure modes. In some cases, the agent repeats the same actions back-and-forth without causing any change in the environment, as depicted in Fig. 5, top row. In other cases the agent moves near the target object but does not interact with the objects occluding it, as shown in Fig. 5, bottom row. These failure modes can likely be addressed by modifying the agent's architecture, which we leave to future work.

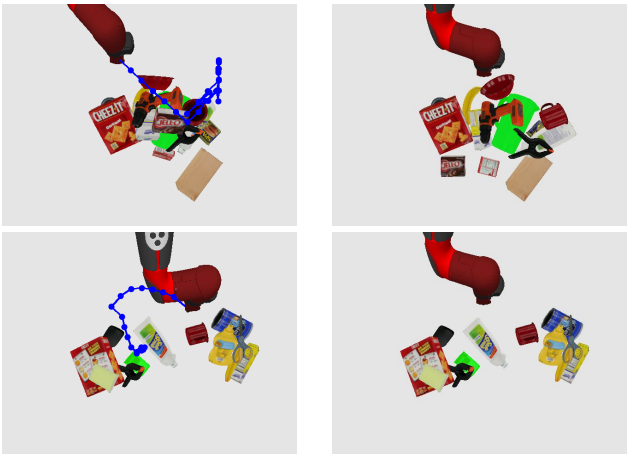


Fig. 5: Visualizations of failed trajectories from given start states (left) to the end state (right) for two rollouts in simulation. Segmentation mask overlay is shown in green and full trajectory is shown as blue lines on the left column images.

V. Conclusion

We presented a novel approach combining teacher guidance, asymmetric learning, and a mid-level representation to learn pushing strategies that uncover a known object of interest in cluttered piles of objects. Our learned policies demonstrate behaviors adapted from the teacher demonstrations to the lack of privileged information in the deployment conditions. Our solution achieved positive results, uncovering the target and improving its graspability in most experiment. We plan to work on the failure cases, improving the interaction strategies with better suggested motions from teachers.

Acknowledgement

We acknowledge the support of Toyota (1186781-31-UDARO). AG is supported in part by CIFAR AI chair. We thank our colleagues and collaborators who provided helpful feedback, code, and suggestions, especially Professor Ken Goldberg, Michael Danielczuk, Matt Matl, and Ashwin Balakrishna.

References

[1] M. Danielczuk, A. Kurenkov, A. Balakrishna, M. Matl, D. Wang, R. Martín-Martín, A. Garg, S. Savarese, and K. Goldberg, "Mechanical search: Multi-step retrieval of a target object occluded by clutter", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 1614–1621.

[2] T. Hermans, J. M. Rehg, and A. Bobick, "Guided pushing for object singulation", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 4783–4790.

[3] Y. Yang, H. Liang, and C. Choi, "A deep learning approach to grasping the invisible", *IEEE Robotics and Automation Letters*, 2020.

[4] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. 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)*, IEEE, 2018, pp. 4238–4245.

[5] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, "Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation", *ArXiv preprint arXiv:1806.10293*, 2018.

[6] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, "Gibson Env: Real-world perception for embodied agents", in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, IEEE, 2018.

[7] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots", *ArXiv preprint arXiv:1804.10332*, 2018.

[8] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning", *ArXiv preprint arXiv:1710.06542*, 2017.

[9] A. Kurenkov, A. Mandlekar, R. Martín-Martín, S. Savarese, and A. Garg, "Ac-teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers", *ArXiv preprint arXiv:1909.04121*, 2019.

[10] T. Kim, Y. Park, Y. Park, and I. H. Suh, "Acceleration of actor-critic deep reinforcement learning for visual grasping in clutter by state representation learning based on disentanglement of a raw input image", *ArXiv preprint arXiv:2002.11903*, 2020.

[11] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action", *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[12] R. Martín-Martín, S. Höfer, and O. Brock, "An integrated approach to visual perception of articulated objects", in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 5091–5097.

[13] H. Van Hoof, O. Kroemer, H. B. Amor, and J. Peters, "Maximally informative interaction learning for scene exploration", in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2012, pp. 5152–5158.

[14] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments", in *2009 IEEE International Conference on Robotics and Automation*, IEEE, 2009, pp. 1377–1382.

[15] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation", *Autonomous Robots*, vol. 37, no. 4, pp. 369–382, 2014.

[16] M. Gupta, J. Müller, and G. S. Sukhatme, "Using manipulation primitives for object sorting in cluttered environments", *IEEE transactions on Automation Science and Engineering*, vol. 12, no. 2, pp. 608–614, 2014.

[17] A. Eitel, N. Hauff, and W. Burgard, "Learning to singulate objects using a push proposal network", in *Robotics Research*, Springer, 2020, pp. 405–419.

[18] D. Berenson and S. S. Srinivasa, "Grasp synthesis in cluttered environments for dexterous hands", in *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, IEEE, 2008, pp. 189–196.

[19] M. Moll, L. Kavraki, J. Rosell, *et al.*, "Randomized physics-based motion planning for grasping in cluttered and uncertain environments", *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 712–719, 2017.

[20] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato, "Online planning for target object search in clutter under partial observability", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8241–8247.

[21] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer", in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, pp. 639–646.

[22] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences", in *Conference on robot learning*, 2017, pp. 515–524.

[23] T. Novkovic, R. Pautrat, F. Furrer, M. Breyer, R. Siegwart, and J. Nieto, "Object finding in cluttered scenes using interactive perception", *ArXiv preprint arXiv:1911.07482*, 2019.

[24] L. Berscheid, P. Meißner, and T. Kröger, "Robot learning of shifting objects for grasping in cluttered environments", *ArXiv preprint arXiv:1907.11035*, 2019.

[25] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping", *ArXiv preprint arXiv:1707.01932*, 2017.

[26] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei, "Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity", in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019.

[27] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the coordconv solution", in *Advances in Neural Information Processing Systems*, 2018, pp. 9605–9616.

[28] M. A. Lee, Y. Zhu, K. Srinivasan, P. Shah, S. Savarese, L. Fei-Fei, A. Garg, and J. Bohg, "Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks", in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 8943–8950.

[29] A. Raffin, A. Hill, K. R. Traoré, T. Lesort, N. Díaz-Rodríguez, and D. Filliat, "Decoupling feature extraction from policy learning: Assessing benefits of state representation learning in goal based robotics", *ArXiv preprint arXiv:1901.08651*, 2019.

[30] A. Sax, B. Emi, A. R. Zamir, L. Guibas, S. Savarese, and J. Malik, "Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies", *ArXiv preprint arXiv:1812.11971*, 2018.

[31] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[32] C. Wang, D. Xu, Y. Zhu, R. Martín-Martín, C. Lu, L. Fei-Fei, and S. Savarese, "Densefusion: 6d object pose estimation by iterative dense fusion", in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3343–3352.

[33] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning", *ArXiv preprint arXiv:1509.02971*, 2015.

[34] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation, games, robotics and machine learning*, `http://pybullet.org/`, 2017.

[35] F. Xia, C. Li, K. Chen, W. B. Shen, R. Martín-Martín, N. Hirose, A. R. Zamir, L. Fei-Fei, and S. Savarese, "Gibson env v2: Embodied simulation environments for interactive navigation", Stanford University, Tech. Rep., Jun. 2019.

[36] P. Henderson, T. Doan, R. Islam, and D. Meger, "Bayesian policy gradients via alpha divergence dropout inference", *ArXiv preprint arXiv:1712.02037*, 2017.

[37] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: The ycb object and model set and benchmarking protocols", *ArXiv preprint arXiv:1502.03143*, 2015.

[38] V. Satish, J. Mahler, and K. Goldberg, "On-policy dataset synthesis for learning robot grasping policies using fully convolutional deep networks", *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1357–1364, 2019.