Affordance Learning from Play for Sample-Efficient Policy Learning

Jessica Borja-Diaz*, Oier Mees*, Gabriel Kalweit,
Lukas Hermann, Joschka Boedecker, Wolfram Burgard
University of Freiburg

Abstract

Robots operating in human-centered environments need to understand how to interact with them: *what* can be done with each object, *where* this interaction may occur, and *how* the object is used to achieve a goal. To this end, we propose a novel approach that extracts a self-supervised visual affordance model from human teleoperated play data and leverages it to enable efficient policy learning and motion planning. We combine model-based planning with model-free deep reinforcement learning (RL) to learn grasping policies that favor the same object regions favored by people, while requiring minimal robot interactions with the environment. We evaluate our algorithm, Visual Affordance-guided Policy Optimization (VAPO), with both diverse simulation manipulation tasks and real world robot tidy-up experiments to demonstrate the effectiveness of our affordance-guided policies. Code, pretrained models and dataset are available at `http://vapo.cs.uni-freiburg.de`.

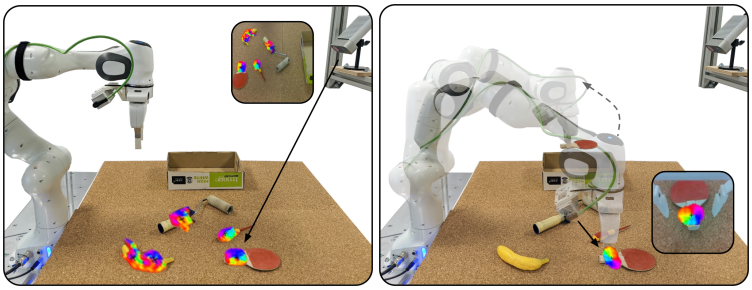


Figure 1: Real-world setup for a tidy up task: our self-supervised visual affordance model guides the robot to the vicinity of actionable regions in the environment with a model-based policy. Once inside this area, we switch to a local reinforcement learning policy, in which we embed our affordance model to favor the same object regions favored by people and to boost sample-efficiency.

1 Introduction

Humans have the ability to effortlessly recognize and infer functionalities of objects despite their large variation in appearance and shape. For example, we understand that we need to pull the handle of a drawer to open it or grasp a knife by the handle to use it. This capacity to focus on the most relevant behaviors in a given situation enables efficient decision making by limiting the choices of action that are even considered. Gibson's theory of affordances [1] provides a way to reason about object function. It suggests that objects have "action possibilities", e.g., a mug is "graspable" and a door is "openable"

*These authors contributed equally. Correspondence to `meeso@informatik.uni-freiburg.de`

NeurIPS 2021 Workshop on Robot Learning: Self-Supervised and Lifelong Learning, Virtual, Virtual

and has been extensively studied in both the robotics and the computer vision communities [2]. However, the abstract notion of "what actions are possible?" addressed by current affordance learning methods is limited. A robot needs to know *where* are actionable regions in an environment, the specific points on the object that need to be manipulated for a successful interaction, *what* it can achieve with it and *how* the object is used to achieve a goal. Current affordance learning methods have two major problems. First, they are limited by requiring heavy supervision in the form of manually annotated segmentation masks [3, 4, 5, 6] or expensive interactive exploration [7, 8], restricting their scalability and applicability in practical robotics scenarios. Second, current affordance-augmented robotic systems are limited in the complexity of the actions they model by relying often on predefined action templates [7, 8, 9, 10]. Together, these limitations naturally restrict the scope of affordance learning systems to a narrow set of objects and robotics applications.

In light of these issues, we propose a method for sample-efficient policy learning of complex manipulation tasks that is guided by a self-supervised visual affordance model. Therefore, we call our algorithm Visual Affordance-guided Policy Optimization (VAPO). Towards overcoming the issues of expensive manual supervision and exploration, we propose to learn affordances that are *grounded* in real human behavior from teleoperated "play" data [11]. Play data is not random, but rather structured by human knowledge of object affordances (e.g., if people see a drawer in a scene, they tend to open it). Moreover, affordances discovered from unlabeled play are *functional affordances*, priming a robot to approach an object the way a human would. We hence leverage this visual affordance model to guide a robot to perform complex manipulation tasks. Our approach decomposes object manipulation into a sample-efficient combination of model-based planning and model-free reinforcement learning, inspired by a recent line of work that aims to combine classical motion planning with machine learning [12, 13, 14]. Concretely, we first predict object affordances and drive the end-effector from free-space to the vicinity of the afforded region with a model-based method. Once inside this area, the model cannot be trusted and we switch to a reinforcement learning policy in which the agent is rewarded for interacting with the afforded regions. This way, the local policy has a "human prior" for how to approach an object, but is free to discover its exact grasping strategy. The contribution of our visual affordance model to boost sample-efficiency is two-fold: 1) driving the model-based planner to the vicinity of afforded regions, 2) guiding a local grasping RL policy to favor the same object regions favored by people.

2 Approach

Our approach consists of three steps. First, we train a network to discover and learn object affordances in unlabeled play data (Sec. 2.1). Second, we divide the space into regions where a model-based policy is reliable and regions where it may have limitations handling perception errors or physical interactions. We leverage the learned affordance model to drive the end-effector from free-space to the vicinity of the afforded region with a model-based policy π_{mod}. Third, once inside this area we switch to a local reinforcement learning policy π_{rl}, in which we embed our affordance model to favor the same object regions favored by people and to boost sample-efficiency (Sec. 2.2). Thus, our final policy is defined as a mixture $\pi(a|s) = (1 - \alpha(s)) \cdot \pi_{mod}(a|s) + \alpha(s) \cdot \pi_{rl}(a|s)$. We use an estimate of the distance between the robot's gripper and the affordance region $\alpha(s)$ to switch between the policies. An overview of the system is given in Figure 1.

2.1 Learning Visual Affordances from Play

We decouple the affordance prediction task into different components. First, the affordance model $\mathcal{F}_a$ learns to transform a grayscale image I into a binary segmentation map $F \in \mathbb{R}^{H \times W}$, indicating regions that afford an interaction. Second, similar to Xie *et al.* [15] it estimates 2D pixel coordinates of the affordance region centers by predicting a vector $V \in \mathbb{R}^{H \times W \times 2}$ from each affordance pixel towards the center. Estimating the center points of the afforded regions is a key component in order to disambiguate affordances from multiple objects in a scene. In order to discover affordances in unlabeled data we make use of the gripper action as a heuristic to detect human-object interactions, see 2. From the play data we get a tuple $O^t = (f^t_{cam}, p^t_{robot}, a^t_{gripper})$ for each timestep t. We denote f^t_{cam} as the current image frame, p^t_{robot} as the cartesian coordinates of the robot's end-effector and $a^t_{gripper}$ as a binary gripper open/close command. Intuitively, if the gripper closes during play, it is indicative of a possible interaction that will start at that position. Thus, we can project the gripper's 3D point p^t_{grip} to a camera image pixel u^t_{grip} and label the pixels within a radius r for

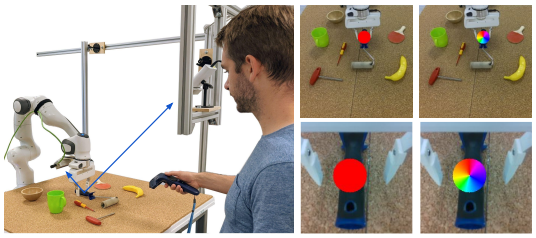


Figure 2: We leverage a self-supervised signal of a robot's gripper opening and closing during human teleoperation to project the 3D tool-center-point into the static and gripper cameras. We label the neighboring pixels within a radius around the afforded region with a binary segmentation mask and direction vectors from each pixel towards the affordance region center.

the past n frames as an afforded region. Similarly, if the gripper transitions from being closed to open, it means that we ended an interaction with an object at the 3D position p_i^t. As the frames are labeled, new interaction points are discovered until we get a set of points P_{fixed}^k, which represent the world coordinates of where interactions have ended until timestep k. Finally, we define the set of discovered object positions for a timestep k as $P_{active}^k = P_{fixed}^k \cup p_{grip}^t$ for $k = t-n, \cdots, t$. Each 3D point inside P_{active}^k is projected to a camera image pixel u_i^k to create the affordance mask label by marking neighboring pixels. The pixel coordinates of the projected points are used as the affordance region centers U^k. In order to disambiguate affordances from multiple objects in a scene, we let the network predict 2D pixel coordinates of the affordance region centers by predicting a vector $V \in \mathbb{R}^{H \times W \times 2}$ from each affordance pixel towards the center. To train the full affordance model F_a we apply two different loss functions on the affordance segmentation F and the direction prediction V, more training details are found in the Appendix.

2.2 Affordance-guided Reinforcement Learning Grasping

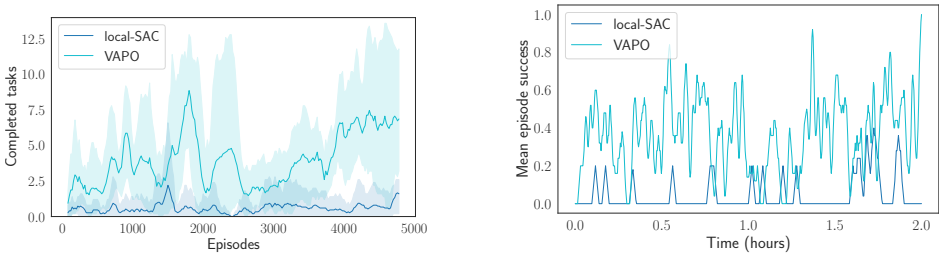
We consider the standard Markov decision process (MDP) $M = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \mu_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state space and action space respectively. $\mathcal{T}(s'|s,a)$ is the probability of transitioning from state s to state s' when applying action a. The actions are drawn from a probability distribution over actions $\pi(a|s)$ referred to as the agent's policy. $r(s,a)$ is the reward received by an agent for executing action a in state s, μ_0 the initial state distribution, and $\gamma \in (0,1)$ the discount factor. The observation space is composed of two parts: 1) the proprioceptive state including the 3D world coordinates of the end effector, the orientation euler angles and the gripper width. 2) The visual inputs consisting of the current RGB image observed by the gripper camera, the corresponding depth image and the binary affordance mask predicted from the gripper camera based affordance model. We use a 7-DOF Franka Emika Panda robot with a parallel gripper both in simulation and in the real-world. We leverage the visual affordance model to guide the agent to get close to the affordance centers by reward R_{aff}. Additionally if the agent goes outside the neighborhood, it receives a negative reward R_{out} and if it successfully lifts an object it receives a positive reward of R_{succ}, i.e. the immediate reward is defined as $r(s,a) = \lambda_1 R_{succ} + \lambda_2 R_{aff} + \lambda_3 R_{out}$.

3 Experimental Results

We evaluate our method with both diverse simulation manipulation tasks and real world robot tidy-up experiments. To test the sample efficiency of the affordance-guided RL policy, we compare against a sparse-reward SAC agent, **local-SAC**. Unlike our method, in which we make use of the gripper camera affordances to guide the RL policy, the baseline only takes an RGB-D image as input and its corresponding reward function is $r(s,a) = \lambda_1 R_{succ} + \lambda_3 R_{out}$. In both the baseline and our method, the model based policy moves the robot towards the vicinity of the afforded region.

3.1 Simulation Experiments

We evaluate two tasks in simulation: a grasping task and opening a drawer. The grasping task consists on lifting different objects in a PyBullet simulated environment. The policy is trained over 20 different objects with varying degrees of complexity, such as hammers, teapots, knifes and power drills, as



(a) Simulated pick-up task. Shaded regions depict the standard deviation.

(b) Real-world tidy-up task.

Figure 3: VAPO vs. local-SAC in (a) simulated pick-up tasks and (b) a real-world tidy-up task. Mean episode success over the last ten episodes is shown. Our method is able to successfully lift the most amount of objects as compared to the baseline.

shown in the Appendix. After the policy executes a close-gripper action, the gripper attempts to lift the object and wait in the air for two seconds. If the object is still in the gripper at the end of this time, we define the grasp as being successful. VAPO is not exclusive to a grasping task. To show this, we train a policy to open a drawer in a simulation as shown in Figure 5b. Every episode consists of the drawer on a closed position and the robot in a neutral position. To complete the task, the robot must open the drawer at least 15cm. To train the affordance models we teleoperate the robot using a virtual reality (VR) controller to collect unscripted play data. We gather around two hours of human interaction which amounts to 100K images for each environment to train the static camera and gripper camera affordance models. For the grasping task, we train all policies over the 20 objects using the same configuration. The learned policies are tested on grasping the set of all the training objects, the quantitative results are shown in Figure 3a. Our method is able to lift significantly more objects than the baselines as it has a strong prior on how objects should be interacted with. For the opening-drawer experiment we report a success rate of 84% for VAPO over 100 episodes and of 52% for the baseline. The results indicate that the affordances boost the performance and the sample efficiency of the reinforcement learning agent.

3.2 Real World Experiments.

For the real world experiment, we setup the environment using a seven DOF Franka Emika Panda robot. The full setup can be seen in Figure 1. Similar as in simulation, we collect play data by teleoperating the robot using a VR controller as shown in Figure 2. We accumulate around 1.5 hours of human interaction, which results in 70K images and use this to train both the gripper camera and static camera affordance models. We evaluate our approach on a real-world tidy-up experiment. We show the learning curves for this experiment in Figure 3b. We use a 7-DOF Franka Emika Panda robot and run our policy at 20 Hz. We train all methods to pickup four objects: a plastic banana, a screwdriver, a table tennis racket and a paint roller. After two hours of training VAPO is able to consistently "functionally" grasp all the objects, e.g., grasping the objects by the handles, while the SAC baseline very rarely achieves to lift any object, despite the agent starting at the same robot pose as our method. This is due to the low number of samples that sparse-reward SAC is trained on, since most success stories of RL in the real world require several orders of magnitude more data [16]. Overall, our results demonstrate the effectiveness of our approach to learn sample-efficient policies by leveraging self-supervised visual affordances.

4 Conclusion

In this paper, we introduced the novel approach VAPO (Visual Affordance-guided Policy Optimization) as a method for sample-efficient policy learning of manipulation tasks that is guided by a self-supervised visual affordance model. The key advantage of our formulation is the extraction of visual affordances from unlabeled human teleoperated play data to learn a strong prior about *where* actionable regions in an environment are. To the best of our knowledge, this work is the first one to demonstrate the effectiveness of visual affordances to guide model-based policies and closed-loop RL policies to learn robot manipulation tasks in the real-world.

References

[1] James J Gibson. The ecological approach to visual perception. *Boston: Houghton Mifflin*, 1979.

[2] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018.

[3] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Detecting object affordances with convolutional neural networks. In *IROS*, 2016.

[4] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *ICRA*, 2015.

[5] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IROS*, 2017.

[6] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *ICRA*, 2018.

[7] Kaichun Mo, Leonidas Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *ICCV*, 2021.

[8] Tushar Nagarajan and Kristen Grauman. Learning affordance landscapes for interaction exploration in 3d environments. In *NeurIPS*, 2020.

[9] 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 *ICRA*, 2018.

[10] Lin Yen-Chen, Andy Zeng, Shuran Song, Phillip Isola, and Tsung-Yi Lin. Learning to see before learning to act: Visual pre-training for manipulation. In *ICRA*, 2020.

[11] Corey Lynch, Mohi Khansari, Ted Xiao, Vikash Kumar, Jonathan Tompson, Sergey Levine, and Pierre Sermanet. Learning latent plans from play. In *CoRL*, 2020.

[12] Tom Silver, Kelsey Allen, Josh Tenenbaum, and Leslie Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

[13] Michelle A Lee, Carlos Florensa, Jonathan Tremblay, Nathan Ratliff, Animesh Garg, Fabio Ramos, and Dieter Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. In *ICRA*, 2020.

[14] Brian Ichter, Pierre Sermanet, and Corey Lynch. Broadly-exploring, local-policy trees for long-horizon task planning. *arXiv preprint arXiv:2010.06491*, 2020.

[15] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. Unseen object instance segmentation for robotic environments. *IEEE Transactions on Robotics (T-RO)*, 2021.

[16] 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. In *CoRL*, 2018.

[17] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *arXiv preprint arXiv:1505.04597*, 2015.

[18] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. In *RSS*, 2018.

[19] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

A Implementation Details

Teleoperated play data During the unscripted teleoperated interactions we record images from two cameras: a static camera that can capture the global scene, and a camera mounted on the robot's gripper. We label of the images of the static camera with a radius $r = 10$ of pixels around p, $p \in P_{active}$ and the the gripper camera images with $r = 25$.

Affordance model For the affordance segmentation loss F, we use a weighted sum between a cross entropy ℓ_{ce} and a dice loss ℓ_{dice} to account for class imbalance. For the direction prediction we optimize a weighted cosine similarity loss given by $\ell_{dir} = \sum_{i \in \mathcal{O}} \alpha_i (1 - V_i^{\mathbf{T}} \bar{V}_i) + \frac{\lambda_b}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \left(1 - V_i^{\mathbf{T}} \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right)$,

where $V_i, \bar{V}_i$ are the predicted and ground truth unit directions of pixel i respectively. $\mathcal{B}, \mathcal{O}$ are the sets of pixels belonging to the background and affordance region classes. The total loss for the affordance model is given by $w_{ce}\ell_{ce} + w_{dice}\ell_{dice} + w_{dir}\ell_{dir}$. We use a U-net [17] architecture followed by two parallel branches of convolutional layers that produce the affordance mask and center directions. Similar to Xiang *et al.* [18], we use a Hough voting layer to predict the 2D object centers during inference. The Hough voting layer takes the affordance mask and the direction vectors as input to compute a score for every pixel. This score indicates the likelihood for a pixel of being an affordance region center. Each pixel in the image casts a vote for image locations along the predicted direction from the network, then the object center is selected as the location with the maximum score.

We define a two-stage affordance detection by training separate models for the two cameras. One model is trained with images from a static camera and predicts a spatial interaction hotspots map, indicating actionable regions. Similarly, we train an affordance model with images from a gripper camera, which gives a finer-grained spatial interaction map about where humans tend to interact with each object. The images are converted to gray-scale before being fed to the networks. As the affordance models are intended to be used in the reinforcement learning policy, we follow the same pre-prossessing steps for obtaining the affordances during the policy learning.

Both affordance models are trained with stochastic gradient descent (SGD) using a fixed learning rate of 1e-5. We take a batch size of 128 and set the loss weights $w_{ce} = 1$, $w_{dice} = 5$, $w_{dir} = 2.5$. For the static camera we use an image resolution of $H = 200$, $W = 200$ and train it for 150 epochs. For the gripper camera we use an image resolution of $H = 64$, $W = 64$ and train it for 100 epochs.

Affordance-guided Reinforcement Learning Grasping

We train the policy using soft actor critic [19] and visualize our architecture in Figure 4. First we concatenate the RGB image to the corresponding depth image and affordance mask to use this as input for a convolutional neural network. The CNN is composed by three convolutional layers with kernel size [8,4,3] respectively and one linear layer to obtain a feature representation of size 16. Then we concatenate the obtained representation to the robot state and the distance to the affordance center. Finally this is passed through three fully connected layers. All the network layers use ReLU activations. The critic and actor are implemented following the same architecture but they do not share the network.

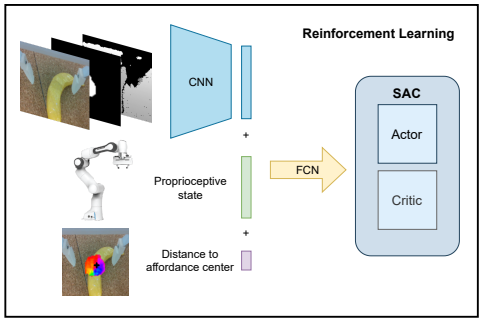


Figure 4: Architecture of the actor-critic networks. The inputs to the policy consist of the current RGB gripper image, its corresponding depth image and affordance binary mask as well as the robot state and the distance to the affordance center.

For the simulation experiments, we train a single policy for all the objects with an episode length of 100 steps during 600K episode steps. This amounts to around 20hrs of learning experience. We train for 3 seeds initializations. In the reward function we set $\lambda_1 = \lambda_2 = \lambda_3 = 1$ and the rewards $R_{succ} = 200$, $R_{out} = -1$. For the grasping simulation task we perform curriculum learning in order to train a single policy that can lift all 20 objects. Concretely, we divide the objects into classes and when the policy gets good at grasping an object, we replace it with a different object from the same class. Finally, we keep track of class specific success rates in order to bias the exploration towards hard to lift objects.

(a) Tidy-up task. (b) Drawer opening task.

Figure 5: Visualization of the simulated environments.