Enhancing Sample Efficiency through Affordance-Based Exploration

Samuel Li
Robotics Institute
Carnegie Mellon University United States
swli@andrew.cmu.edu

Yunchao Yao
Robotics Institute
Carnegie Mellon University United States
yunchaoy@andrew.cmu.edu

1 Motivation

Sample efficiency and generalization are persistent challenges in reinforcement learning and robotics. In particular, exploration in large state spaces from scratch is a poor approach, especially in settings of sparse reward. The robot can behave randomly without picking up any reward signals and get stuck in places with no value, or take prohibitively long to converge. Furthermore, in the real-world setting, exploration without constraints or strong initialization from high-level guidance can be unsafe and lead to inappropriate or undesirable behaviors. Random exploration of the environment is not how humans learn to interact with novel objects in new places. Based on an understanding of the functions that objects afford and how they should be interacted with, humans can efficiently and effectively identify interesting or useful objects [1, 2] to learn how they may behave or figure out how to manipulate them to achieve a goal. Our motivation is to bring this ability to robot learning by leveraging affordance understanding and open-world knowledge in large models, which can provide strong initialization for exploration and learning. Additionally, while existing segmentation models are capable of segmenting objects, they are not able to segment object parts, in particular articulated parts that possess the desired affordance in an object. For example, a faucet can only be turned on by manipulating its handle. We thus also explore how the coarse regions segmented from high-level affordance-based guidance can be further refined through online interaction to identify fine-grained useful object parts.

2 Prior Work

We mainly focused on two previous works. [1] learns an image-based affordance model using self-supervised labels from video of human interaction. This affordance representation can then be directly transferred to downstream tasks through various learning paradigms. While the learned affordance map is effective in guiding the agent towards locations with high rewards, learning the affordance requires a large number of demonstration videos. Since many tasks involve an object to be manipulated by the agent, directly guiding the robot to explore and learn the affordance map within proximity of the target object may increase the sample efficiency and will not need demonstrations.

7th Conference on Robot Learning (CoRL 2023), Atlanta, USA.

[2] maps human actions learned from internet videos to robot action representation and achieves zero-shot generalization on manipulation tasks such as opening cabinets. The effective mapping of human actions to robot action representations shows that robots can achieve task goals by behaviors similar to the ones of humans. This has inspired us to utilize recent visual language models, which are trained on a wide range of human-generated text and images, to produce objects of interest and reasonable action choices for the agent to execute.

3 Our Idea

3.1 Methods and Intuitions

Leveraging affordance understanding and broad internal knowledge in large models can increase the efficiency and performance of robot agents by guiding exploration and learning. Our idea is to use Visual-Language Models (VLM) for high-level planning. We focus on the goal-conditioned learning setting, where we provide the VLM with a goal image and the current scene image. Conditioned on these images, we rely on the VLM to identify what object in the scene should be interacted with to achieve the goal (as well as what action sequence out of a predefined set is appropriate). The VLM can output the desired object name that can be passed to a language-conditioned segmentation model to identify the object location in the camera frame as both a 2D mask and a bounding box. Querying the depth map uniquely identifies the third dimension, at which point a transformation can be applied to find the 3D object points in the world/robot frame. The robot acts in the end-effector space, so the transformed segmentation mask provides low-level control. However, this mask is still coarse; while VLM may often be able to identify fine-grained object parts, existing segmentation models fail to correctly segment beyond the object level. Many objects may only have a small region that affords the desired interaction. To address this, we learn a Gaussian policy that is initialized by sampling from the coarse mask. The policy initialized in this way is presumably already "close" to the desired region, and much more likely to sample high-reward actions than a policy that does not have this initialization and must start by sampling randomly from the entire space. Inspired by [3], we refine the Gaussian policy using the Cross-Entropy Method (CEM) for iterative policy updates. The pipeline is illustrated by 1

Intuitively, VLM can find the object in the scene that is most likely to be interesting to humans or manipulable to reach the goal. Every action sampled from the segmented object is more likely to give a high reward than from random sampling. We segment based on the object name text to locate the object in the camera frame, and through iterative learning using CEM we can refine the mask. The final Gaussian policy can be visualized as a heatmap and interpreted as the correct part with the desired affordance. The connection to prior work is that the policy is learned through initialization and constraints from human-driven affordance recognition. The VLM and segmentation pipeline can substitute the learned affordance models in prior works and is more generalizable to different scenes, tasks, perspectives, etc.

3.2 Implementation details

While in theory a library of action primitives can be generated beforehand for the VLM to choose, we create the action primitives produced by the VLM for our experiments. We implemented the "turn" primitive as a sequence of pre-defined actions that reaches a 3D location in the world frame and then performs a "v" shaped motion with the end-effector closed to turn the faucet. With this action primitive

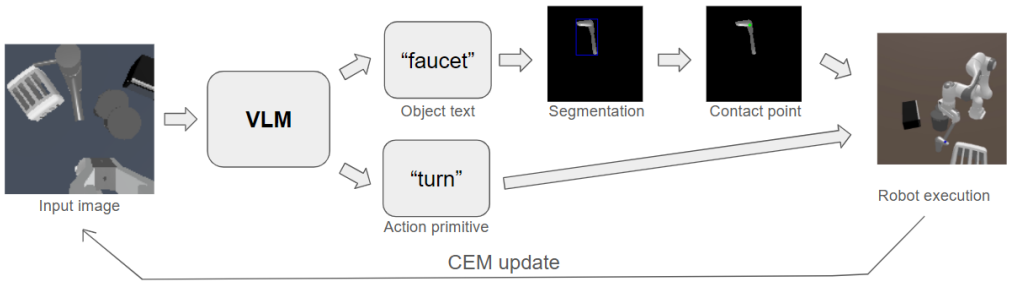

Figure 1: The overview of our pipeline. We utilize VLM to provide action primitive and target objects. Then off-the-shelf segmentation models (such as LangSAM) are used to segment the target object in the RGBD image frame. A Gaussian distribution is constructed from within the object segmentation mask to create locations with higher chances of creating meaningful explorations. The points are projected as 3D points using the depth information, and the agent refines the Gaussian with CEM updates guided by the reward of each interaction.

parameterized by its starting location, we utilize CEM with Gaussian distribution to refine the affordance map for the robot to perform the turn action. The affordance distributions are generated and learned in the 2D image frame, and the actual 3D locations for the robot to interact with are projected from the 2D locations using the camera parameters and depth information of RGBD observations.

We implemented sampling from the segmentation mask and sampling from the bounding box of the target object. A Gaussian distribution is constructed from a random location within the mask or bounding box and n locations are sampled for the agent to perform the action. We then pick m elites who have the highest episode returns and update the mean using the mean of elites and the variance using an exponential moving average:

$$\mu_i = \mu_e$$

$$S_i = (1-\alpha)S_{i-1} + \alpha S_e$$

where μ and S are the mean and variance, and α is the EMA coefficient, which we set to 0.9 in our experiments. Moreover, since the Gaussian distribution's initialization may cover an arbitrary portion of the target object, we also employ an exponentially decaying random exploration. For each iteration i of the T total iterations, the agent has a probability of $2\frac{T-i}{T^2}$ to randomly sample a location from within the segmentation mask or bounding box for exploration, while the rest of the samples are drawn according to the Gaussian. We found this approach helps the agent interact with the target object more uniformly in the beginning and stabilizes the policy learning, and the policy will eventually depend more on the learned Gaussian which ensures exploitation. Finally, we use the mean of the Gaussian as the evaluation policy.

4 Experiments

4.1 VLM extraction

We utilize GPT4-V [4] as the VLM, and ask it to extract key objects and action primitives on real-world images using the following prompt template:

Figure 2: Faucet initial and goal images used as visual prompts.

Prompt:
Given images of the current (first) and goal (second) state, tell me 1) which object in the scene a robot should interact with to achieve the goal and 2) which action to use from the set {"turn", "pick up", ...}. Be succinct, respond with (object name, action) and nothing else.
GPT-4V Response:
({Object}, {Action})

We test our prompt with images of a real-world kitchen sink 2 with the faucet being turned between the initial and goal frames. After applying the above prompt template, the VLM can successfully respond with the tuple (faucet, turn). We carry out the robot experiment using a simulated faucet-turning environment. Using the image observation returned by the camera mounted on the robot end-effector, we employ "faucet" as the query term for LangSAM [5] to acquire the mask and bounding box of the faucet. The action "turn" is then used to achieve the goal. Note that this pipeline only requires visual observation at the beginning of the episode.

4.2 Agent and Environment setup

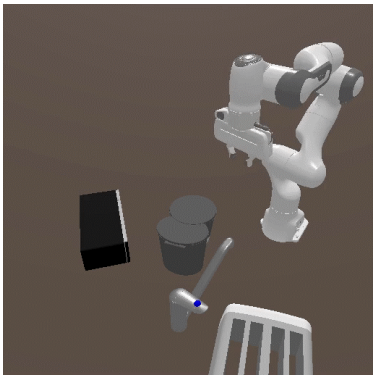

Figure 3: The modified "TurnFaucet" environment with cluttering props. The blue dot demonstrates the location for the agent to start the "turn" sequence of actions. This 3D location is acquired through projection from a 2D point in the image frame generated by the policy's affordance map

The ManiSkill2 benchmark [6] is a collection of manipulation tasks focusing on low-level control. We focus on the "TurnFaucet" environment, which requires a single Franka Emika Panda manipulator with a parallel gripper to turn on a water faucet. The environment also features a dense reward function that encourages 1) reducing the distance between the gripper and the Faucet, 2) increasing the angle of the faucet's handle joint, and 3) increasing the change in the handle joint's angle compared to the previous step. While our pipeline functions well with the original dense reward, we also experimented with a "sparse" reward by removing the proximity term in the reward function to better demonstrate the effectiveness of our pipeline; the proximity term likely provides information similar to our guided exploration method. Moreover, the original environment contains an isolated faucet, making the region for exploration very explicit. We modified the environment by adding random objects to simulate a cluttered scene, which is closer to the real-world setting that we run the visual language model on. Object variations are available for the environments to test the generalizability of our method. We evaluated three different models of faucets, and we applied random initialization to the locations of the faucets. An illustration of the modified environment is provided by 3

4.3 Results

We compared our methods against two baselines: 1) Random exploration, where the agent randomly picks a 3D location in the scene and conducts the turning action primitive, and 2) Constrained exploration, where the agent picks a location within a cube with a width of 2 unit in the x,y,z directions that contains the faucet. Our main result focuses on the "spare reward" setting with 3 faucet variances for clear comparison, but we also experimented with the "dense reward" setting on one faucet variance for fairness. All methods using sparse reward are trained for

20 iterations, and for each iteration, 20 locations are sampled for exploration while 10 are selected as elite for CEM updates. For the dense reward setting, 50 samples are drawn per iteration, while the elite and iteration numbers are the same as in the sparse setting. We report the average episode return of our method and the baselines across 3 random seeds, and we also visualize the progression of the Gaussian policy by visualizing the distribution as heat maps projected in the image frame.

As shown by 4, our method outperforms the baselines and converges faster. This confirms our intuition that explicitly guiding the exploration toward areas that are more likely to be interacted with by humans can improve the sample efficiency of the robot agent. On the dense setting 5, our method still converges quicker, but the random agent aided by the proximity return can also eventually achieve comparable episode returns.

5 Summary

To address the challenge of exploration and learning in large state and action spaces in robotics and reinforcement learning, we propose to use human-like affordance understanding and knowledge to guide agent exploration and interactions. We leverage the large-scale training and world knowledge of VLM and language-based open-world segmentation models to provide an "affordance mask" that the policy can sample and be initialized from. To refine the coarse mask and identify fine-grained parts with the desired affordances to robustly achieve goals, we update the affordance-initialized policy through online interactions. Our experiments show that affordance-guided exploration outperforms random exploration in both reward and efficiency. In particular, in the dense setting, where behavior is dominated by the signal coming from the distance reward, the guided exploration can start with a high reward and start interacting with the faucet handle almost immediately (within a couple of iterations at 20 samples each). The random policy moves toward the faucet using the distance reward but is unable to find the faucet handle within the limited experiment parameters. In the sparse setting where the reward comes only from the angle of the faucet handle, the random policy receives no signal through 20 iterations even scaling up to 50 samples/iteration, given the small likelihood of directly sampling a handle point, and thus receives no learning signal for positive updates. The learning rate decay further decreases exploration over time, forcing the agent to exploit an effectively random policy. Using the same experiment parameters, the affordance-guided agent can find a small region that allows it to effectively move the handle. This is only possible at this level of efficiency because the policy quickly starts sampling points from the handle when being constrained to the object mask.

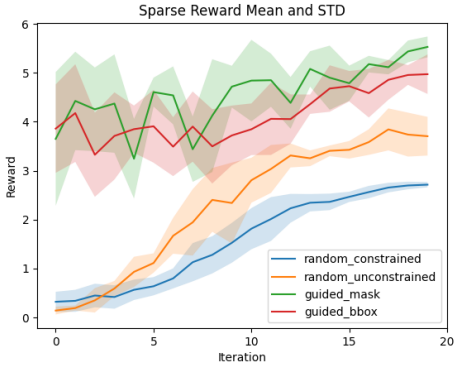

Figure 5: The episode returns under the dense reward setting. Our method still converges faster than the baseline methods, but the proximity term of the dense reward also allows the random agents to achieve similar performance as our method

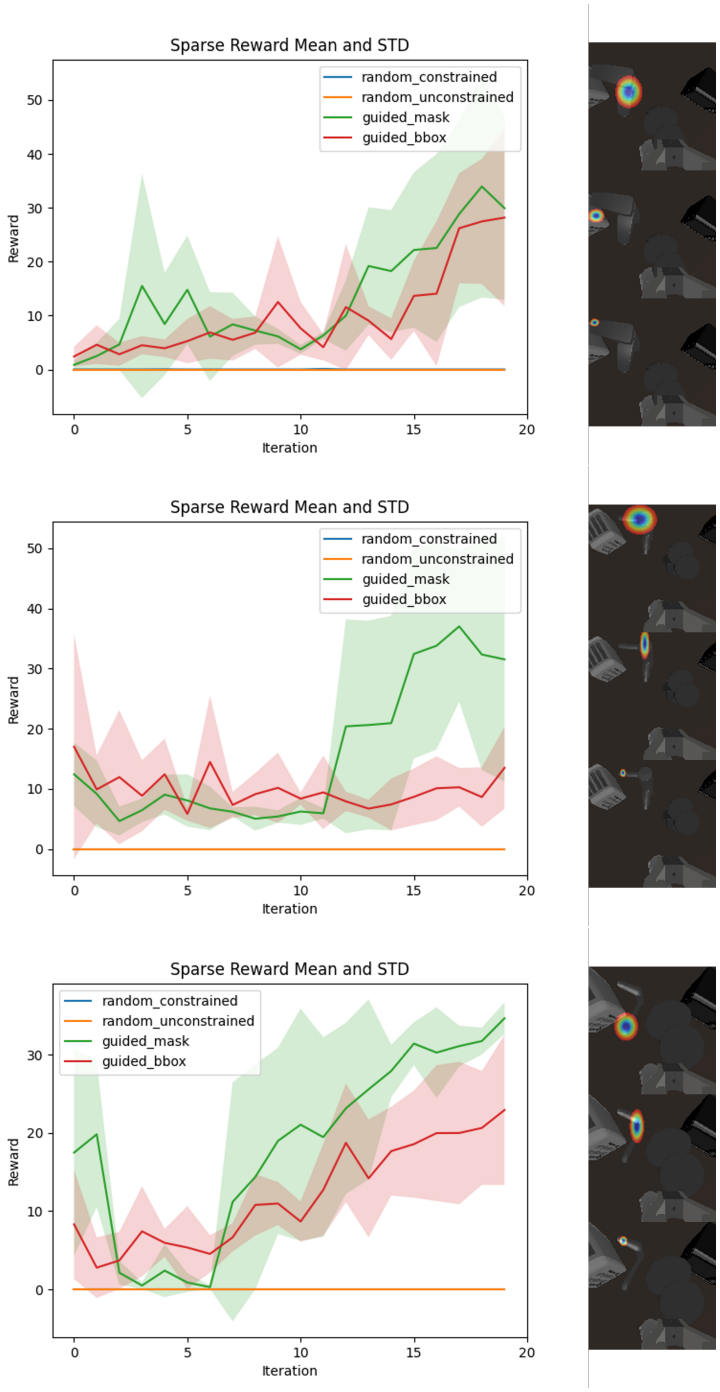

Figure 4: Plot of episode returns and progression of Gaussians on sparse reward setting. Our methods outperform the baselines on all variances of the environment. The heatmaps to the right of each plot show the Gaussian distribution that represents the affordance learned by the agent after 5,15,20 iterations (from top to bottom). The policy quickly converges to the handles of the faucets, leading to efficient interactions

6 Future Directions

One notable aspect of our method is that the learning (Gaussian policy updates) is agnostic to task, scene, etc. The pipeline is not trained for specific objects in

specific environments and thus is presumably generalizable to diverse settings. Our plan for future work is to set up different tasks and settings and demonstrate how the open-world capabilities of the affordance pipeline give us this desirable agnostic behavior. From there, we want to deploy the pipeline to learn in the real world to test its robustness to real-world variabilities. In addition, we also acknowledge that formulating an effective action primitive may not be trivial, and not all action primitives can be parameterized only by their starting positions alone, limiting our method to generalize across different tasks. While it is in theory possible to simplify the action primitives as a sequence of target end-effector poses, this formulation may require querying the VLM after every simulation step, drastically increasing computational cost and time of rollout. These limitations also opens up opportunities for future research.

References

[1] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak. Affordances from human videos as a versatile representation for robotics. 2023.

[2] H. Bharadhwaj, A. Gupta, S. Tulsiani, and V. Kumar. Zero-shot robot manipulation from passive human videos. *arXiv preprint arXiv:2302.02011*, 2023.

[3] A. Nagabandi, K. Konoglie, S. Levine, and V. Kumar. Deep Dynamics Models for Learning Dexterous Manipulation. In *Conference on Robot Learning (CoRL)*, 2019.

[4] OpenAI. Gpt-4 technical report, 2023.

[5] Lang-Segment-Anything Contributors. Language segment-anything. https://github.com/luca-medeiros/lang-segment-anything, 2023.

[6] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023.