UAD: Unsupervised Affordance Distillation for Generalization in Robotic Manipulation

Yihe Tang, Wenlong Huang, Yingke Wang, Chengshu Li, Roy Yuan,
Ruohan Zhang, Jiajun Wu, Li Fei-Fei
Stanford University

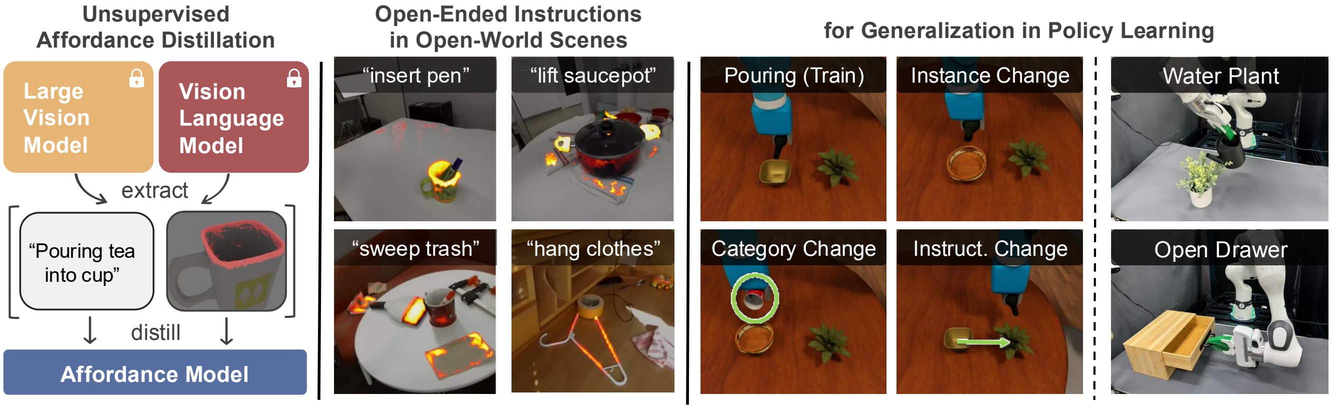


Fig. 1: **Unsupervised Affordance Distillation (UAD)** extracts affordance annotations from large pre-trained models and distills them into a task-conditioned affordance model, which is capable of predicting fine-grained affordance in open-world scenes with open-ended instructions, enabling diverse generalization properties in downstream policy learning.

***Abstract*— Understanding fine-grained object affordances is imperative for robots to manipulate objects in unstructured environments given open-ended task instructions. However, existing methods of visual affordance predictions often rely on manually annotated data or conditions only on a predefined set of tasks. We introduce Unsupervised Affordance Distillation (UAD), a method for distilling affordance knowledge from foundation models into a task-conditioned affordance model *without any manual annotations*. By leveraging the complementary strengths of large vision models and vision-language models, UAD automatically annotates a large-scale dataset with detailed <instruction, visual affordance> pairs. Training only a lightweight task-conditioned decoder atop frozen features, UAD exhibits notable generalization to in-the-wild robotic scenes and to various human activities, despite only being trained on rendered objects in simulation. Using affordance provided by UAD as the observation space, we show an imitation learning policy that demonstrates promising generalization to unseen object instances, object categories, and even variations in task instructions after training on as few as 10 demonstrations. Project website: unsup-affordance.github.io/.**

I. Introduction

Understanding the affordances of objects underpins a robot's capability to perform purposeful interactions in unstructured environments [1–3]. Given an open-ended task instruction specified in natural language, a robot must first identify the action possibilities afforded by the environment based on its visual perception. In particular, this understanding should extend beyond objects or object parts to encompass fine-grained details down to the level of pixels. For instance, the robot might need to identify the exact grasp point on an unseen saucepot, the broom area to sweep trash, or the hanger's shoulders to hang clothes (Fig. 1). While learning visual affordances from manually annotated datasets with closed vocabulary has been extensively explored in the literature [4–9], scaling affordance learning to open-world scenarios conditioned on free-form task instructions remains a long-standing challenge.

Vision-language models (VLMs) have demonstrated the ability to internalize world knowledge by pretraining on large-scale image-text datasets [10, 11]. Recent works also suggest that they encode affordance knowledge in the language space [12] (e.g., "handle should be grasped for opening drawers"). However, the effective grounding of this knowledge in the continuous *spatial* domain remains an open question. In contrast, self-supervised vision models [13, 14] provide general-purpose pixel-level features that capture low-level structures of objects. However, they are not conditioned on specific open-world task semantics, which is imperative for task-level generalization in robotic manipulation.

In this work, we introduce **Unsupervised Affordance Distillation (UAD)**, a method that extracts affordance knowledge from foundation models and distills it into a task-conditioned affordance model, *all without manual annotation*. Notably, UAD leverages the complementary strengths of vision-language models and large vision models to automatically annotate a large-scale dataset with fine-grained `<instruction, visual affordance>` pairs. We then use the dataset to train a task-conditioned affordance model, by reusing and freezing the weights of DINOv2 [14] and training only a lightweight task-conditioned decoder. We

Correspondence: Wenlong Huang (wenlongh@stanford.edu)

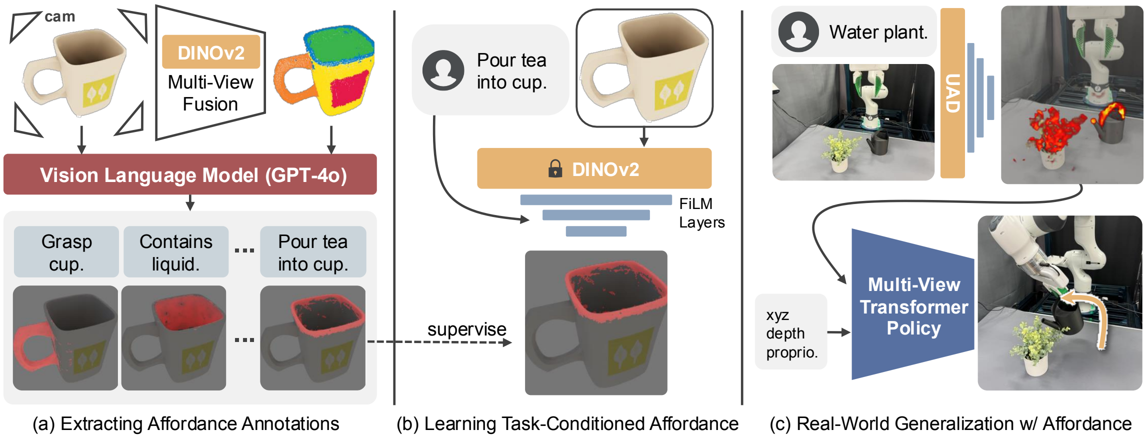

cam
DINOv2
Multi-View Fusion
Vision Language Model (GPT-4o)
Grasp cup.
Contains liquid.
...
Pour tea into cup.
supervise
Pour tea into cup.
DINOv2
FiLM Layers
(a) Extracting Affordance Annotations
(b) Learning Task-Conditioned Affordance
Water plant.
UAD
xyz
depth
proprio.
Multi-View Transformer Policy
(c) Real-World Generalization w/ Affordance

Fig. 2: **Overview of Unsupervised Affordance Distillation (UAD).** Using renderings of 3D objects, we first perform multi-view fusion of DINOv2 features and clustering to obtain fine-grained semantic regions of objects, which are then fed to VLM for proposing relevant tasks and corresponding regions **(a)**. The extracted affordance is then distilled by training a language-conditioning FiLM atop frozen DINOv2 features **(b)**. The learned task-conditioned affordance model provides in-the-wild prediction for diverse fine-grained regions, which are used as observation space for manipulation policies **(c)**.

demonstrate a superior performance of the model evaluated *zero-shot* on existing benchmarks [15], along with exceptional generalization to a real-world robotic dataset [16] involving unseen objects in novel environments.

To translate these generalization properties into robust manipulation behaviors, we propose an imitation learning policy that uses *affordance as the observation space* [17] provided by the pre-trained UAD. This approach sidesteps the common challenge of learning generalizable visual representations in vision-based manipulation on scarce interaction data and provides a manipulation-centric alternative to various pre-trained visual representations that are often tailored for vision tasks, such as CLIP [18]. Specifically, we demonstrate that the proposed framework possesses the unique advantage of generalizing to unseen environment configurations, object instances, object categories, and even novel task instructions, after training on as few as 10 demonstrations.

To summarize, our contributions are as follows: 1) We propose an unsupervised pipeline to automatically extract fine-grained affordance annotations using off-the-shelf vision-language models (VLMs) and large vision models (LVMs); 2) We scale the training of a task-conditioned affordance model that outperforms prior methods on existing benchmarks, despite evaluated zero-shot; 3) We show that using affordance as the observation space in an imitation learning policy enables generalization to unseen environments, object instances, object categories, and task instructions, while training with only a handful of demonstrations.

II. Related Works

A. Learning and Discovering Visual Affordance for Robotics

Affordance [19] can be defined as action possibilities that are readily perceivable by an actor [20]. This topic has two levels, namely learning and discovering affordance, and using affordance for downstream tasks [9]. These topics have been extensively studied in robotics and related fields as covered in several recent surveys [4–9]. Affordance is typically expressed in perceptual space of the agent. They differ in how the afforded actions are inferred: one can infer the action from probability maps (e.g., action possibility estimates), or by a direct mapping from the observations (e.g., keypoints or descriptors [21–36]). Action possibilities are often represented as affordance maps, e.g., in the formats of probability distributions over image space [2, 37–46] or continuous action possibilities [35, 47–57], which have the same dimensions as the input image. Their values typically indicate the likelihood of executing a certain action at each pixel location [9]. To learn a model that predicts affordance, deep learning-based methods are widely adopted [7], which require a large amount of training data. For training, one can utilize supervised learning with existing datasets (e.g., [37, 58–61]) or self-supervised learning [25, 50–52, 62]. While many works focus on developing models to be trained on existing datasets, our work uniquely investigates extracting affordance from large general-purpose, pre-trained models.

B. Pre-trained Visual Representation for Manipulation

An important application of UAD is visuomotor learning for robotic manipulation. Specifically, we incorporate UAD as the observation space for robot policy, akin to related literature studying pre-trained visual representation for manipulation [63–77], which can be coarsely categorized into those that are task-agnostic [63–71, 78–80] and those that are task-conditioned [72–77, 81]. We study the later setting, where visual representation differs depending on the agent's objective. To learn the association between visual features and language features, previous work typically relies on a CLIP-like [18] objective, which often exhibits "bag-of-words" behaviors that focus little on fine-grained visual details, as suggested by recent studies [82–85]. In this work, we aim to address such limitations by proposing a data annotation pipeline that can effectively scale up the training of task-conditioned visual features *without human annotations* while focusing on fine-grained predictions.

C. Foundation Models for Robotics

Leveraging foundation models for robotics is an active area of research [86–89], with many works focusing on open-world reasoning and goal specification [90–104]. In this work, we are interested in acquiring general-purpose knowledge about affordance from existing foundation models through extraction and distillation, to obtain a model that maps task instructions to visual affordance. To this end, we focus on VLMs that can perform visual question answering [10, 12, 18, 105–107] and self-supervised vision models [13, 14, 108–113] that can provide fine-grained pixel-level features. However, none of the aforementioned models directly supports the desired input-output mapping in this work. As a result, the proposed UAD consists of a two-stage extraction and distillation process, with the critical insight being reformatting the visual affordance understanding as a visual question-answering problem. Similar visual prompting techniques are also explored in prior work [36, 90–92]. In comparison, in this work, we further distill the extracted knowledge into a specialized visual affordance model that is not only more efficient but also provides *fine-grained continuous* predictions *directly in visual space*. Furthermore, we focus our study on its extensive utility in supporting generalization to various conditions in robotic manipulation.

III. Method

In this section, we discuss (A) how we extract affordance annotations from foundation models, (B) how we train a task-conditioned affordance model based on these annotations, and (C) how we leverage the learned affordance in imitation learning policies by using affordance as the observation space for generalization for robotic manipulation.

A. Extracting Affordance Annotations

We are interested in visual affordance on pixel-level functional regions of objects, which we posit to be useful for downstream vision-based manipulation tasks. However, manually labeling a large-scale affordance dataset is costly. Therefore, we want to extract affordance annotations from existing foundation models to construct a diverse dataset of the following triplets: RGB images $I \in \mathbb{R}^{H\times W\times 3}$, free-form task instructions $\mathcal{T}$, and task-conditioned affordance map $A \in [0,1]^{H\times W}$.

Dataset. Although we are interested in obtaining a dataset of 2D annotations, we empirically find that the proposed pipeline performs significantly better when 3D consistency is enforced—a similar observation was also made in the recent investigation of pre-trained 2D visual features for manipulation and open-vocabulary 3D segmentation [79, 114, 115]. To this end, we focus on generating unsupervised affordance annotations in 2D from individually rendered 3D objects. We will later discuss how we can train an affordance model that nevertheless generalizes to multi-object scenes even in the real world despite only being trained on single objects rendered in simulation.

We use a subset of the 3D assets from BEHAVIOR-1K [116], as the objects are tailored for the manipulation context. In total, the object database consists of 206 objects from 76 object categories, along with 667 task instructions. Post paper acceptance, we additionally conduct a case study of scaling to more diverse object database, such as Objaverse-XL [117], which collectively amounts to more than 10,000 object-instruction pairs. Details are provided in Appendix.

An overview of our pipeline is shown in Fig. 2(a). At a high level, we leverage LVMs to find fine-grained semantic regions for each object and VLMs to propose candidate task instructions relevant to each object. Then, we use VLMs to associate the regions and the task instructions. In the following, we introduce each component in order:

Fine-Grained Region Proposal. For each 3D object, we first spawn it in an empty scene and render 14 views around the object to obtain K RGB images $I_{i=1}^{K} \in \mathbb{R}^{H\times W\times 3}$ and its aggregated point cloud in world coordinates $P \in \mathbb{R}^{N\times 3}$. For each I_i, we extract pixel-wise features $F_i \in \mathbb{R}^{H\times W\times d}$ from DINOv2. We then adopt the procedure in [114] to fuse the 2D features to obtain a global 3D feature field $F_{\text{global}} \in \mathbb{R}^{N\times d}$ over the object point cloud P. To obtain fine-grained semantic regions of objects, we first apply PCA on F_{global} to obtain $F_{\text{reduced}} \in \mathbb{R}^{N\times 3}$, which is found to make features less sensitive to local texture. Finally, we cluster the object points into M regions using the Euclidean distance on their features F_{reduced}, where the clustering algorithm automatically identifies the number of clusters M. We denote the region labels for all object points as $r_{n=1}^{N} \in \{1, ..., M\}$.

Task Instruction Proposal. To identify the plausible task instructions for each object conditioning on the proposed regions, we perform visual prompting using a VLM, i.e., GPT-4o [10] in this work. For each object, we first identify a natural-looking view by calculating the cosine similarity between the object category name and the corresponding RGB image under CLIP [18] embedding. We then visualize the proposed regions by assigning unique colors and overlay them on the original image. We provide the overlaid image, original image, and object category name as input for the VLM and query it to propose a set of task instructions $\{\mathcal{T}_1, ..., \mathcal{T}_J\}$ associated with this object. For instance, for a coffee mug, the VLM will propose task instructions such as "rim of the coffee mug - region for drinking and pouring". The complete prompts can be found in the Appendix.

Region and Instruction Mapping We then use a similar procedure to query the VLM to associate the task instructions it proposed with the most appropriate clustered region. As a result, we obtain a mapping from each task instruction to exactly one region on the object. Eventually, we want to create a continuous affordance map $A \in [0,1]^{H\times W}$ because we believe that affordance is fundamentally continuous rather than binary. Certain regions are more closely associated to the specified task (e.g. middle of a handle for grasping) than others (e.g. the tip of the handle), which is better captured by a continuous formulation. To do so, for each region r identified by the VLM, we first average the features of the corresponding points to obtain a reference feature $f_{\text{ref}} \in \mathbb{R}^d$. Then we compute the cosine similarity score between f_{ref} and F_{global} to obtain a $[0,1]$ similarity score for each 3D

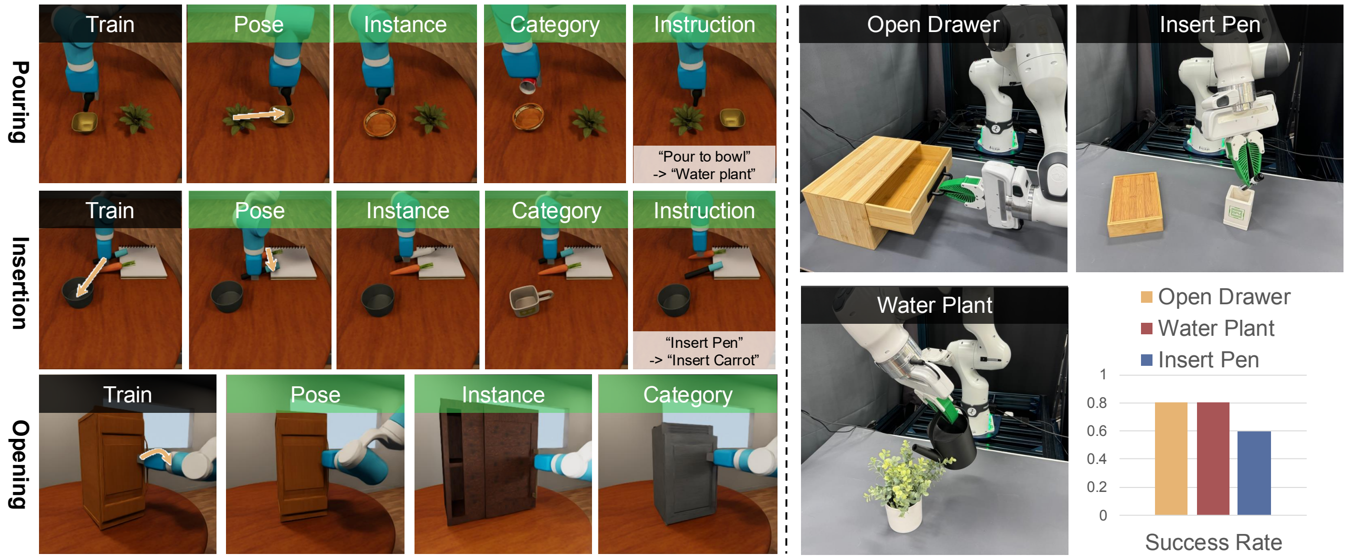


Fig. 3: Tasks for evaluating UAD. Left: tasks in simulation along with different generalization requirements. Right: tasks in the real world and the corresponding success rate achieved by UAD-based policies.

point. Finally, we project object points along with their scores to each camera view to obtain the final affordance map $A \in [0,1]^{H\times W}$. This operation converts *discrete* decisions produced by VLM to *continuous* pixel-level values; intuitively, it can be interpreted as per-pixel likelihood of whether it "affords" the given task. This is in contrast to many prior works [118] that consider image-level distributions (i.e., all pixels sum to one) where affordance magnitude is normalized by afforded region size. In summary, the pipeline produces a dataset with triplets of $(I, \mathcal{T}, A)$.

B. Learning Task-conditioned Affordance Model

To train an affordance model that generalizes to real-world multi-object scenes using only synthetic single-object data, we leverage the pre-trained DINOv2 [14] by freezing its weights and only training a lightweight language-conditioned module on top. Specifically, we first obtain the language embedding $e_{\mathcal{T}}$ for each entry in the dataset using OpenAI APIs. To condition the features from DINOv2 on the language embeddings, we use FiLM layers [119] that take in the language embedding $e_{\mathcal{T}}$ as well as the pixel-space features $X \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ and output $X' \in \mathbb{R}^{H\times W\times C_{\text{out}}}$, where C_{in} and C_{out} are input and output channel dimensions, respectively. We use 3 FiLM layers with output channels $[256, 64, 1]$, which produce logits at each pixel location as the final output $\hat{A} \in [0,1]^{H\times W}$. We note that the learned transformation for each channel by FiLM is agnostic to pixel location, which is suitable for our intent to build an association between the DINOv2 feature and task instructions. We use binary cross-entropy as the loss function, computed between the ground truth affordance map A and the predicted logits of the affordance map $\hat{A}$. We term the learned affordance model as UAD.

C. Policy Learning with Affordance as Observation Space

UAD can be naturally integrated into existing vision-based policy architectures for manipulation as an encoder for the visual input. Effectively, instead of learning a *task-agnostic* visual representation as in most existing policy architectures, UAD serves as fine-grained visual attention for the policy that contains prior knowledge *conditioned on tasks at hand*. To investigate its capability, we integrate UAD with a multi-view transformer policy adopted from RVT [120, 121]. We assume access to detailed language instructions (e.g., "grasp watering can", "align spout", "water plant"). Using the given instruction, we first predict the affordance map for each view $\mathbb{R}^{H\times W}$. Then, we follow RVT to augment each view with the corresponding depth value and the (x, y, z) coordinates of points in the world frame, as well as a global proprioception vector. The policy outputs a 7-dimension action that includes a 6-DoF end-effector pose and a binary gripper action. We train the policy using imitation learning and focus our investigation on its generalization capabilities. Even though we do not finetune the affordance model in policy training, we can effectively train the policy using only a handful of demonstrations while exhibiting generalization capabilities to various conditions leveraging UAD.

IV. Experiments

We seek to answer the following research questions: (A) Despite only being trained on rendered 3D objects, can UAD generalize to real-world scenes from existing robotic datasets in affordance prediction, and how does it compare to prior methods on visual affordance benchmarks? (B) Using UAD as observation space, what generalization properties does a visuomotor policy have compared to other pre-trained representations? (C) How well does an UAD-based policy perform in real-world manipulation tasks?

A. Task-Conditioned Affordance Prediction

In this section, we focus on how UAD performs on task-conditioned visual affordance prediction. Note that we train a *single* UAD only on rendered 3D objects, which is used to perform evaluations across all settings discussed below.

UAD generalizes to novel instances, categories, and instructions on rendered objects. We first perform a sanity check on how well UAD performs within the same domain of simulator rendered, single object images. We construct four evaluation sets of image-text pairs that contain, respectively, 1) training data, 2) novel object instances, 3) novel categories, and 4) novel instructions. We use Amazon MTurk to obtain the ground truth for evaluations, with details in the Appendix. Based on recent study [118], we use the Area Under ROC Curve (AUC) as the metric for evaluation, as it evaluates the predicted affordance map as a per-pixel classifier of the ground-truth mask [118], closest to our interpretation. Evaluated on 100 <instruction, visual affordance> pairs per setting, UAD achieved an AUC score of at least 0.92 across all four settings, indicating its strong generalization capability, as well as the consistency between UAD and human predictions.

Leveraging pre-trained features, UAD can seamlessly generalize to real-world robotic scenes. To create an affordance prediction evaluation set tailored for manipulation, we investigate UAD on a subset of DROID [16], a real-world robotic dataset containing trajectories of robots performing manipulation tasks in diverse, in-the-wild scenes. Specifically, we select task episodes from DROID that involve interaction with specific, fine-grained object regions, rather than tasks with ambiguous instructions (e.g., "pick up the colored cube," where any part of the object can be manipulated). We extend the built-in task descriptions with additional text details at the same level of granularity as those in training. For example, the task "pick up the lid and put it on the pot" is broken down into "pick up the lid" and "align with the rim of the pot". For each episode, we capture the first frame from two table-mounted third-person cameras. We center-crop the images and filter out those where the key objects are not clearly visible (e.g., due to occlusion). We follow the same procedure as in the previous section to obtain the ground-truth labels from Amazon MTurk.

We compare UAD to CLIP [18] and OpenSeeD [122], an open-vocabulary segmentation model. Results are shown in Fig. 4. Despite only trained on rendered single objects, by leveraging DINOv2 [14] as backbone, UAD can generalize to in-the-wild, multi-object, often even cluttered scenes. Notably, compared to CLIP, UAD provides much more fine-grained and robust features, often focused specifically on potential regions of interaction. Compared to open-vocabulary segmentation, which outputs binary segmentation, UAD produces a continuous representation. Notably, even when target objects or parts are small, UAD can consistently produce per-pixel, continuous prediction, whereas this is observed to be a typical failure case for segmentation models.

UAD performs competitively on human activity affordances even though certain activities or objects are entirely unseen. Motivated by the promising generalization of UAD, we are also interested in investigating how UAD may perform in existing affordance prediction benchmark focused on human activities, such as AGD20K [58]. The results are shown in Fig. 5 and Tab. I.

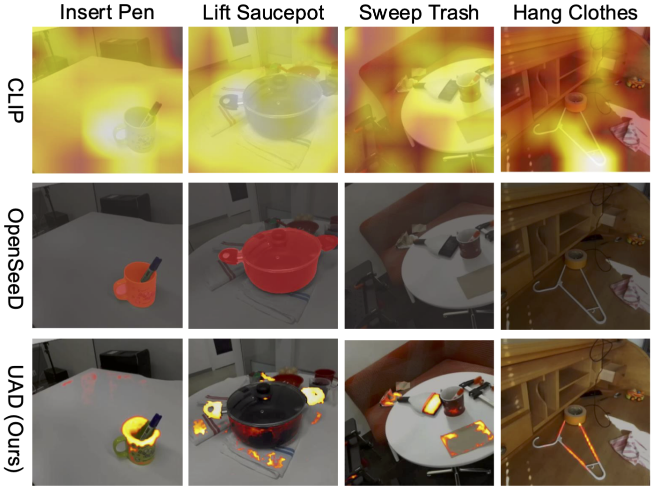


Fig. 4: Task-conditioned affordance prediction results on the DROID dataset. Average AUC scores (evaluated on the entire dataset): 0.500 (CLIP), 0.836 (OpenSeeD), 0.840 (Ours).

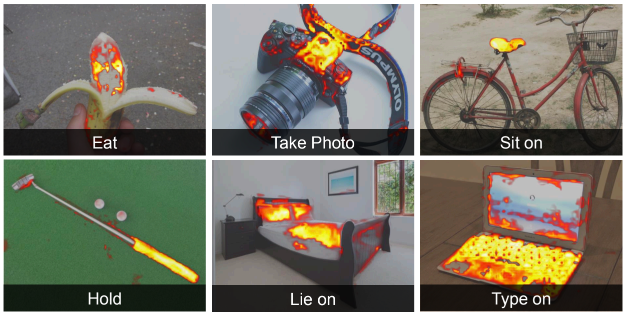


Fig. 5: Zero-shot generalization to affordance predictions in human activities from AGD20K.

Methods	KLD (↓)	SIM (↑)	NSS (↑)
Cross-View-AG [58]	1.787	0.285	0.829
LOCATE [123]	1.405	0.372	1.157
3DOI [124]	3.565	0.227	0.657
AffordanceLLM [125]	1.463	**0.377**	1.070
UAD (Ours)	**0.526**	0.366	**1.359**

TABLE I: Evaluation results on AGD20K test split.

For evaluation, since UAD conditions on free-form language and AGD20K contains only a list of pre-defined actions, we format the instructions as "region to <action> the <object>". We follow the evaluation procedure defined in AGD20K [58] and evaluate UAD on the test split using the same metrics as in previous work, namely KL Divergence (KLD), Similarity Metric (SIM), and Normalized Scanpath Saliency (NSS). Most of these metrics consider affordance prediction as an image-level distribution rather than a pixel-level distribution. As a result, we adopt a simple normalization and observe that while UAD is not trained on a similar distribution, UAD still performs competitively compared to prior work. Interestingly, UAD can even generalize to a number of human activities that involve objects and task instructions completely out of distribution from our training set, such as "eating bananas", "taking photos", "sitting on bicycles", "holding golf clubs", "lying on bed", and "typing on computers", as shown in Fig. 5.

B. *Policy Learning in Simulation*

Using UAD as observation space (Fig. 2(c)), we evaluate the generalization properties of a transformer-based policy learned via imitation learning. We perform our experiments in OmniGibson [116], equipped with photo-realistic rendering and a variety of everyday objects.

We select three tasks that require fine-grained reasoning of object affordance, *Pouring*, *Opening*, and *Insertion*, each with varying objects designed to assess the generalization performance of the policy (Fig. 3). For each task, we train a policy on 10 scripted demonstrations with randomization in object poses. We evaluate the trained policy against four generalization settings: new object poses, instances/models, categories, and task instructions (Fig. 3). To better attribute generalization capabilities solely brought by affordance prediction, for novel categories, we choose objects with similar functional structure—for example, one needs to grasp the lower body to lift up both the beer bottle and Coke can. Since UAD focuses on generalization in task-conditioned visual prediction, when designing evaluation to novel instructions, we focus on scenarios where correct identifying affordance would lead to successful completion of the task. For instance, for both pouring fluid and watering plants, the robot needs to approach the correct region near the target object (bowl and pot plant, respectively) and tilt the fluid container.

We compare against baseline policies that use RGB images or other pre-trained visual representations as observations, including DINOv2 [14], CLIP [18], and Voltron [73]. Additional setup details can be found in the Appendix. The results are shown in Fig. 6. Each bar represents the average success rate across all three tasks, with each task-generalization setup combination being evaluated over 15 trials. In general, although trained on only a handful of demonstrations, the UAD-based policy demonstrates promising generalization capabilities in all settings evaluated. We summarize our main takeaways below:

- UAD is robust against variations in object appearance, e.g. successfully manipulating white markers in *Insertion* tasks despite training only on black ones..
- UAD is particularly advantageous on tasks requiring fine-grained visual perception, allowing it to outperform baselines on the *Opening* task involving detecting grasp points on thin handles of drawers.
- As UAD is conditioned on task instructions while offering precise affordance prediction, it can also generalize to variations in instructions that control the objects of interaction via natural language.

C. *Policy Learning in the Real World*

To demonstrate that UAD-based policies can solve real-world tasks, we further evaluate models on three robotic manipulation tasks as shown in Fig. 3. We use a Franka Emika Panda robot with a tabletop setup (more details about hardware setup can be found in the Appendix), with two RGB-D cameras mounted on the opposite sides of the workspace. Following the policy learning setup in Sec. IV-B, we train a policy for each task using 10 human demonstrations collected using kinesthetic teaching. Compared to the simulation setup, the real-world environment poses additional challenges, such as identifying affordances for visually diverse real-world objects, as well as subjecting to additional noise introduced by various components in the real-world system stack, including RGB-D cameras and low-level controllers. The results are shown in Fig. 3, with success rates averaged across 10 trials for each task. Overall, the UAD-based policy can perform real-world manipulation tasks with an average success rate of 73%. The predicted affordance maps contain fine-grained details that allow for precise 6-DoF manipulation, such as inserting a pen and opening a drawer.

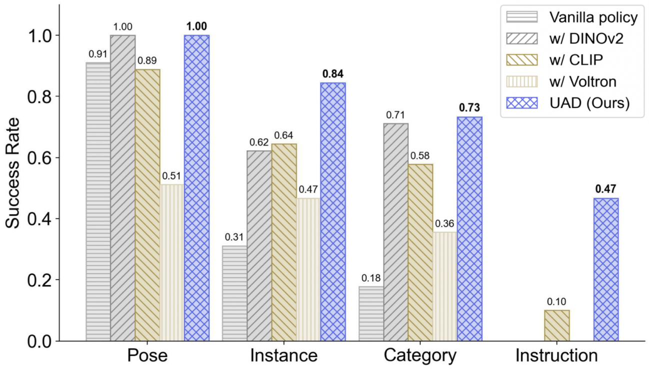


Fig. 6: Generalization performance of UAD in three simulation tasks. UAD shows better generalization capabilities compared to the baselines.

V. Conclusion & Limitations

Learning and discovering object affordance is an important step toward generalizable robotic manipulation. We propose Unsupervised Affordance Distillation (UAD), a novel method that distills affordance knowledge from foundation models into a task-conditioned affordance model, without relying on manually annotated datasets. Our model achieves competitive performance on existing affordance prediction benchmarks and demonstrates strong generalization capabilities in real-world robotic tasks.

Despite the promising findings, a few limitations remain. First, UAD focuses on extracting visual affordance from foundation models. Although we observe promising generalization when using it as the observation space for imitation learning policy, it does not immediately provide generalization at the motion level. Second, we only consider the interpretation of affordance for a single static frame. However, manipulation tasks are typically concerned with multi-step visual understanding and behaviors. Third, the extracted training dataset contains only single objects renderings – extending the annotations to real-world multi-object images may enable better grounding of world knowledge in foundation models to continuous spatial domains.

Acknowledgment

This work is in part supported by NSF RI #2211258 and #2338203, ONR N00014-23-1-2355, ONR MURI N00014-22-1-2740, and ONR MURI N00014-24-1-2748.

References

[1] E. Şahin, M. Cakmak, M. R. Doğar, E. Uğur, and G. Üçoluk, "To afford or not to afford: A new formalization of affordances toward affordance-based robot control," *Adaptive Behavior*, vol. 15, no. 4, pp. 447–472, 2007.

[2] S. Bahl, R. Mendonca, L. Chen, U. Jain, and D. Pathak, "Affordances from human videos as a versatile representation for robotics," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 13 778–13 790.

[3] C.-C. Hsu, Z. Jiang, and Y. Zhu, "Ditto in the house: Building articulation models of indoor scenes through interactive perception," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3933–3939.

[4] L. Jamone, E. Ugur, A. Cangelosi, L. Fadiga, A. Bernardino, J. Piater, and J. Santos-Victor, "Affordances in psychology, neuroscience, and robotics: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 1, pp. 4–25, 2016.

[5] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, "A brief review of affordance in robotic manipulation research," *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.

[6] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *ACM Computing Surveys (CSUR)*, vol. 54, no. 3, pp. 1–35, 2021.

[7] D. Chen, D. Kong, J. Li, S. Wang, and B. Yin, "A survey of visual affordance recognition based on deep learning," *IEEE Transactions on Big Data*, 2023.

[8] W. Liu, A. Daruna, M. Patel, K. Ramachandruni, and S. Chernova, "A survey of semantic reasoning frameworks for robotic systems," *Robotics and Autonomous Systems*, vol. 159, p. 104294, 2023.

[9] X. Yang, Z. Ji, J. Wu, and Y.-K. Lai, "Recent advances of deep robotic affordance learning: a reinforcement learning perspective," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, no. 3, pp. 1139–1149, 2023.

[10] OpenAI, "Gpt-4 technical report," *arXiv*, 2023.

[11] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth *et al.*, "Gemini: a family of highly capable multimodal models," *arXiv preprint arXiv:2312.11805*, 2023.

[12] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat *et al.*, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[13] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

[14] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[15] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Grounded affordance from exocentric view," *arXiv preprint arXiv:2208.13196*, 2022.

[16] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," in *Robotics: Science and Systems*, 2024.

[17] R. Jonschkowski and O. Brock, "Learning state representations with robotic priors," *Autonomous Robots*, vol. 39, pp. 407–428, 2015.

[18] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International Conference on Machine Learning*. PMLR, 2021, pp. 8748–8763.

[19] J. J. Gibson, "The theory of affordances," *The Ecological Approach to Visual Perception*, 1977.

[20] D. Norman, *The design of everyday things: Revised and expanded edition*. Basic books, 2013.

[21] T. Schmidt, R. Newcombe, and D. Fox, "Self-supervised visual descriptor learning for dense correspondence," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 420–427, 2016.

[22] P. R. Florence, L. Manuelli, and R. Tedrake, "Dense object nets: Learning dense visual object descriptors by and for robotic manipulation," *arXiv preprint arXiv:1806.08756*, 2018.

[23] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *The International Symposium of Robotics Research*. Springer, 2019, pp. 132–157.

[24] T. D. Kulkarni, A. Gupta, C. Ionescu, S. Borgeaud, M. Reynolds, A. Zisserman, and V. Mnih, "Unsupervised learning of object keypoints for perception and control," *Advances in neural information processing systems*, vol. 32, 2019.

[25] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "Keto: Learning keypoint representations for tool manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7278–7285.

[26] P. Sundaresan, J. Grannen, B. Thananjeyan, A. Balakrishna, M. Laskey, K. Stone, J. E. Gonzalez, and K. Goldberg, "Learning rope manipulation policies using dense object descriptors trained on synthetic depth data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9411–9418.

[27] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning," *arXiv preprint arXiv:2009.05085*, 2020.

[28] B. Chen, P. Abbeel, and D. Pathak, "Unsupervised learning of visual 3d keypoints for control," in *International Conference on Machine Learning*. PMLR, 2021, pp. 1539–1549.

[29] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural descriptor fields: Se (3)-equivariant object representations for manipulation," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6394–6400.

[30] A. Simeonov, Y. Du, Y.-C. Lin, A. R. Garcia, L. P. Kaelbling, T. Lozano-Pérez, and P. Agrawal, "Se (3)-equivariant relational rearrangement with neural descriptor fields," in *Conference on Robot Learning*. PMLR, 2023, pp. 835–846.

[31] M. Vecerik, C. Doersch, Y. Yang, T. Davchev, Y. Aytar, G. Zhou, R. Hadsell, L. Agapito, and J. Scholz, "Robotap: Tracking arbitrary points for few-shot visual imitation," *arXiv preprint arXiv:2308.15975*, 2023.

[32] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1830–1836.

[33] C. Wen, X. Lin, J. So, K. Chen, Q. Dou, Y. Gao, and P. Abbeel, "Any-point trajectory modeling for policy learning," *arXiv preprint arXiv:2401.00025*, 2023.

[34] H. Bharadhwaj, R. Mottaghi, A. Gupta, and S. Tulsiani, "Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation," 2024.

[35] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2870–2877, 2021.

[36] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," *arXiv preprint arXiv:2409.01652*, 2024.

[37] A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 1374–1381.

[38] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 91–98.

[39] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5908–5915.

[40] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[41] F.-J. Chu, R. Xu, L. Seguin, and P. A. Vela, "Toward affordance detection and ranking on novel objects for real-world robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4070–4077, 2019.

[42] P. Mandikal and K. Grauman, "Learning dexterous grasping with object-centric visual affordances," in *2021 IEEE international con-*

ference on robotics and automation (ICRA). IEEE, 2021, pp. 6169–6176.

[43] A. Hämäläinen, K. Arndt, A. Ghadirzadeh, and V. Kyrki, "Affordance learning for end-to-end visuomotor robot control," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 1781–1788.

[44] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, "Affordance learning from play for sample-efficient policy learning," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6372–6378.

[45] H. Bharadhwaj, A. Gupta, and S. Tulsiani, "Visual affordance prediction for guiding robot exploration," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3029–3036.

[46] M. K. Srirama, S. Dasari, S. Bahl, and A. Gupta, "Hrp: Human affordances for robotic pre-training," *arXiv preprint arXiv:2407.18911*, 2024.

[47] B. Moldovan, P. Moreno, M. Van Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *2012 ieee international conference on robotics and automation*. IEEE, 2012, pp. 4373–4378.

[48] C. Pohl, K. Hitzler, R. Grimm, A. Zea, U. D. Hanebeck, and T. Asfour, "Affordance-based grasping and manipulation in real world applications," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9569–9576.

[49] A. Zeng, S. Song, K.-T. Yu, E. Donlon, F. R. Hogan, M. Bauza, D. Ma, O. Taylor, M. Liu, E. Romo *et al.*, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 690–705, 2022.

[50] J. Cai, H. Cheng, Z. Zhang, and J. Su, "Metagrasp: Data efficient grasping by affordance interpreter network," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 4960–4966.

[51] H. Wu, Z. Zhang, H. Cheng, K. Yang, J. Liu, and Z. Guo, "Learning affordance space in physical world for vision-based robotic object manipulation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4652–4658.

[52] S. Yang, W. Zhang, R. Song, J. Cheng, and Y. Li, "Learning multi-object dense descriptor for autonomous goal-conditioned grasping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 4109–4116, 2021.

[53] D. Xu, A. Mandlekar, R. Martín-Martín, Y. Zhu, S. Savarese, and L. Fei-Fei, "Deep affordance foresight: Planning through what can be done in the future," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6206–6213.

[54] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6813–6823.

[55] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan, "Do as i can and not as i say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[56] O. Mees, J. Borja-Diaz, and W. Burgard, "Grounding language with visual affordances over unstructured data," *arXiv preprint arXiv:2210.01911*, 2022.

[57] Y.-H. Wu, J. Wang, and X. Wang, "Learning generalizable dexterous manipulation from human grasp affordance," in *Conference on Robot Learning*. PMLR, 2023, pp. 618–629.

[58] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Learning affordance grounding from exocentric images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[59] G. Li, V. Jampani, D. Sun, and L. Sevilla-Lara, "Locate: Localize and transfer object parts for weakly supervised affordance grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[60] J. Jian, X. Liu, M. Li, R. Hu, and J. Liu, "Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 713–14 724.

[61] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 1778–1787.

[62] D. Turpin, L. Wang, S. Tsogkas, S. Dickinson, and A. Garg, "Gift: Generalizable interaction-aware functional tool affordances without labels," *arXiv preprint arXiv:2106.14973*, 2021.

[63] R. Shah and V. Kumar, "Rrl: Resnet as representation for reinforcement learning," *arXiv preprint arXiv:2107.03380*, 2021.

[64] J. Pari, N. M. Shafiullah, S. P. Arunachalam, and L. Pinto, "The surprising effectiveness of representation learning for visual imitation," *arXiv preprint arXiv:2112.01511*, 2021.

[65] S. Parisi, A. Rajeswaran, S. Purushwalkam, and A. Gupta, "The unsurprising effectiveness of pre-trained vision models for control," in *international conference on machine learning*. PMLR, 2022, pp. 17 359–17 371.

[66] T. Xiao, I. Radosavovic, T. Darrell, and J. Malik, "Masked visual pre-training for motor control," *arXiv preprint arXiv:2203.06173*, 2022.

[67] I. Radosavovic, T. Xiao, S. James, P. Abbeel, J. Malik, and T. Darrell, "Real-world robot learning with masked visual pre-training," in *Conference on Robot Learning*. PMLR, 2023, pp. 416–426.

[68] Y. J. Ma, S. Sodhani, D. Jayaraman, O. Bastani, V. Kumar, and A. Zhang, "Vip: Towards universal visual reward and representation via value-implicit pre-training," *arXiv preprint arXiv:2210.00030*, 2022.

[69] A. Majumdar, K. Yadav, S. Arnaud, J. Ma, C. Chen, S. Silwal, A. Jain, V.-P. Berges, T. Wu, J. Vakil *et al.*, "Where are we in the search for an artificial visual cortex for embodied intelligence?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 655–677, 2023.

[70] K. Burns, Z. Witzel, J. I. Hamid, T. Yu, C. Finn, and K. Hausman, "What makes pre-trained visual representations successful for robust manipulation?" *arXiv preprint arXiv:2312.12444*, 2023.

[71] K. Yadav, R. Ramrakhya, A. Majumdar, V.-P. Berges, S. Kuhar, D. Batra, A. Baevski, and O. Maksymets, "Offline visual representation learning for embodied navigation," in *Workshop on Reincarnating Reinforcement Learning at ICLR 2023*, 2023.

[72] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[73] S. Karamcheti, S. Nair, A. S. Chen, T. Kollar, C. Finn, D. Sadigh, and P. Liang, "Language-driven representation learning for robotics," *arXiv preprint arXiv:2302.12766*, 2023.

[74] Y. J. Ma, V. Kumar, A. Zhang, O. Bastani, and D. Jayaraman, "Liv: Language-image representations and rewards for robotic control," in *International Conference on Machine Learning*. PMLR, 2023, pp. 23 301–23 320.

[75] A. Khandelwal, L. Weihs, R. Mottaghi, and A. Kembhavi, "Simple but effective: Clip embeddings for embodied ai," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 829–14 838.

[76] Y. Cui, S. Niekum, A. Gupta, V. Kumar, and A. Rajeswaran, "Can foundation models perform zero-shot task specification for robot manipulation?" in *Learning for dynamics and control conference*. PMLR, 2022, pp. 893–905.

[77] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on robot learning*. PMLR, 2022, pp. 894–906.

[78] X. Lin, J. So, S. Mahalingam, F. Liu, and P. Abbeel, "Spawnnet: Learning generalizable visuomotor skills from pre-trained network," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 4781–4787.

[79] Y. Wang, G. Yin, B. Huang, T. Kelestemur, J. Wang, and Y. Li, "GenDP: 3d semantic fields for category-level generalizable diffusion policy," in *8th Annual Conference on Robot Learning*, 2024. [Online]. Available: https://openreview.net/forum?id=7wMlwhCvjS

[80] G. Jiang, Y. Sun, T. Huang, H. Li, Y. Liang, and H. Xu, "Robots pre-train robots: Manipulation-centric robotic representation from large-scale robot datasets," 2024. [Online]. Available: https://arxiv.org/abs/2410.22325

[81] H. Huang, F. Liu, L. Fu, T. Wu, M. Mukadam, J. Malik, K. Goldberg, and P. Abbeel, "Otter: A vision-language-action model

with text-aware visual feature extraction," 2025. [Online]. Available: https://arxiv.org/abs/2503.03734

[82] S. Tong, Z. Liu, Y. Zhai, Y. Ma, Y. LeCun, and S. Xie, "Eyes wide shut? exploring the visual shortcomings of multimodal llms," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9568–9578.

[83] T. Thrush, R. Jiang, M. Bartolo, A. Singh, A. Williams, D. Kiela, and C. Ross, "Winoground: Probing vision and language models for visio-linguistic compositionality," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5238–5248.

[84] M. Yuksekgonul, F. Bianchi, P. Kalluri, D. Jurafsky, and J. Zou, "When and why vision-language models behave like bags-of-words, and what to do about it?" in *The Eleventh International Conference on Learning Representations*, 2023.

[85] C.-Y. Hsieh, J. Zhang, Z. Ma, A. Kembhavi, and R. Krishna, "Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality," *Advances in neural information processing systems*, vol. 36, 2024.

[86] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, Z. Zhao *et al.*, "Toward general-purpose robots via foundation models: A survey and meta-analysis," *arXiv preprint arXiv:2312.08782*, 2023.

[87] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman *et al.*, "Foundation models in robotics: Applications, challenges, and the future," *arXiv preprint arXiv:2312.07843*, 2023.

[88] K. Kawaharazuka, T. Matsushima, A. Gambardella, J. Guo, C. Paxton, and A. Zeng, "Real-world robot applications of foundation models: A review," *arXiv preprint arXiv:2402.05741*, 2024.

[89] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, "Foundation models for decision making: Problems, methods, and opportunities," *arXiv preprint arXiv:2303.04129*, 2023.

[90] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "Copa: General robotic manipulation through spatial constraints of parts with foundation models," *arXiv preprint arXiv:2403.08248*, 2024.

[91] F. Liu, K. Fang, P. Abbeel, and S. Levine, "Moka: Open-vocabulary robotic manipulation through mark-based visual prompting," *arXiv preprint arXiv:2403.03174*, 2024.

[92] S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu *et al.*, "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," *arXiv preprint arXiv:2402.07872*, 2024.

[93] Y. Hu, F. Lin, T. Zhang, L. Yi, and Y. Gao, "Look before you leap: Unveiling the power of gpt-4v in robotic vision-language planning," *arXiv preprint arXiv:2311.17842*, 2023.

[94] Y. Du, M. Yang, P. Florence, F. Xia, A. Wahid, B. Ichter, P. Sermanet, T. Yu, P. Abbeel, J. B. Tenenbaum *et al.*, "Video language planning," *arXiv preprint arXiv:2310.10625*, 2023.

[95] Y. Hong, H. Zhen, P. Chen, S. Zheng, Y. Du, Z. Chen, and C. Gan, "3d-llm: Injecting the 3d world into large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 20 482–20 494, 2023.

[96] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "Spatialvlm: Endowing vision-language models with spatial reasoning capabilities," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 455–14 465.

[97] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[98] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[99] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," *arXiv preprint arXiv:2309.02561*, 2023.

[100] Y. Wang, T.-H. Wang, J. Mao, M. Hagenow, and J. Shah, "Grounding language plans in demonstrations through counterfactual perturbations," *arXiv preprint arXiv:2403.17124*, 2024.

[101] J. Hsu, J. Mao, and J. Wu, "Ns3d: Neuro-symbolic grounding of 3d objects and relations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 2614–2623.

[102] J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 462–12 469.

[103] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, "Robopoint: A vision-language model for spatial affordance prediction for robotics," *arXiv preprint arXiv:2406.10721*, 2024.

[104] J. Duan, W. Yuan, W. Pumacay, Y. R. Wang, K. Ehsani, D. Fox, and R. Krishna, "Manipulate-anything: Automating real-world robots using vision-language models," *arXiv preprint arXiv:2406.18915*, 2024.

[105] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, "Zero-shot text-to-image generation," in *International conference on machine learning*. Pmlr, 2021, pp. 8821–8831.

[106] J. Li, D. Li, C. Xiong, and S. Hoi, "Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation," in *International conference on machine learning*. PMLR, 2022, pp. 12 888–12 900.

[107] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[108] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[109] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *Advances in neural information processing systems*, vol. 33, pp. 9912–9924, 2020.

[110] A. Baevski, W.-N. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "Data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning*. PMLR, 2022, pp. 1298–1312.

[111] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[112] T. Darcet, M. Oquab, J. Mairal, and P. Bojanowski, "Vision transformers need registers," *arXiv preprint arXiv:2309.16588*, 2023.

[113] X. Wang, J. Yang, and T. Darrell, "Segment anything without supervision," 2024. [Online]. Available: https://arxiv.org/abs/2406.20081

[114] Y. Wang, Z. Li, M. Zhang, K. Driggs-Campbell, J. Wu, L. Fei-Fei, and Y. Li, "D^{3}fields: Dynamic 3d descriptor fields for zero-shot generalizable robotic manipulation," *arXiv preprint arXiv:2309.16118*, 2023.

[115] Z. Ma, Y. Yue, and G. Gkioxari, "Find any part in 3d," 2024. [Online]. Available: https://arxiv.org/abs/2411.13550

[116] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 80–93.

[117] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi, "Objaverse-xl: A universe of 10m+ 3d objects," *arXiv preprint arXiv:2307.05663*, 2023.

[118] Z. Bylinskii, T. Judd, A. Oliva, A. Torralba, and F. Durand, "What do different evaluation metrics tell us about saliency models?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 740–757, 2019.

[119] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, 2018.

[120] A. Goyal, J. Xu, Y. Guo, V. Blukis, Y.-W. Chao, and D. Fox, "Rvt: Robotic view transformer for 3d object manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 694–710.

[121] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox, "Rvt-2: Learning precise manipulation from few demonstrations," *arXiv preprint arXiv:2406.08545*, 2024.

[122] H. Zhang, F. Li, X. Zou, S. Liu, C. Li, J. Yang, and L. Zhang, "A simple framework for open-vocabulary segmentation and detection,"

in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 1020–1031.

[123] G. Li, V. Jampani, D. Sun, and L. Sevilla-Lara, "Locate: Localize and transfer object parts for weakly supervised affordance grounding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 922–10 931.

[124] S. Qian and D. F. Fouhey, "Understanding 3d object interaction from a single image," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[125] S. Qian, W. Chen, M. Bai, X. Zhou, Z. Tu, and L. E. Li, "Affordancellm: Grounding affordance from vision language models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7587–7597.

[126] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[127] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Proceedings of the 6th Conference on Robot Learning (CoRL)*, 2022.

[128] S. James and A. J. Davison, "Q-attention: Enabling efficient learning for vision-based robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1612–1619, 2022.

[129] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13 739–13 748.

[130] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[131] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," *6th Annual Conference on Robot Learning*, 2022.

Appendix

A. Details on Unsupervised Affordance Annotation Extraction Pipeline (Sec III-A)

Implementation Details on Fine-Grained Region Proposal. In this section we discuss additional implementation details to obtain candidate regions.

From multi-view RGBD renderings of an asset, we obtain aggregated point cloud $P \in \mathbb{R}^{N\times 3}$ by projecting fore-ground pixels (we obtain this mask from the rendering simulator) of each view to world frame and uniformly down-sample the point cloud. Then for each RGB image, we extract patch-wise features from DINOv2 with registers (ViT-L14) [14] and perform bilinear interpolation to upsample the features to original image size.

To fuse the DINOv2 features from all views to P, we adapt the following procedure from [114]: for each point $p \in P$, we compute it's corresponding pixel on each camera view. We consider it to be visible in a camera view if the projection depth is close to the depth image reading at that pixel by a small threshold. The fused feature for p is the average of DINOv2 features on it's corresponding pixel across all the views p is visible.

With the procedure above we obtain global feature field $F_{\text{global}} \in \mathbb{R}^{N\times d}$, and then we apply PCA to obtain $F_{\text{reduced}} \in \mathbb{R}^{N\times 3}$ to mitigates affect of local texture or appearance on cluster result.

We group object points into candidate regions by running clustering algorithm on F_{reduced}. Since our data processing pipeline handles over-segmentation better than under-segmentation (reasons established in next su-bsection), we first run Mean Shift on the features, and if it found less than 5 clusters over the object, we re-run the k-means to find 5 clusters. Specifically, for articulated objects such as cabinets, we obtain the per-link mask from the rendering process and run the aforementioned clustering pipeline for each link individually. This allows us to find finer object regions such as drawer knobs.

Implementation Details on Task Instruction Proposal and Region-Instruction Mapping. After we obtain the region labels for all object points, we visualize the clustering on the view we selected with the following procedure: for each foreground pixel in that view, we compute the corresponding 3D point with using the depth map. Then we find its closest neighbor p in the aggregated pointcloud P, and use the label r_p as the region label for this pixel.

We assign a unique color to all pixels within each cluster and overlay this on the original RGB image. We input the cluster visualization with the original image to the VLM prompt below. The prompt contains only generic instructions and a few text-based examples to illustrate expected output and format. VLM is queried to propose a set of task instructions $\{\mathcal{T}_1, ..., \mathcal{T}_J\}$ closely related to the object, and associate each instruction with a single candidate region. We use GPT-4o [10] for all our experiments.

To convert the instruction-region matching to continuous affordance map $A \in [0,1]^{H\times W}$, we average the features for points in the corresponding region and calculate cosine similarity score of this reference feature and F_{global}. We project this to each camera view following the same procedure as we visualize cluster results, i.e. for each pixel, find its corresponding 3D point's nearest neighbor in P and assign that point's value. All values below 0 are set as 0 to ensure the correct value range of the obtained affordance map.

We found over-segmentation by the clustering step is preferred over under-segmentation. When a object part is over-segmented, empirically GPT-4o is still capable for correctly associating the instruction with one of the regions, and through the cosine similarity calculation, the other not selected regions within the same part is likely still computed to have high cosine similarity to the reference feature. On the other hand, under-segmentation could cause the affordance map to highlight regions that are not most closely related to the instruction, which is not desired for our purpose of finding fine-grained affordance.

We obtain triplets of RGB object image, task instruction, and affordance map (I, $\mathcal{T}$, A), from our data extraction pipeline. We further process the affordance map by setting all values below a threshold 0.5 to 0, with the purpose to create a ground-truth map more focused on the most relevant regions. We then apply a Gaussian blur to A with kernel size = 3, to accommodate for the boundaries created by the previous thresholding process and improve training stability.

Our model contains 3 FiLM-conditioned convolution layers with output channels $[256, 64, 1]$. We use linear layers to predict channel transformations from language embeddings. We initialize the linear layers with weights to be 1 and bias to be 0, adapted from implementation in RT-1 [126]. We use Adam optimizer with a learning rate of 0.001. We train our model for 30 epochs with a batch size 8, which takes approximately 12 hours on a single NVIDIA A6000 GPU.

B. Details on Task-conditioned Affordance Prediction (Sec IV-A)

Mturk Interface. Fig 7 is the Amazon MTurk interface we use to collect human affordance annotations on our valuation sets and DROID images.

MTurk Task Assignments and Label Post-processing. To obtain the ground-truth for evaluations, we prompt the image and corresponding text to Amazon MTurk workers and ask them to draw a fine-grained mask on the image for the region they believe corresponds to the text. On average, worker complete labelling assignment for each image in 40 seconds, for which they are compensated for 0.6 dollars.

For each (text, image) query pair, we collect annotations from 7 MTurk workers and apply a pixel-wise voting scheme. A pixel is marked as part of the ground truth mask (value 1) if more than three workers label it accordingly.

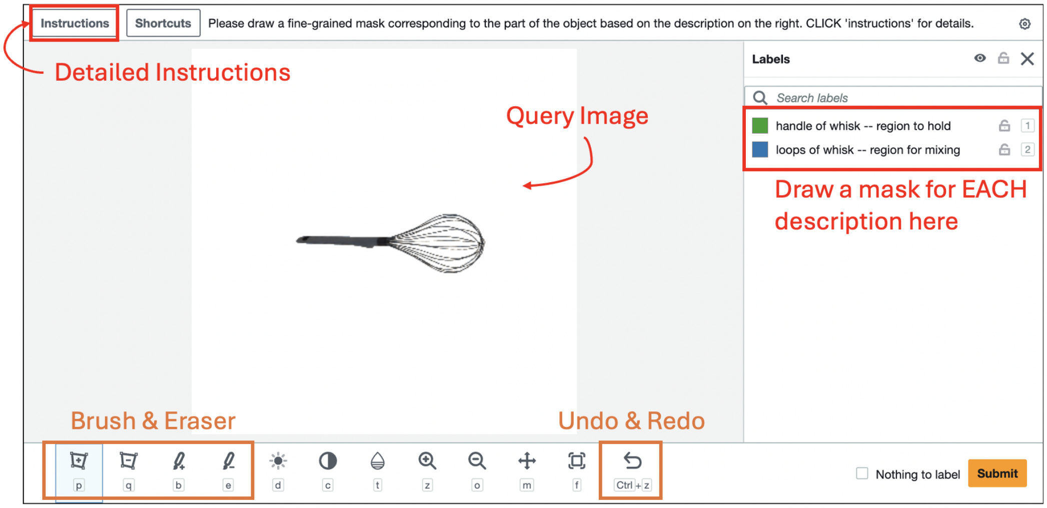
Details for Evaluation on AGD20K Dataset. We evaluate our model on the easy split of AGD20K and compare with the baseline performance reported in [125]. To avoid numerical instability in KLD computation as we consider per-pixel affordance instead of per-image affordance as in

```
## System Prompt

Given (a) the category of an object, (b) a reference image of the original image, (c) an image
    visualizing clustering of the object into a few regions, each indicated by a distinct color, and (d)
    a related list of used colors, please:
1. Identify specific regions of the object that serve different purposes in various manipulation tasks.
- Focus on crucial parts and offer detailed and fine-grained descriptions of the regions of interest.
- For each identified region, provide both a Region Description and a Region for action xxx. For example,
     "handle of plastic bag -- region for agent to hold and lift the bag." Use double quotes only to
    represent a string element.
- Avoid using multiple single quotes for an element. For example, instead of "adjusting the lamp's
    position", use "adjusting the lamp's position." Avoid trivial regions like: power cord, power plug,
    seal, small edges, small corners, different sides of the walls or body, interior base, or exterior
    edge.
2. Match the colored region in (c) with the proposed task in step 1, considering the functionality and
    the granularity of the task. The requirements are as follows:
- Compare the original image to the proposals to find the colored region that matches the description,
    considering the context provided by the explanation. For example, if a given proposal image indicates
     that the red region covers the handle, and the description mentions a task related to the handle,
    you should identify the answer as "Red."
- When the described region is clustered into more than one cluster on the image, pick the cluster that
    is most appropriate according to the explanation provided in the description.
- If the described region is within a cluster but still contains other parts of the object, you should
    still select the cluster.
- Consider the reason/explanation to make the final decision, and provide your best guess.
If you cannot identify the counterpart on the given image, give your best guess. Do not say you cannot
    identify something.

## User Prompt

The first image is the original image of the object, and the second image shows the clustering of regions
     in colors.
This is the color list: {CLUSTER_COLORS}, and this is the object category:{OBJ_CATEGORY}.
I need you to propose the task-guided fine-grained region description and match region description with
    the one most appropriate color in the images.
Output format: Start with the word "ANSWER: ", followed by a dict, where each key-value pair is in the
    format of "region description -- region for xxx" : "Color", separated by commas. Specifically, the
    content after "ANSWER: " should be parseable with Python's ast.literal_eval() and nothing else. All
    elements in the dict keys or values should be enclosed by double quotes only. The color could only be
     one color, with the first letter of the color name capitalized and the rest in lowercase.
```



Please draw a fine-grained mask corresponding to the part of the object based on the description on the right. CLICK 'instructions' for details.

🟩 handle of whisk -- region to hold 1
🟦 loops of whisk -- region for mixing 2

Fig. 7: Amazon MTurk Annotation Interface.

other works, we post-process our model's prediction by adding a small ϵ to each pixel before normalization.

Details for Evaluation on DROID Images. We implement the baselines as follows:

- CLIP: we obtain the per-patch visual feature of query image and use bilinear interpolation to original image height and width. We compute the per-pixel cosine similarity with the text feature of query instruction and

clip the minimum value to be 0.
- OpenSeeD: Since OpenSeeD is an open-vocabulary segmentation model, we use the same query instructions as for other methods to query the predicted mask. We set all pixels in the predicted object mask to 1 and 0 otherwise. The prediction image is all 0 if no mask is found.

C. Policy Learning in Simulation Details (Sec IV-B)

Environment Setup. Our simulation environment in OmniGibson contains one Fetch robot, for which we use operational space controller for the end-effector pose, multi-finger gripper controller for the gripper, and kept location of the robot base fixed. Grasping is physically simulated for all tasks.

We use a key-frame based policy for both demonstrations and learnt policies. Key-frames are commonly used by prior works [120, 121, 127–129] as "important or bottleneck steps of gripper during task execution". To execute an action <end-effector pose, gripper action>, we first command the end-effector controller of an interpolated trajectory from current to target pose, then execute the gripper action afterwards.

We use 3 cameras at the front, left, and right of the workspace for *Pouring* and *Inserting*. We use 2 cameras on both sides of the robot for *Opening* as the articulated objects are typically large in size and would occlude the other cameras.

Tasks. Below we discuss the details of environment setup, scripted policy steps, success criteria, and evaluation generalization setting for each task.

Pouring The environment includes a beer bottle, a bowl, and a pot plant. The scripted policy involves four key-frames: reaching a pre-grasp pose next to the bottle, grasping the bottle, lifting and moving it next to the bowl, and tilting it to pour into the bowl. Success is defined by the alignment and tilting of the bottle's opening directly over the bowl.

At training time, object poses are randomized within a [±5cm, ±3cm, 0] range, with the bowl and the pot plant positions randomly swapped. This randomization is maintained during evaluations to test the system with varied object poses.

Different object models for the beer bottle and bowl are used for the novel object instance evaluation. The beer bottle is replaced with a Coke can in the novel object category evaluation. For the novel instruction, the task is changed to watering the pot plant.

Opening The task environment features a cabinet with a revolute door. The task sequence includes two steps: reaching and grasping the cabinet door handle, followed by pulling it open. The task is considered successful if the door opens to at least 45 degrees.

During training, the position and orientation of the cabinet are randomized within a range of [±5cm, ±5cm, 0] for position, and ±15 degrees around the z-axis for rotation. This randomization is also applied during evaluations to assess performance with varied object poses.

For the novel object instance evaluation, a different cabinet model is used. A small refrigerator substitutes the cabinet in the novel object category setting, testing adaptability to different objects. Novel instruction scenarios are not evaluated for this task.

Insertion The environment contains a marker, a carrot, and a pencil holder. The task involves two key steps: picking up the marker and positioning it directly above the pencil holder's opening in an upright orientation. The task is considered successful if the marker is in the holder.

During training, the positions of the pen and carrot are randomized within ±1.5 cm in the x-direction, and the pencil holder is adjusted within ±3 cm in both x and y directions. The pen and carrot positions are also randomly swapped. The same randomization parameters are used during evaluation.

A different marker model, varying in color and size, is used for evaluating a new object instance. The pencil holder is replaced with a coffee cup for the novel object category evaluation. The task of inserting the pen is changed to inserting the carrot for the novel instruction evaluation.

Details on Baseline Visual Representations. Herein we introduce our implementation for each baseline visual representations.

- Vanilla policy: original rgb observation from each camera.
- w/ DINOv2: we first obtain per-pixel DINOv2 features for the rgb image of each camera. Then, we have a trainable 1D convolution layer with kernel size of 1 to reduce the number of channels to 3.
- w/ CLIP: we obtain CLIP text embedding for each detailed instruction for the task. Then we calculate the cosine similarity against per-pixel CLIP visual embedding of each camera observation.
- w/ Voltron [73]: we load a frozen Voltron (V-cond) model and obtain the visual embedding conditioned on task description. We interpolate the per-patch embedding to pixel space, and use a trainable 1D convolution layer with kernel size 1 to reduce the number of channels from 384 to 3. We have also experimented with using a trainable multi-head attention pooling layer for feature extraction, as suggested by the original paper, yet haven't observed improved performance.

Training Details We trained each policy for 4000 epochs. Training with batch size of 3 on a single NVIDIA A40 GPU takes approximately 16 hours for *Pouring*, and 8 hours for *Opening* and *Inserting*.

During training, we normalized the channels for visual observation by: normalize visual representation to $(-1, 1)$, clip (x, y, z) to the min and max workspace bounds, and set depth for all out-of-bound points to 0. Additionally, we append channels according to pixel location, following the original implementations in RVT. We apply random cropping augmentation to visual input during each training step.

D. Policy Learning in Real-World Details (Sec IV-C)

Environment Setup. Our real-world evaluation platform uses a Franka arm mounted in a tabletop setup built with Vention frames. Since the learned policy outputs 6-DoF end-effector poses and gripper actions, we use position control in all experiments, which is running at a fixed frequency of 20 Hz. Specifically, given a target end-effector pose in the world frame, we first clip the pose to the pre-defined workspace. Then we perform linear interpolation from the current pose of the robot to the target pose with a step size of 5mm for position and 1 degree for rotation. To move to each interpolated pose, we first calculate inverse kinematics to obtain the target joint positions based on current joint positions using the IK solver implemented in PyBullet [130]. Then we use the joint impedance controller from Deoxys [131] to reach to the target joint positions. Two RGB-D cameras, Orbbec Femto Bolt, are mounted on the left side and the right side of the robot facing the workspace center. The cameras capture RGB images and point clouds at a fixed frequency of 20 Hz.

Tasks. We mirror the setup in simulation to evaluate on three similar tasks in the real-world: watering plant, opening drawer, and inserting pen into pen holder. We collect a total of 10 demonstrations for each task and train a policy using the same training procedure described above. The demonstrations are collected using kinesthetic teaching, consisting of varying numbers of keyframes (as described above) that are required to complete the task. Success rates are visually examined by the operator. 10 trials with varying object configurations are performed, and the average success rate for each task is reported.

E. Case Study: Scaling to Objaverse-XL Assets

In this section, we describe an additional case study exploring the scalability of our pipeline to a more diverse set of 3D assets from Objaverse-XL [117].

- **Asset selection:** We selected objects from Objaverse-XL with LVIS category annotations, which represent common daily object categories. To ensure balanced representation across different categories, we randomly sampled 50 models from categories with abundant available models.
- **Asset filtering:** We filtered out assets with partially or fully transparent materials, as these produce incomplete depth renders required for accurate pointcloud reconstruction in our pipeline.
- **Asset rendering:** We rendered multi-view images of the assets using Blender with the official Objaverse codebase scripts. Assets where the rendering failed to fully capture the object were excluded from further processing.
- Following the asset preparation, we applied the same unsupervised affordance annotation extraction pipeline described in Section III-A to these Objaverse-XL assets.