

Self-Supervised Learning of Action Affordances as Interaction Modes

Liquan Wang[†], Nikita Dvornik^{*}, Rafael Dubeau[†], Mayank Mittal^{‡*}, Animesh Garg^{†*}

Abstract—When humans perform a task with an articulated object, they interact with the object only in a handful of ways, while the space of all possible interactions is nearly endless. This is because humans have prior knowledge about what interactions are likely to be successful, i.e., to open a new door we first try the handle. While learning such priors without supervision is easy for humans, it is notoriously hard for machines. In this work, we tackle unsupervised learning of priors of useful interactions with articulated objects, which we call *interaction modes*. In contrast to the prior art, we use no supervision or privileged information; we only assume access to the depth sensor in the simulator to learn the interaction modes. More precisely, we define a successful interaction as the one changing the visual environment substantially and learn a generative model of such interactions, that can be conditioned on the desired goal state of the object. In our experiments, we show that our model covers most of the human interaction modes, outperforms existing state-of-the-art methods for affordance learning, and can generalize to objects never seen during training. Additionally, we show promising results in the goal-conditional setup, where our model can be quickly fine-tuned to perform a given task. We show in the experiments that such affordance learning predicts interaction which covers most modes of interaction for the querying articulated object and can be fine-tuned to a goal-conditional model. For supplementary: <https://actaim.github.io/>.

I. INTRODUCTION

Humans demonstrate tremendous flexibility in operating objects around them. By leveraging *prior* experiences, we can adapt and manipulate new objects through careful interactions or exploration. A standard method in robotics for building object priors is by hand-crafting models based on our knowledge of an object’s relevant properties for a given task. For instance, various works in articulated object manipulation design modules to detect handles, obtain part-wise object segmentation, and estimate articulation parameters to define interaction plans [1–3]. The rigidity in these explicit models limits their ability to generalize and capture novel ways of changing the state of an unknown object, such as grasping its edge or pushing its surface. In this work, we aim to build representations that allow defining these *interaction modes* of an object implicitly, thereby providing *prior* knowledge for manipulating unseen objects. To do so, we focus on interacting with articulated objects with multiple moving parts as they do provide multiple affordances to be discovered.

Discovering *interaction modes* of an object has often been connected to the idea of *affordances* [4, 5], i.e., what does the object offer to an actor in the environment. For instance, a cabinet with three drawers has six possible interaction modes;

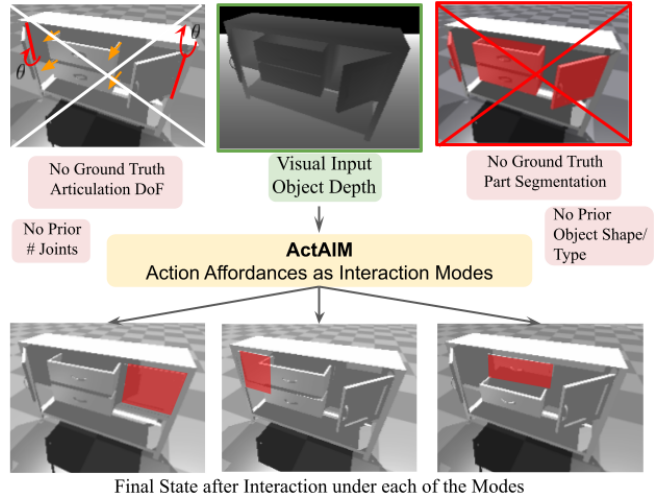


Fig. 1: ActAIM Overview. We propose a novel model architecture to learn action affordances of articulated objects using purely visual data. Without the use of any privileged ground-truth information or explicit supervision, our model learns distinct interaction modes that can generalize to new unseen objects.

each drawer providing the affordance to open or close it. Supervised learning approaches require collecting a large dataset of interactions, typically with a random policy, and labeling their effect as a change in the object’s state [6–8]. However, the sparsity of events that cause a noticeable change leads to a huge data imbalance and makes these approaches sample inefficient. Reinforcement learning (RL) methods avoid large-scale data annotation but suffer from a similar exploration issue. The policy tends to exploit only a limited region of the object for manipulation, thereby failing to discover all the interaction modes [9].

Besides the exploration issue in discovering interaction modes, existing works [3, 10–17] use the object’s state information directly as part of their observations, rewards function computation, or for scoring the amount of change caused by a particular action. However, humans primarily learn and act in partially observed settings. Relying solely on visual information exacerbates the learning problem since discriminating between interaction modes from images only (not using additional privileged information) is challenging.

In this paper, we present ActAIM (**A**ction **A**ffordances as **I**nteraction **M**odes) to overcome these issues by introducing the concept of interaction modes which can be clustered with the specific feature encoder and using only the visual observations during training. ActAIM discovers semantically meaningful & varied interaction modes and is also able to provide goal-conditional task completion. We use implicit geometry feature to build the semantic representation of

Correspondence to: garg@cs.toronto.edu

[†]University of Toronto & Vector Institute, [‡]ETH Zurich, ^{*}Nvidia, ^{*}Samsung.

the object instance which helps generalized across different categories. Our key contributions are as follows:

- 1) We propose the idea of interaction modes and a method to learn meaningful interactions in a self-supervised manner, from visual observations only.
- 2) We propose a new clustering-based data collecting strategy that increases the diversity among interaction modes in the collected data.
- 3) We experimentally show that ActAIM generates actions that cover a variety of ground-truth interaction modes and lead to successful goal completion, when conditioned on the goal observation (in the goal-conditional setup).

II. RELATED WORK

Affordance is an important concept in many fields. Based on the definition from Gibson *et al.* [4, 5], there has been a considerable amount of research on affordance in psychology, neuroscience, cognitive science, human-computer interaction, robotics, vision, and design. Researchers captured this general notion in various ways, including language, semantic segmentation, and key points.

Affordance in robot learning – Prior works have shown that learning to solve the manipulation problem could benefit from understanding the concepts of affordance. [18–21] focus on extracting affordance features using neural networks directly from image observation in supervised learning. Furthermore, semantic segmentation could be further extended to scenes with multi-object in [22]. Besides, affordance can be also defined as the probability of transition function representing the possibility of taking action in a certain area in [23]. [24] defines affordance with primitive actions and trains the agent to learn feasible action in different states which boosts the efficiency and scalability in performance. Inspired by the above papers, our work learns affordance from visual input by defining proper action primitives and trains the model without any supervision or privileged information from humans.

Semantic keypoint description – An explicit affordance representation is the keypoint representation [25–28]. Using the two-gripper robot, with the help of existing grasping algorithms such as [29, 30], keypoint representation can be used to guide robotic grasping in tool or articulated object manipulation tasks. For task-directed affordance, keypoint representation is sufficient [26, 27, 30] since the choice of interaction point is required to be limited and robotic moving trajectory is compatible to decompose into point movement. Considering the multi-mode interaction discovery task, we picked a certain keypoint as the interaction point conditional on the interaction modes combining with the dense local geometry feature following the ideas from [31–33].

Dense pixel-wise affordance – Existing dense pixel-wise affordance learning papers such as [6, 8, 34, 35] predict per-pixel affordance labels and query from these encodings. Among these papers, [6] and [36] enlightened us to combine implicit representation in articulated object manipulation. [36] solves the grasping task using the Convolutional Occupancy Network [37] implicit representation model and explores the synergy of using geometric features. But [36] is limited

to grasping tasks and relies on privileged information such as object meshes. [6] uses pure visual input to predict interaction points on articulated objects with predefined action primitive but requires fixed modes and part segmentation. Our work takes advantage of [36] by using the neural implicit representation to extract local geometry features to help generalize among different articulated objects.

III. PROBLEM FORMULATION

Inspired by the concept of prior knowledge in manipulation, we formulate the problem of discovery of object affordance in an unsupervised setting without access to privileged information. The goal is to build object-centric priors using only perception throughout the learning process, such that they facilitate: 1) realizing different types of *interaction modes* an object offers implicitly, and 2) capturing *where* and *how* to interact with an object for a given interaction mode.

ActAIM takes the depth image of an object as input and outputs the possible actions that can be executed with the object through interaction modes. In contrast to Where2Act [6] that discretizes the action space into six primitives, we consider a continuous action space for a parallel-jaw robotic gripper, $a = (\mathbf{p}, \mathbf{R}, \mathbf{F})$. The primitive action first reaches and attempts to grasp an interaction point \mathbf{p} over the visible articulated parts of the object P with an orientation $\mathbf{R} \in SO(3)$, and then moves a certain fixed distance along a unit direction $\mathbf{F} \in [-1, 1]^3$.

Formally, we obtain a prior distribution over possible interactions with an articulated object leading to a change in its state. Under a partially observed setting, this distribution can be denoted as $\mathbb{P}(a|o)$, where o is a visual observation of the articulated object, such as its depth image D , point cloud P or truncated signed distance field (TSDF) representation V . To discriminate between different interaction modes for an articulated object, we introduce a latent variable $z \in \mathcal{Z}$, and write the prior as:

$$\mathbb{P}(a|o) = \underbrace{\mathbb{P}(a|o, z)}_{\text{action predictor}} \underbrace{\mathbb{P}(z|o)}_{\text{mode selector}} \quad (1)$$

The distribution $\mathbb{P}(z|o)$ models the possible interaction modes (latent affordance) of an object given its current observation. For instance, a completely closed cabinet can only be opened, so only half of the maximum interaction modes associated with it are feasible. The distribution $\mathbb{P}(a|o, z)$ models the actions that would lead to the change associated with the interaction mode z .

IV. ACTAIM: LEARNING INTERACTION MODES

Our goal is to obtain the distributions in (1) without requiring explicit supervision labels or reward signals, which are typically computed using articulated objects’ joint state. To this end, we propose a self-supervised learning method that generates its own labels through interacting with an object (Sec. IV-A), and uses these to learn a common visual embedding for articulated objects that generalizes over unseen articulated object categories and instances (Sec. IV-B).

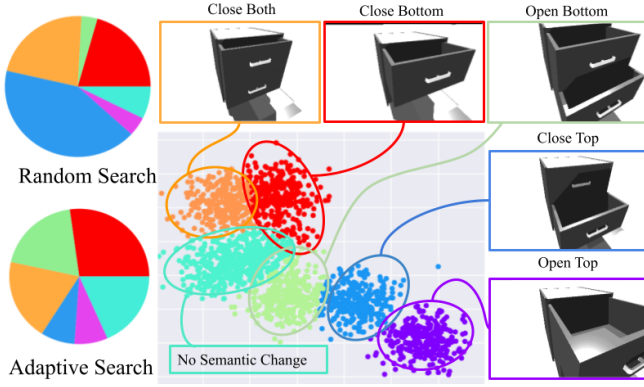


Fig. 2: Adaptive data collection using GMM. Visualization of using GMM to cluster different interaction modes. The clustering is presented with the t-SNE projection of the manifold. It illustrates that different interaction modes can be distinguished and clustered using simple image encoding. The two pie charts on the left show that using adaptive search for data collection increases the proportion of rare modes of interaction.

A. Adaptive data collection

We start the data collection by executing actions from the random policy in the simulator. For every executed action primitive, we store the tuple (o, a, o', \hat{y}) , where $o = (D_0, V)$ is the depth image and TSDF at the initial state of the articulated object, a is the executed action primitive parameters, $o' = (D_1)$ is the depth image of the articulated object after the interaction, and \hat{y} is the computed label determining whether the interaction is successful. We compute the TSDF before interaction using multi-view depth images [38].

To collect diverse interaction data, we need to vary object categories, instances, their initial states, camera views, and action primitive parameters a . Sampling actions with the random policy is sub-optimal due to the poor coverage of all possible interaction modes (e.g., it’s harder to randomly pull the handle than to push the door). Hence, we propose an adaptive scheme using unsupervised learning to improve data more balanced across different modes of interaction.

First, to get an embedding function for the depth images, we train an autoencoder, $\hat{D} = \mathcal{D}_D(\mathcal{E}_D(D))$, using an L_2 reconstruction loss. In the rest of the paper, we use the encoder representations, i.e., $\mathcal{E}_D(D)$, not the raw depth maps D , as subsequent inputs. Then, we fit a multi-modal distribution that clusters different interactions into, presumably, interaction modes. For this, we use the Gaussian Mixture Models (GMMs): $\mathbb{P}(a|D_0, \theta) = \sum_{k=1}^K \alpha_k p(a|D_0, \theta_k)$ where θ is the distribution parameters and K is the maximum number of mixtures. We fit this GMM iteratively for a single object instance with a fixed initial state. We define the effect of an executed action as, $\tilde{\tau} = \mathcal{E}_D(D_1) - \mathcal{E}_D(D_0)$. At the start, we collect interaction data of size M using a random policy. We fit GMM only on data that leads to a change in the embedded space: $\{(D_0, a, D_1), \|\tilde{\tau}\|_2 \geq \lambda\}$, where λ is a fixed threshold. We determine $\hat{y} = 1$ if $\|\tilde{\tau}\|_2 \geq \lambda$ otherwise $\hat{y} = 0$. For the following iterations, we sample ϵM interactions with a random policy and $(1 - \epsilon)M$ interactions from the GMM, and fit a GMM again over the collected data. As shown in Fig. 2, clustering using GMM discriminates among different

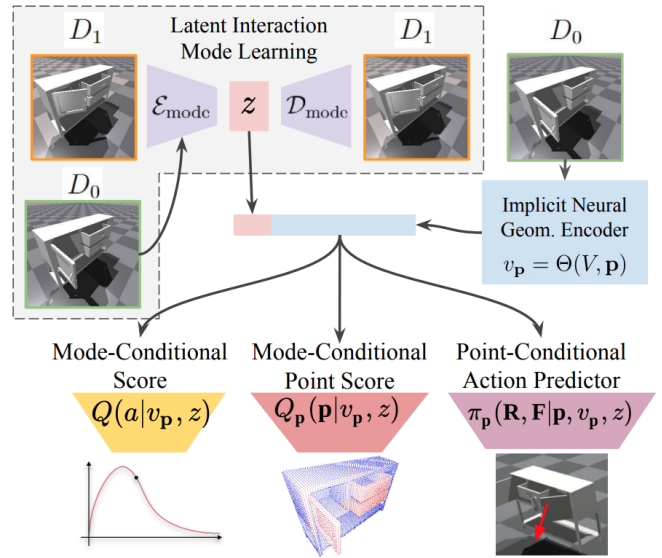


Fig. 3: ActAIM training overview. This figure shows the process of model training. During the training, ActAIM takes in the initial and final depth image observations D_0 and D_1 . D_0 and D_1 are passed into the encoder $\mathcal{E}_{\text{mode}}$ to produce the mode latent z , and D_0 is passed into the implicit neural geometry encoder to produce a local geometry feature $v_{\mathbf{p}}$. Mode latent z and local feature $v_{\mathbf{p}}$ are passed into the mode-conditional score function, mode-conditional point score, and point-conditional action predictor to predict scoring over the point cloud and the action \mathbf{R} and \mathbf{F} .

interaction modes and this adaptive sampling strategy yields a more balanced coverage of interaction modes compared to a random policy.

B. Self-supervised model for interaction discovery

While the GMM approach described for data generation clusters different interaction modes, it only does so for single object instance at a specific state. Instead, we want a model that generalizes across poses, instances and categories of objects. Given that $a = (\mathbf{p}, \mathbf{R}, \mathbf{F})$, we can redefine (1) as:

$$\mathbb{P}(a|o, z) = \mathbb{P}_{\mathbf{R}, \mathbf{F}|\mathbf{p}}(\mathbf{R}, \mathbf{F}|\mathbf{p}, o, z) \mathbb{P}_{\mathbf{p}}(\mathbf{p}|o, z) \quad (2)$$

where $\mathbb{P}_{\mathbf{p}}$ defines the probability of selecting interaction point \mathbf{p} under an interaction mode z , and $\mathbb{P}_{\mathbf{R}, \mathbf{F}|\mathbf{p}}$ defines that for the gripper rotation and moving direction given a selected interaction point \mathbf{p} and interaction mode z . This decomposition of the action predictor helps reducing the sampling complexity during inference. Our implementation of this decomposed pipeline is given in Fig. 3.

1) Mode selector $\mathbb{P}(z|o)$

We model the mode selector $\mathbb{P}(z|o)$ with a Conditional Variational Autoencoder (CVAE) [39] with encoder $\mathcal{E}_{\text{mode}}$, decoder $\mathcal{D}_{\text{mode}}$. This autoencoder operates on the embeddings of the depth images, $(\mathcal{E}_D(D_0), \mathcal{E}_D(D_1))$, as described above, to generate the latent interaction mode z . Latent interaction mode z captures the difference between D_0 and D_1 which represents the state change. The decoder $\mathcal{D}_{\text{mode}}$ takes in the conditional variable $\mathcal{E}_D(D_0)$ and the latent interaction mode z to reconstruct $\mathcal{E}_D(D_1)$. We train this CVAE structure together with the further model and optimize it with the regularization loss and reconstruction loss.

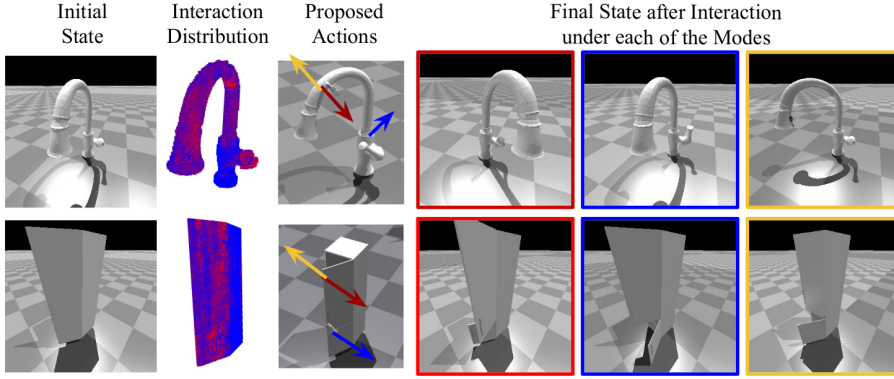


Fig. 4: Generative model object manipulation results: ActAIM takes in the initial state and predicts the interaction distribution based on the interaction point scoring model. The distribution shows the valid part (colored in red) to interact with. We sample interaction point based on the point score and execute the predicted action. The action will lead to different interaction mode as shown.

2) Implicit Neural Geometry Encoder

We use an implicit neural geometry encoder that encodes local geometry features to improve the generalizability of the model over different categories of articulated objects. Implicit representation is a continuous function with neural network as the input. We formalize local geometry feature extraction as $v_{\mathbf{p}} = \Theta(V, \mathbf{p})$ where V is the TSDF of the object and \mathbf{p} is a queried point. The structure of the feature extraction network, Θ , is adapted from Convolutional Occupancy network [37] (ConvOnet). The ConvOnet decoder conveys the core idea of implicit representation which provide memory-sufficient storage of point-wised feature data. Consider the plane feature $\Omega = \{\Omega_{xy}, \Omega_{yz}, \Omega_{zx}\}$ from the ConvOnet, we extract the point specific feature $v_{\mathbf{p}}$ given the querying point \mathbf{p} . Following [36], we concatenate features using querying point projected on the corresponding plane and perform bilinear interpolation ϕ around the neighborhood of the querying point \mathbf{p} . We represent our local feature extraction as,

$$v_{\mathbf{p}} = [\phi(\Omega_{xy}, \mathbf{p}_{xy}), \phi(\Omega_{yz}, \mathbf{p}_{yz}), \phi(\Omega_{zx}, \mathbf{p}_{zx})] \quad (3)$$

Where \mathbf{p}_{xy} denotes the x and y coordinate of the point \mathbf{p} . We combined the action predictor with this Implicit Neural Geometry Encoding and express the action predictor as,

$$\mathbb{P}(a|o, z) = \mathbb{P}_{\mathbf{R}, \mathbf{F}|\mathbf{p}}(\mathbf{R}, \mathbf{F}|\mathbf{p}, v_{\mathbf{p}}, z) \mathbb{P}_{\mathbf{p}}(\mathbf{p}, v_{\mathbf{p}}|o, z) \quad (4)$$

C. Training procedure

Through the above formulation, we want to jointly train the distributions ($\mathbb{P}_{\mathbf{p}}, \mathbb{P}_{\mathbf{R}, \mathbf{F}|\mathbf{p}}$), the CVAE ($\mathcal{E}_{\text{mode}}, \mathcal{D}_{\text{mode}}$), and the neural geometry encoder (Θ).

Mode-conditional score function $Q(a|o, z)$ – We regard the distribution $\mathbb{P}(a|o, z)$ as a score function $y = Q(a|v_{\mathbf{p}}, z)$ to denote the probability of success when taking action a with the local geometry feature $v_{\mathbf{p}}$. We use the data collected from GMM represented as $\{D_0, D_1, a, \hat{y}\}_i$ and z from the mode selector CVAE model. We optimize this module using the binary cross entropy loss with self-supervised label \hat{y} from the dataset. The predicted mode conditional critic will be used to help evaluate the point score function later.

Mode-conditional point score function $Q_{\mathbf{p}}(\mathbf{p}|o, z)$ – We model the distribution $\mathbb{P}_{\mathbf{p}}$ (of successful interaction at point p) as a score function $y_{\mathbf{p}} = Q_{\mathbf{p}}(\mathbf{p}|v_{\mathbf{p}}, z)$, taking the local geometric feature $v_{\mathbf{p}}$ as input. We train $Q_{\mathbf{p}}$ using the data $\{D_0, a\}_i$ and the computed z , and optimize with binary cross

entropy loss. The ground-truth $\hat{y}_{\mathbf{p}}$ referring the probability of point \mathbf{p} becoming the interaction point. To obtained the training signal $\hat{y}_{\mathbf{p}}$, we randomly sample N (100) rotations \mathbf{R}_i and moving directions \mathbf{F}_i and compute the label with mode conditional score function $Q(o, a|z)$ as followed,

$$\hat{y}_{\mathbf{p}} = \max\{Q(\hat{\mathbf{R}}_i, \hat{\mathbf{F}}_i, \mathbf{p}|o, z) | i = 1, \dots, N\} \quad (5)$$

We are using the maximum to express that the point is valid to interact with once there exists a successful interaction using this interaction point.

Point-conditional action predictor $\pi_{\mathbf{p}}(\mathbf{R}, \mathbf{F}|\mathbf{p}, o, z)$ – Given the interaction point \mathbf{p} sampled from the $Q_{\mathbf{p}}$, ActAIM predicts the rotation and moving direction with Point conditional action predictor $\pi_{\mathbf{p}}(\mathbf{R}, \mathbf{F}|\mathbf{p}, v_{\mathbf{p}}, z)$ together with the local implicit geometry feature. The module produces rotation \mathbf{R} and moving direction \mathbf{F} directly and optimized with collected data $\{D_0, D_1, a\}_i$. Denoting the ground-truth rotation and moving direction as $\hat{\mathbf{R}}$ and $\hat{\mathbf{F}}$, the loss can be written as

$$\mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\mathbf{F}} = (\mathbf{F} - \hat{\mathbf{F}})^2 + (1 - |\mathbf{R} \cdot \hat{\mathbf{R}}|) \quad (6)$$

Final Loss – The complete training loss is now denoted as

$$\mathcal{L} = \mathcal{L}_{\text{CVAE}} + \mathcal{L}_Q + \mathcal{L}_{Q_{\mathbf{p}}} + \mathcal{L}_{\mathbf{R}} + \mathcal{L}_{\mathbf{F}} \quad (7)$$

D. Goal-conditional inference

ActAIM can be used for goal-conditional inference, by providing the desired goal observation D_1 as an extra input. To do so, we replace the generative model with a deterministic one and fine-tune the system. We treated D_1 as the conditional variable g and re-format the training as followed,

$$\tilde{\mathbb{P}}(a|s, g) = \underbrace{\mathbb{P}(a|o, z, g)}_{\text{action predictor}} \underbrace{\tilde{\mathbb{P}}(z|o, g)}_{\text{goal-conditional mode selector}} \quad (8)$$

During the training, we keep the same action predictor and only fine-tune the mode selector $\mathbb{P}(z|o, g)$ to be goal-conditional. This goal conditional mode selector produces the corresponding interaction mode latent z for action predictor which in turn proposes action a leading to goal g .

V. EXPERIMENTS

Our experiments aim to evaluate the proposed method, ActAIM, in terms of: 1) the ability to capture diverse interaction modes across varying object instances and categories, 2) its performance on unseen object states and instances

TABLE I: Self-Supervised Affordance Mode Discovery: We evaluate our design choices with baseline and ablation study using the metrics of sample-success rate, weighted modes ratio and normalized conditional entropy. We also illustrated an extra column which is the average of the section of articulated objects. In each column section, we bold the best numbers and show that our model outperforms most of the time. Categories: faucet, table, cabinet, door, window, fridge, kitchen pot, kettle, switch

Test Set	Unseen States of Training Objects							Unseen instances of Training Categories							Unseen Categories			
Sample Success Rate % \uparrow																		
Random Interaction	5.61	6.05	8.47	14.59	18.15	5.66	9.76	7.99	3.38	6.47	13.86	20.92	4.51	9.52	13.32	10.42	6.04	9.93
Where2Act [6]	33.32	7.05	7.05	17.88	11.64	4.07	13.50	32.99	13.78	6.94	18.89	15.00	5.42	15.50	18.43	9.49	4.14	5.34
ActAIM-PN++	41.25	44.92	32.46	21.64	46.27	19.52	34.34	21.04	31.21	31.91	19.21	36.14	12.43	25.32	21.37	18.34	21.61	20.44
ActAIM [ours]	49.26	41.36	36.21	28.64	58.31	19.68	38.91	21.98	38.10	35.54	21.03	41.61	16.19	29.07	21.09	24.68	24.13	23.30
Weighted Modes Ratio % \uparrow																		
Random Interaction	5.61	5.27	7.62	12.77	15.61	5.26	8.69	4.47	3.12	6.13	9.73	16.72	3.92	7.35	13.32	10.23	5.87	9.81
Where2Act [6]	11.77	6.06	6.25	14.50	8.59	3.51	8.44	10.86	9.71	6.02	11.00	10.63	5.17	8.89	18.43	8.81	3.91	5.19
ActAIM-PN++	29.11	25.52	20.48	12.81	45.54	17.48	25.16	14.28	24.18	26.66	17.29	25.92	10.44	19.79	18.51	15.76	16.09	16.79
ActAIM [ours]	39.20	36.49	25.56	18.76	57.42	17.29	32.45	15.12	34.58	32.55	18.90	33.21	14.56	24.82	18.92	17.68	17.31	17.97
Weighted Normalized Entropy % \uparrow																		
Random Interaction	5.19	4.45	6.80	10.49	10.41	4.18	6.92	7.09	2.82	4.89	5.74	11.30	3.01	5.81	10.02	7.41	5.79	7.74
Where2Act [6]	12.12	5.08	5.41	9.03	5.15	3.23	6.67	15.62	8.31	5.93	6.75	5.30	4.23	7.68	17.84	7.60	3.91	4.89
ActAIM-PN++	24.60	38.28	28.48	17.85	32.66	8.74	25.10	6.51	13.02	16.43	7.22	14.76	6.00	10.66	15.78	12.34	12.40	13.51
ActAIM [ours]	34.79	36.49	35.64	25.48	41.76	9.31	30.58	7.14	19.38	24.15	7.77	19.27	8.16	14.31	16.31	16.17	15.58	16.02

from known categories, 3) its generalization on objects from unknown categories, and 4) the utility of the learned priors for goal-conditioned behaviors.

A. Experimental setup

Following [6], we use articulated objects from the SAPIEN dataset [40]. For training (seen categories), we pick nine categories: faucet, table, storage furniture, door, window, refrigerator, box, trashbin, and safe. For testing (unseen categories), we pick the 3 extra categories: kitchen pot, kettle, and switch. For each category, we use 8 object instances with 4 different initial states for training and testing. We use IsaacGym [41] simulator to collect interaction data using a floating Franka parallel-jaw gripper. For collecting depth images, we vary the view angle of the camera in front of the object between $\{-45^\circ, -22.5^\circ, 0^\circ, 22.5^\circ, 45^\circ\}$. To obtain the TSDF of the articulated object, we only use the given single depth image to compute the TSDF using [38] during testing since we have found that using fewer cameras to reconstruct TSDF does not affect the testing results.

Evaluation Metrics – To evaluate the multi-modal interaction modes, we use the following metrics to evaluate the prior distribution $\mathbb{P}(a|o)$:

1) **sample-success-rate (ssr)** – measures the fraction of proposed interaction trials which are successful [6],

$$ssr = \frac{\# \text{ successful proposed interaction}}{\# \text{ total proposed interaction}}. \quad (9)$$

2) **weighted modes ratio (η)** – measures the success rate weighted by the fraction of the interaction modes discovered,

$$\eta = ssr \times \frac{\# \text{ successful discovered mode}}{\# \text{ total GT modes}}. \quad (10)$$

3) **weighted normalized entropy ($\bar{\mathcal{H}}$)** – measures the success rate weighted by the entropy of the distribution.

$$\bar{\mathcal{H}} = ssr \times \frac{\mathcal{H}(\mathcal{M})}{\mathcal{H}_{max}}, \quad (11)$$

where entropy $\mathcal{H}(\mathcal{M}) = -\sum_{m \in \mathcal{M}} p(m) \log p(m)$ is computed using $p(m)$, which is the percentage of sampled interactions leading to mode m . The maximum entropy \mathcal{H}_{max} is computed under the condition of equally distributed proposed interaction modes. Intuitively, for a more balanced prior distribution to sample rarer modes, $\bar{\mathcal{H}}$ should be higher since it covers possible interaction modes equally.

We use the ground-truth articulation state and part information to compute these metrics. We label an interaction successful when any object’s DoF changes by at least 10%. Additionally, the weighted metrics are computed by verifying if interaction triggers a possible ground-truth interaction mode of the articulated object.

Baselines for comparisons and ablations – The proposed problem formulation in III is similar to skill-discovery in unsupervised RL (URL) [42]. While the formulation is similar, URL is still to be shown effective in high-dimensional partially observed settings. In fact, we found that URL baselines including [43–46] performed poorer than a random policy and failed to express the complicated interaction modes. Another approach such as UMPNet [9] looks at articulated object interaction in a goal-conditioned setting and relies on the groundtruth articulation state and part segmentation during training and inference. This is in stark contrast to our goal of learning priors without such privileged information.

Thus, we compare our approach to the following baselines:

- 1) **Random policy:** uniformly samples interaction points from the articulated object’s point cloud, orientation and moving direction.
- 2) **Where2Act [6]:** computes priors per discretized action primitive. Since it evaluates interaction modes depending on separate primitives, we aggregate the push and pull interactions and compute the average from these modes. Different from [6], we provide the whole object point cloud to the model instead of the segmented movable points as mentioned in the paper.
- 3) **ActAIM-PN++:** is a version of ActAIM that computes

TABLE II: Goal Conditional Evaluation We evaluate our goal conditional model in terms of the success rate of reaching to the provided goal. We compare our model to Where2Act under 2 different evaluation tasks and bold the best number in each column section. Categories: faucet, table, cabinet, door, window, fridge, box, trash_can, safe

Test Set		Unseen States of Training Objects							Unseen instances of Training Categories						Unseen Categories				
Eval Task	Sample Success Rate % \uparrow																		
Dec-DoF	Where2Act-Push	26.54	21.12	4.02	23.77	20.27	5.17	16.82	10.12	15.54	15.13	43.04	9.42	6.63	16.64	2.52	23.90	43.52	23.31
(Common)	ActAIM [ours]	25.32	37.45	19.31	62.91	67.32	61.23	45.59	20.31	36.31	18.24	41.21	29.31	31.42	29.47	15.17	22.50	32.51	23.39
Inc-DoF	Where2Act-Pull	12.52	0.27	0.42	0.02	0.98	0.06	2.38	0.00	1.56	0.00	0.04	0.46	0.00	0.34	2.79	0.06	37.52	13.46
(Rare Mode)	ActAIM [ours]	24.15	16.21	11.34	28.14	49.31	17.56	24.45	12.41	14.25	9.48	10.46	15.98	13.12	12.62	7.84	13.12	21.41	14.12

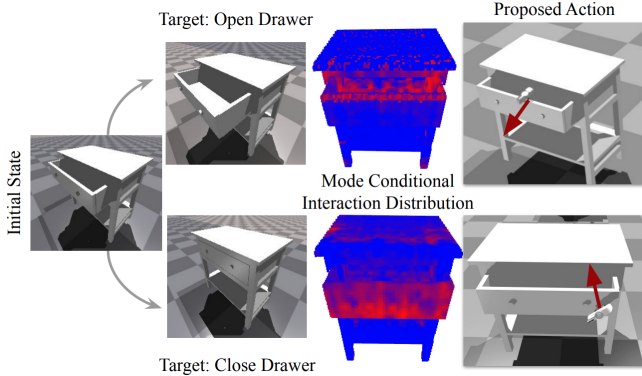


Fig. 5: Goal conditional manipulation results: ActAIM takes the and the target images to generate the mode conditional interaction distribution. Such distribution illustrates the valid part (colored in red) to manipulate. We sample the interaction point from pure point cloud based on the mode conditional point score and predict the action. The execution of the proposed action leads the initial state to the target state.

point features v_p using PointNet++ [47] instead of Con-voNet.

B. Goal conditional experiments

As described in Sec. IV-D, the prior distribution can be used to infer goal-conditioned behaviors. After training the generative model in ActAIM, we swap the input to the CVAE with a goal conditional input and fine-tune the trained model. The goal conditional experiments are set with an extra input, goal observation D_1 . We picked goal observation D_1 for each randomly selected object in a random pose and make sure that goal observation D_1 contains either the degrees-of-freedom increasing or decreasing. We evaluate the goal-conditional success using sample-success-rate ssr_{goal} , defined as

$$ssr_{goal} = \frac{\# \text{ proposed interaction leads to goal } D_1}{\# \text{ total proposed interaction}}. \quad (12)$$

We form the goals D_1 as increasing or decreasing object’s DoF (from D_0), and report the average results per goal. We consider Where2act [6] as a baseline for goal-conditional inference since it explicitly defines the motion of pushing and pulling.

C. Discussion of Results

Interaction mode discovery – We evaluate our self-supervised affordance mode discovery (i.e., non-goal-conditional inference) in Tab I. First, ActAIM leads to a significantly more frequent successful interaction than the baselines, despite the fact that Where2Act uses access to ground-truth interaction modes during training. Also, ActAIM discovers

more interaction modes than the baselines, based on the weighted normalized entropy. Importantly, the competing method often trades interaction mode diversity for success rate, yet, ActAIM leads at both metrics simultaneously. We attribute our success to our task formulation, the advanced data collection strategy, and a stronger generative model.

In Fig. 4, we qualitatively show that ActAIM discovers the correct interaction modes for different types of objects, even though it is trained without explicit knowledge of interaction modes. The distribution of interaction points provides a meaningful object segmentation into the movable part. Additionally, the interactions proposed by ActAIM lead to a meaningful change in the states, resulting in different valid and useful interaction modes.

Goal-conditional inference – To produce the goal-conditional interaction, we provide the model with an extra input O_1 , the visual observation of the goal state. As Table II shows, our ActAIM outperforms all the baselines on seen training categories. However, our results are slightly inferior to Where2Act [6] on the easier task *Dec-DoF* (i.e., pushing), which is likely due to more training signals in Where2Act. Yet, on the rate events (i.e., pulling) our method still does best even on unseen objects. Figure 5 visualized model outputs depending on the interaction goal. The goal-conditional interaction distribution shows how the modes of interaction change, depending on the given goal, which indicates that our method learned to map the goals to affordances. The proposed action illustrates the correct way to grasp or touch the object and move in a reasonable direction.

VI. CONCLUSIONS

We propose a self-supervised method for discovering action affordances as modes of interaction for articulated objects, from purely vision observations. ActAIM generalizes across different modes of interactions and different categories of articulated objects. Our method includes a novel adaptive data collection method, promoting interaction diversity, and a generative model to produce successful interactions with the objects, utilizing implicit object representations. Our results show that our model generates interactions with a high success rate over a wide range of interaction modes, and can generalize to unseen objects and categories.

VII. ACKNOWLEDGEMENT

We would like to thank Kaichun Mo for the helpful discussions and Vector Institute for providing computation resources. This work was supported in part by CIFAR AI Chair, NSERC Discovery Grant, and UofT XSeed Grant.

REFERENCES

- [1] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1289–1299.
- [2] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2751–2757.
- [3] M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, and A. Garg, "Articulated object interaction in unknown scenes with whole-body mobile manipulation," 2022.
- [4] J. J. Gibson, "The theory of affordances," in *Perceiving, acting, and knowing: toward an ecological psychology*. Hillsdale, N.J. : Lawrence Erlbaum Associates, 1977, pp. pp.67–82.
- [5] J. Gibson, *The Ecological Approach to Visual Perception*, ser. Resources for ecological psychology. Lawrence Erlbaum Associates, 1986.
- [6] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," 2021.
- [7] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong, "Vat-mart: Learning visual action trajectory proposals for manipulating 3d articulated objects," in *ICLR*, 2022.
- [8] 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)*, 2018, pp. 5882–5889.
- [9] Z. Xu, H. Zhanpeng, and S. Song, "Umpnet: Universal manipulation policy network for articulated objects," *IEEE RA-L*, 2022.
- [10] C. C. Kessens, J. Rice, D. Smith, S. Biggs, and R. Garcia, "Utilizing compliance to manipulate doors with unmodeled constraints," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 18-22, 2010, Taipei, Taiwan*. IEEE, 2010, pp. 483–489.
- [11] A. J. Schmid, N. Gorges, D. Goger, and H. Worn, "Opening a door with a humanoid robot using multi-sensory tactile feedback," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 285–291.
- [12] T. Harada, A. Tejero-de Pablos, S. Quer, and F. Savarese, "Service robots: A unified framework for detecting, opening and navigating through doors," in *Software Technologies*. Cham: Springer International Publishing, 2020, pp. 179–204.
- [13] T. Rühr, J. Sturm, D. Pangercic, M. Beetz, and D. Cremers, "A generalized framework for opening doors and drawers in kitchen environments," in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 3852–3858.
- [14] A. Jain and S. Niekum, "Learning hybrid object kinematics for efficient hierarchical planning under uncertainty," 2019.
- [15] A. Jain, R. Lioutikov, C. Chuck, and S. Niekum, "Screwnet: Category-independent articulation model estimation from depth images using screw theory," 2020.
- [16] B. Abbatematteo, S. Tellex, and G. Konidaris, "Learning to generalize kinematic models to novel objects," in *Proceedings of the Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 100. PMLR, 30 Oct–01 Nov 2020, pp. 1289–1299.
- [17] E. Klingbeil, A. Saxena, and A. Y. Ng, "Learning to open new doors," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2751–2757.
- [18] M. Stark, P. Lies, M. Zillich, J. Wyatt, and B. Schiele, "Functional object class detection based on learned affordance cues," in *Computer Vision Systems*, A. Gasteratos, M. Vincze, and J. K. Tsotsos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 435–444.
- [19] 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)*, 2015, pp. 1374–1381.
- [20] H. Kjellström, J. Romero, and D. Kragić, "Visual object-action recognition: Inferring object affordances from human demonstration," *Computer Vision and Image Understanding*, vol. 115, no. 1, pp. 81–90, 2011. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S107731421000175X>
- [21] H. O. Song, M. Fritz, D. Goehring, and T. Darrell, "Learning to detect visual grasp affordance," *IEEE Transactions on Automation Science and Engineering*, vol. 13, no. 2, pp. 798–809, 2016.
- [22] T. Nagarajan and K. Grauman, "Learning affordance landscapes for interaction exploration in 3d environments," 2020. [Online]. Available: <https://arxiv.org/abs/2008.09241>
- [23] K. Khetarpal, Z. Ahmed, G. Comanici, D. Abel, and D. Precup, "What can i do here? a theory of affordances in reinforcement learning," 2020. [Online]. Available: <https://arxiv.org/abs/2006.15085>
- [24] A. Manoury, S. M. Nguyen, and C. Buche, "Hierarchical affordance discovery using intrinsic motivation," in *Proceedings of the 7th International Conference on Human-Agent Interaction*, ser. HAI '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 186–193. [Online]. Available: <https://doi.org/10.1145/3349537.3351898>
- [25] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpm: Keypoint affordances for category-level robotic manipulation," 2019.
- [26] D. Turpin, L. Wang, S. Tsogkas, S. Dickinson, and A. Garg, "Gift: Generalizable interaction-aware functional tool affordances without labels," 2021.
- [27] Z. Qin, K. Fang, Y. Zhu, L. Fei-Fei, and S. Savarese, "Keto: Learning keypoint representations for tool manipulation," 2019.
- [28] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," 2018.
- [29] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox, "Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes," 2021.
- [30] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017.
- [31] K. Khetarpal, Z. Ahmed, G. Comanici, and D. Precup, "Temporally abstract partial models," 2021.
- [32] K. Khetarpal, Z. Ahmed, G. Comanici, D. Abel, and D. Precup, "What can i do here? a theory of affordances in reinforcement learning," 2020.
- [33] 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)*, 2021, pp. 6206–6213.
- [34] 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)*, 2015, pp. 1374–1381.
- [35] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2O-Afford: Annotation-free large-scale object-object affordance learning," in *Conference on Robot Learning (CoRL)*, 2021.
- [36] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," 2021.
- [37] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," 2020.
- [38] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and

- T. Funkhouser, "3dmatch: Learning local geometric descriptors from rgb-d reconstructions," in *CVPR*, 2017.
- [39] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
- [40] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su, "SAPIEN: A simulated part-based interactive environment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [41] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
- [42] M. Laskin, D. Yarats, H. Liu, K. Lee, A. Zhan, K. Lu, C. Cang, L. Pinto, and P. Abbeel, "Urb: Unsupervised reinforcement learning benchmark," 2021.
- [43] M. Laskin, H. Liu, X. B. Peng, D. Yarats, A. Rajeswaran, and P. Abbeel, "Cic: Contrastive intrinsic control for unsupervised skill discovery," 2022.
- [44] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto, "Reinforcement learning with prototypical representations," 2021.
- [45] H. Liu and P. Abbeel, "Behavior from the void: Unsupervised active pre-training," 2021.
- [46] B. Eysenbach, A. Gupta, J. Ibarz, and S. Levine, "Diversity is all you need: Learning skills without a reward function," 2018.
- [47] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," 2017.