Embodiment Matters

Affordance Grounding From Robot and Human Videos

RO57035: RO MSc Thesis

Daniel James Wright

TU Delft

TNO

Embodiment Matters

Affordance Grounding From Robot and Human Videos

by

Daniel James Wright

to obtain the degree of Master of Science in Robotics

at the Delft University of Technology,

to be defended publicly on Thursday September 4, 2025 at 10:30 AM.

Student number:	5932033	
Project duration:	November 11, 2024 – September 4, 2025	
Thesis committee:	Dr. ir. Y. B. Eisma ,	TU Delft, supervisor
	Prof. dr. ir. J. C. F. de Winter,	TU Delft
	Ir. A. C. Kemmeren,	TNO, supervisor
	Dr. D. Dodou	TU Delft

Cover:	Photo by ThisisEngineering on Unsplash under Unsplash License
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

TU Delft

TNO

Embodiment Matters: Affordance Grounding From Robot And Human Videos

Daniel Wright[1,2] Anne Kemmeren[2] Gertjan Burghouts[2] Yke Bauke Eisma[1]

***Abstract*— Affordances, or action possibilities, have been explored to enable robotic manipulation with everyday objects, however the effect of an agent's embodiment has not received much attention. Here we investigate how embodiment changes affordances between a human and robot. We present a method to automatically generate affordance pseudo-labels from a robotic manipulator for the task of grounding (localising) affordances on an object, as there is no such existing dataset. We then propose a general model for embodiment-conditioned affordance grounding, and explore three ways to condition on the embodiment. Our model learns to perform an affine transformation on image embeddings based on the effect of embodiment on the affordance. We evaluate all three variants of our model and compare them to a variant without embodiment conditioning and a state-of-the-art affordance grounding method. The results show that our best performing model decreases affordance prediction error by 25% when compared to the variant without embodiment conditioning and by 68% when compared to the state-of-the-art method. Through our results we demonstrate that embodiment matters when perceiving affordances.**

I. Introduction

For intelligent robots to operate in the real world and perform useful tasks, they need to understand how to interact with everyday objects. This requires detecting the actions that an object *affords* to the robot [1], [2]. However, robots with substantially different embodiments may interact with objects differently. [25], [26]. For example, a human may interact with food differently when using a fork as opposed to tongs. This dependency of affordances on embodiments has had little attention in previous works. This work examines affordance grounding, the task of locating object regions that enable interaction.

Traditional affordance grounding methods sought to learn to locate where affordances occurred on an object from hand-labelled affordance datasets [32], [15], [20], [33], [42]. This research focussed on using the concept of affordances to allow models to generalise in a more human-like manner, learning how to interact with objects, rather than simply where an object is in a scene. These methods also learn many different affordances, which can overlap on a single object. This inherent fuzzy nature of affordances made the manual annotation of these datasets expensive and time consuming. The labels were created from an explicitly human point-of-view, including affordances such as "ride" on a motorcycle.

[1]Intelligent Imaging Group, TNO, The Hague, The Netherlands
[2]Cognitive Robotics Department, Delft University of Technology, Delft, The Netherlands

To avoid the effort of manually labelling datasets, recent studies investigated extracting affordance information from human-object interaction videos [3], [4], [5], [6], [34], [43]. These methods simplify the concept of affordances to interaction locations and so extract hand-object contact point pseudo-labels from these videos using a hand-object detection model [7]. They train affordance grounding models on these pseudo-labels and use them for manipulation tasks. How embodiment can change affordance is not considered.

Thus, there is a gap in the literature on affordance detection models that capture how embodiments change affordances. Affordances can be seen as the complement of agent and object [1], [2], as your embodiment changes how you perceive affordances. A robot with a dexterous hand should perceive different object affordances than a robot with a parallel jaw gripper.

We propose to learn to perceive affordances conditioned on embodiment for robotic manipulation. Following previous works [3], [4], [5] we adopt contact points as our affordance representation. This representation is compact, the same across embodiments, and allows for easy manual annotation. If the predictions are highly precise, they also enable real-world object manipulation with grasping models [4].

To address the gap in embodiment-aware affordance grounding, we propose a model that conditions affordance perception on embodiment. Rather than predicting affordances solely from object appearance, our model also receives an input representing the robot's embodiment, allowing it to adapt predictions to the capabilities of the agent. Conditioning in this way enables the model to capture how different embodiments interact with the same object in distinct ways. We explore three forms of conditioning: a categorical embodiment variable, a learned embodiment variable, and a natural language text prompt. By integrating embodiment information into the learning process, we aim to improve affordance grounding performance and generalisation across embodiments.

Learning embodiment-conditioned affordances for robotic manipulation requires extracting affordance information from different embodiments. As there are no existing affordance grounding datasets for a non-human embodiment, we utilize recently published robotic datasets [8], [30] and obtain contact point pseudo-labels using foundation models for free supervision.

We evaluate our method and the three conditioning inputs on in-domain and out-of-domain embodiments. We perform several experiments to answer the following research questions:

1) How much does conditioning on embodiment improve affordance grounding?
2) What is the most effective way to condition on embodiment?
3) Does conditioning on embodiment improve generalisation to unseen embodiments?
4) Why does embodiment change affordance?
5) How does our model condition on embodiment?

II. Related Work

A. Visual Affordance Learning

Initial research on visual affordance learning relied on manually annotated datasets [32], [15], [20], [49], [50]. Early methods used models based on convolutional neural networks [53], [52], [51], which then progressed to methods using transformer-based models [42], [14], [36]. The manual annotation required to create these datasets made them expensive, even for weakly supervised methods. This lead to research into automatically extracting affordance information from human-object interaction videos [3], [4], [5], [6], [34], [43], [45]. In these methods affordances are represented as heatmaps or contact points. Recent methods, also seeking to avoid extensive manual labelling, have extracted affordance knowledge from foundation models [16], [14], [35], [44], [46], [47], [48]. Different from these methods, we propose to automatically extract affordance information from robot-object interaction videos, providing a new source of data for visual affordance learning. This then enables us to learn how affordances differ due to embodiment, which previous methods are unable to do as they rely solely on human-object interaction data.

B. Conditioning For Affordance Learning

Affordances are inherently dependent on a number of factors, such as object size, shape, task semantics and agent's embodiment [25], [54]. To enable deep learning methods to predict affordances based on these factors requires conditioning on them. Previous methods have conditioned affordance predictions on the geometry of objects by using 3D information [32], [33], [15], [14], [34], [37], [38], [39]. Other methods explored the effect of physical properties of the objects on their affordances [40], [35], [41]. Task semantics have been included by conditioning predictions on natural language prompts [5], [55], [56], [57]. One previous method has shown that embodiment-aware affordances can improve manipulation [61], but only explores one embodiment. These methods all make significant contributions to understanding affordances, but do not explore how different embodiments can alter affordance. To explore this factor of affordance, we condition our model on the embodiment of the agent.

C. Learning From Demonstrations

Learning from demonstrations involves using human examples of tasks, where the human controls the robot via teleoperation, to teach robots. The state-of-the-art uses vision-language-actions (VLA) models to learn to directly control robots [27], [28], [58], [60]. VLAs are comprised of a vision encoder and a large language model backbone, and have achieved remarkable success when trained on large datasets of demonstrations [28], [8], [59]. These methods achieve cross-embodiment generalisation, but only for similar embodiments, such as robots with different sizes of parallel jaw grippers. They learn to output control commands specific to an embodiment and so a model trained for a parallel jaw gripper cannot control a dexterous hand. In contrast to these methods, we extract the relevant affordance information from the demonstrations. Furthermore, our approach does not learn a one-size-fits-all model, but is able to switch between modes for different embodiments. We hypothesize that this enables the model to leverage similarities between embodiments and generalisation capabilities, but it is also able to learn where the embodiments are fundamentally different.

III. Method

Our goal is to learn embodiment-conditioned affordances which can be used for robotic manipulation. The structure of this section is as follows: in Section III-A we describe how we use an existing affordance grounding dataset and extract contact point pseudo-labels from a robotic dataset, in Section III-B we detail our general model architecture, in Section III-C we hypothesize how our model conditions on the embodiment, and in Section III-D we describe the three variations of how we condition on embodiment.

A. Affordance Pseudo-label Generation

To condition affordance grounding on different embodiments, we need a dataset which contains images with contact point labels from two embodiments. We use existing affordance labels from a human-object interaction dataset and generate new affordance labels from a robot-object interaction dataset.

1) Human Affordance Data Generation: We follow existing affordance learning methods which extract affordances from human egocentric videos and use the publicly available data from [6]. Their dataset consists of tuples of a pre-interaction RGB image, x_h and contact points c_h^i. We take the average of the contact points as the affordance label, giving one contact point, $\bar{c}_h$ and the final tuple $(x_h, \bar{c}_h)$. We denote this dataset as the Hand dataset.

2) Robot Affordance Data Generation: We propose a novel method to automatically extract affordance information from the robot-object interactions in the DROID dataset [8], shown in Figure 1. The DROID dataset consists of videos and trajectory data of a Franka Panda robotic manipulator with a parallel jaw gripper completing tasks teleoperated by a human. The dataset contains videos of each task from two exocentric cameras and one wrist camera, as well as the end-effector pose and gripper state from the trajectories of the robot arm. The DROID dataset was chosen due to its variety of objects, tasks and scenes, as well as its large size and inclusion of gripper state. Additionally, the embodiment represented in the DROID dataset is both significantly different from a human embodiment, and common in the literature. To automatically extract affordance pseudo-labels from the

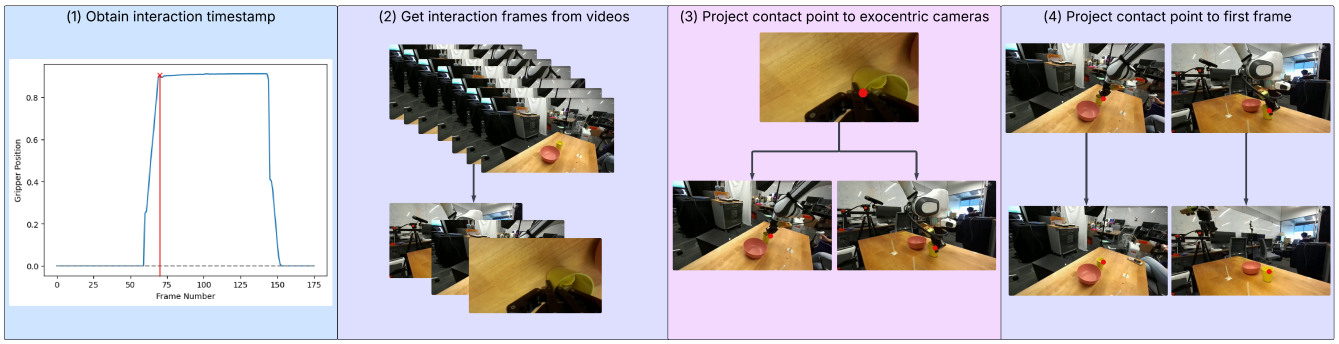

Fig. 1. Dataset creation method: (1) shows the chosen interaction timestamp from the gripper position data, (2) shows the interaction frames being chosen from the videos, (3) shows the contact point being projected from the wrist camera to the two exocentric cameras, and (4) shows the contact point being projected from the interaction frame to the first frame for both exocentric videos. These final images in (4) are the data used to train the proposed models.

DROID dataset we need to determine *when* and *where* the interaction takes place.

When the interaction takes places was determined automatically using the gripper state data from the robot. The interaction between the gripper and object begins when the gripper is fully closed around the object. Each trajectory in the DROID dataset contains the gripper state, which describes the position of the gripper. The value of the gripper state ranges from 0 to 1, where 0 is fully open and 1 is fully closed. For a given interaction, the behaviour of the gripper state is that of a rising edge, increasing as the gripper closes, reaching a peak when fully closed around the object, and then forming a plateau as the interaction unfolds. To automatically obtain the timestamp of the interaction, the gripper state is filtered to find peaks. Any trajectories where the gripper state does not change is discarded. The first peak is taken as the interaction timestamp and used to obtain the frames from each of the three cameras. We denote these frames where the interaction occurs as an *interaction frame*.

To determine *where* the interaction occurs in the video, we exploit the fixed nature of the wrist camera. The wrist camera is mounted directly above the robot's gripper. As the gripper closes, the interaction then occurs in the centre point of the gripper's two fingers. This contact point is then projected from the wrist camera's frame to the two exocentric cameras' frames. We obtain free supervision from the VGGT [9] foundation model for this task. VGGT is a transformer-based model that infers all 3D attributes of a scene, including camera parameters. For our purposes, it also has a point tracking head, which we use to predict where the contact point in the wrist camera image is visible in the two exocentric camera images.

Then for each exocentric camera, the contact point is projected from the interaction frame to the first frame of the video, again using VGGT. This reduces the domain gap between the human and robot datasets, by ensuring the robot gripper is not in the final image. The extracted pseudo-labels were visually inspected to exclude outliers. This then gave contact point pseudo-labels for each video in the DROID dataset in the same tuple format of (x_r, c_r), which we denote as the Gripper dataset.

3) Combining Datasets: We now have two datasets: the Hand dataset containing tuples $(x_h, \bar{c}_h)$ and the Gripper dataset containing tuples (x_r, c_r). To condition our affordance prediction on the embodiment, we propose an embodiment variable, e. We define e as a categorical variable, with $e = 0$ for the Hand dataset and $e = 1$ on the Gripper dataset. We combine these two datasets into a single dataset which we will denote as the Combined dataset, which contains tuples (x, c, e), where x is an RGB image of the pre-interaction scene, c is the pixel location of the contact point and e is the embodiment variable.

To reduce the domain gap between the Hand and Gripper datasets we crop the image to the relevant object. This crop is the input image, x_{crop}, to the model.

4) Object Selection: Each dataset contains a wide range of objects. We aimed to choose objects that fulfilled a number of criteria. These criteria are:

1) The objects are common, everyday objects.
2) The objects have a range of sizes and geometries.
3) The objects may have different affordances for different embodiments.

The set of objects present in the HOI4D and DROID dataset fulfil the first criteria. From this set we selected five specific objects: mugs, bottles, bowls, knives and scissors. These objects span a range of geometries. The mug is asymmetrical, the bowl is concave, the bottle is large and symmetrical, while the knife and scissors introduce thin, elongated geometries with specific interaction areas. This range of geometries enables our method to control for the effect of geometry and better determine the effect of embodiment on affordance.

5) Affordance Task Relationship: Many previous studies have shown the dependence of affordance localisation on the task semantics [25]. We control for the effect of the task on the affordance location by only choosing pick-and-place tasks from HOI4D and DROID. For example, from HOI4D a task is to "Pick and place the mug", whilst from DROID a task is to "Pick the mug and move it to the bottom left of the table".

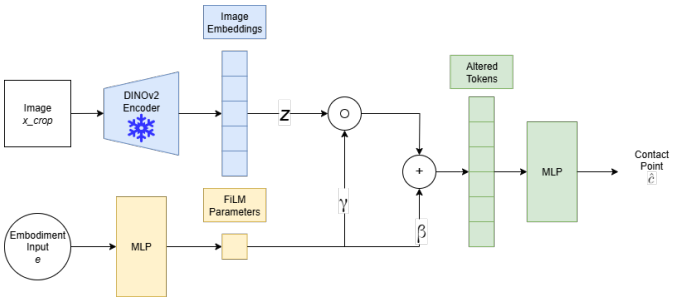

Fig. 2. Overview of the proposed models' architecture. It is comprised of a DINOv2 image encoder which outputs image embeddings, an MLP projector which projects the embodiment input to the image embedding space and an MLP decoder which outputs the predicted contact point.

B. Model Architecture

The embodiment-conditioned affordance model takes an image and an embodiment variable and predicts a contact point. The general model architecture is shown in Figure 2. We propose three variations with different embodiment variables and describe them in Section III-D. For ease of notation, we denote the whole model as f_θ and the modules of the model as $(\cdot)_\theta$, where θ denotes a neural network. The input image x_{crop} is encoded using a frozen DINOv2 visual encoder [10], g_θ, to give image embeddings z. The image embeddings from DINOv2 encode depth information [10], which is why we do not include a depth map as additional input to our model.

$$z = g_\theta(x_{crop}) \tag{1}$$

With $x_{crop} \in \Re^{h \times w}$, where h is the height of the crop in pixels, w is the width of the crop in pixels, $z \in \Re^{p \times l}$, where p denotes the number of tokens output by the visual encoder, and l is the token length.

The embodiment variable, e is projected by a modified FiLM [11] layer, h_θ, which we parametrise as a two-layer MLP, to a size twice that of the token length. This is reshaped into the scaling and bias terms, γ and β respectively:

$$\gamma, \beta = h_\theta(e) \tag{2}$$

With $\gamma, \beta \in \Re^{1 \times l}$. The image embeddings are then altered by the scaling and bias terms, before being decoded by a two-layer MLP, f_θ:

$$\hat{c} = f_\theta(\gamma \cdot z + \beta) \tag{3}$$

Where $\hat{c} \in R^2$ is the predicted contact point. The loss for the contact point is the mean square error between the prediction, $\hat{c}$ and the ground truth, c, formally:

$$L = \frac{1}{n} \Sigma_{i=1}^{n} (c_i - \hat{c}_i)^2 \tag{4}$$

With $c \in R^2$, and n is the number of samples in the mini-batch.

C. Conditioning on Embodiment

To condition the model's predictions on the embodiment, we seek to alter the image embeddings. We use a modified version of the general-purpose conditioning layer from FiLM. FiLM layers learn an affine transformation which is applied to the image embeddings, scaling or shifting them. Intuitively this can be thought of as moving the conditioned embeddings in the high-dimensional embedding space, enabling a model to draw a decision boundary between them.

As our method alters image embeddings [11], it can be used with future affordance grounding methods which follow the classic encoder-decoder model design.

D. Embodiment Input Variations

We propose three variations of model architecture based on different embodiment inputs, which we denote as Categorical, Learned and Text.

1) Categorical: The embodiment variable is 0 for Hand data and 1 for Gripper data, with $e \in R^1$.

2) Learned: The embodiment variable is a learnable embedding for each embodiment with dimension 1. The embeddings are initialised as 0 for Hand data and 1 for Gripper data, with $e \in R^1$.

3) Text: The embodiment variable is a text description of the gripper tokenised and encoded using the DINOv2 text encoder [29]. For Hand data the text prompt is "Five fingered dexterous hand" and for Gripper data the text prompt is "Two fingered parallel jaw gripper", with $e \in R^l$.

IV. Experiments and Results

In this section we present a thorough evaluation of our approach and answer the proposed research questions:

1) How much does conditioning on embodiment improve affordance grounding?
2) What is the most effective way to condition on embodiment?
3) Does conditioning on embodiment improve generalisation to unseen embodiments?
4) Why does embodiment change affordance?
5) How does our model condition on embodiment?

A. Experimental Setup

1) Train and Validation Splits: We split the Combined dataset into a train set and a validation set. The train set consists of 1,237 images and the validation set consists of 303 images. Each image in the train set has one annotation, whilst the images in the validation set have two annotations: their original annotation and a manually annotated contact point for the other embodiment.

2) Model Settings: We train the baseline, Categorical, Learned, Text and ablation models on an Nvidia RTX-3090 for 250 epochs with a learning rate of 5×10^-4, a batch size of 8 and an AdamW optimiser [31]. Loss curves are shown in Appendix C.

3) Early Stopping: After training each model for 250 epochs, we report the metrics using the model checkpoint from the epoch with the lowest validation loss.

4) Baseline Models: We compare our proposed model to a baseline model, as well as a state-of-the-art affordance grounding method, VRB [3]. The baseline consists of a frozen DINOv2 encoder and 2 layer MLP decoder. This architecture is the same as the proposed model in Figure 2 without the embodiment conditioning. It is trained with the same MSE loss as the proposed model. As we want to compare the accuracy of contact point predictions, to have a fair comparison we constrain the output of VRB. VRB outputs a heatmap, or probability distribution over the affordance area. We take the point in the heatmap predicted by VRB with the highest probability as the contact point.

5) Evaluation Metrics: For the purpose of measuring the performance of our method, we create ground truth heatmaps for the validation set. This allows us to compute more detailed metrics which enable deeper analysis of our proposed method. To create the heatmaps we apply a Gaussian blur to the contact points, following [20], [21].

Root Mean Square Error (RMSE): The RMSE is calculated as the average of the L2 distances between the predicted contact points and the ground truths, normalised for the size of the image. For predicted points, $\hat{c}$, and ground truths c, the formula of RMSE is:

$$Err = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}(\hat{c}_i - c_i)^2} \tag{5}$$

Normalised Scanpath Saliency (NSS): Measures the correlation between saliency maps and fixed points, considering their accuracy and saliency. A higher value corresponds to a contact point closer the center of ground truth heatmap. We use the metric adapted to calculate the average normalised value of the ground truth map at the predicted contact points [4]. For ground truth heatmap M, and predicted contact points $\hat{c}$, the formula of NSS is:

$$NSS = \frac{1}{n}\Sigma_{i=1}^{n}(\frac{M(\hat{c})}{\max_{c \in M} M(c)}) \tag{6}$$

Success Rate (SR): Success rate is calculated as the fraction of predicted contact points which fall within the ground truth heatmap. The values of the ground truth heatmap range from 0 to 255, so a threshold of 122 is chosen to determine if the output is feasible [4]. The formula of SR is:

$$SR = \frac{1}{n}\Sigma(s_i) \tag{7}$$

Where the success value s_i for a predicted contact point $\hat{c}$ is given by:

$$s_i = \begin{cases} 1, & \text{if } M(\hat{c}) > 122 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

Distance to Mask (DTM): This metric calculates the shortest distance between the predicted contact point and the ground truth heatmap. We threshold the mask using the same value as in success rate and then calculate the distance. If the predicted contact point is inside the thresholded mask, the DTM value is 0. We report the average normalised value. For a ground truth heatmap M, predicted contact point $\hat{c}$ and point p in M closest to c, The formula for DTM is:

$$DTM = \begin{cases} 0, & \text{if } \hat{c} \in M \\ \min\limits_{p \in M} \|\hat{c} - p\|, & \text{if } \hat{c} \notin M \end{cases} \tag{9}$$

B. Affordance Grounding Results

1) Discussion of proposed methods: Table I presents the experimental results for the entire validation set, for our proposed methods and the comparison methods. We report the mean and standard deviation for each metric over five training runs for the baseline, Categorical, Learned and Text models. Each individual run is reported in Appendix D. The proposed Learned method achieves the lowest RMSE, 0.217, outperforming the baseline by 25% and VRB by 68%. It also achieves the highest NSS. This shows that the Learned method's predictions are more consistent and closer to the correct affordance region, as they are closer to the center of the ground truth heatmap. The proposed Categorical method has the highest success rate and DTM. The Text method is the worst performing of the proposed methods, which shows that the text prompts do not allow the model to distinguish embodiments as clearly as the other proposed methods.

TABLE I
CONDITIONING ON EMBODIMENT IMPROVES AFFORDANCE GROUNDING PERFORMANCE.

Model	RMSE ↓ Mean (SD)	NSS ↑ Mean (SD)	SR ↑ Mean (SD)	DTM ↓ Mean (SD)
VRB [3]	0.365 (-)	0.579 (-)	0.644 (-)	0.023 (-)
Baseline	0.272 (0.001)	0.697 (0.003)	0.770 (0.010)	0.019 (0.001)
Categorical	0.218 (0.001)	0.771 (0.001)	**0.872** (0.010)	**0.016** (0.001)
Learned	**0.217** (0.003)	**0.774** (0.004)	0.871 (0.008)	0.017 (0.003)
Text	0.255 (0.006)	0.718 (0.010)	0.821 (0.014)	0.019 (0.002)

2) Comparison to baselines: All of our proposed methods outperform the baseline and VRB. The proposed methods' improved performance over the baseline show the benefit of conditioning on embodiment. The improvement over VRB is more multifaceted. VRB is trained only on data from human egocentric videos and so has not seen data from robot-object interactions. It does not condition on embodiment, and uses an image encoder trained from random initialisation on a dataset curated specifically for affordance grounding. Our simple approach, using a strong backbone, outperforms this train-from-scratch approach.

C. Generalisation to Novel Object-Embodiment Combinations

At test time we evaluate on novel combinations of object and embodiment. While the models have prior exposure to the object categories and embodiments separately from the Hand and Gripper datasets, they have not seen object instances from the Hand dataset with Gripper embodiment labels and vice-versa. For each image in the validation set we manually annotate an additional contact point corresponding

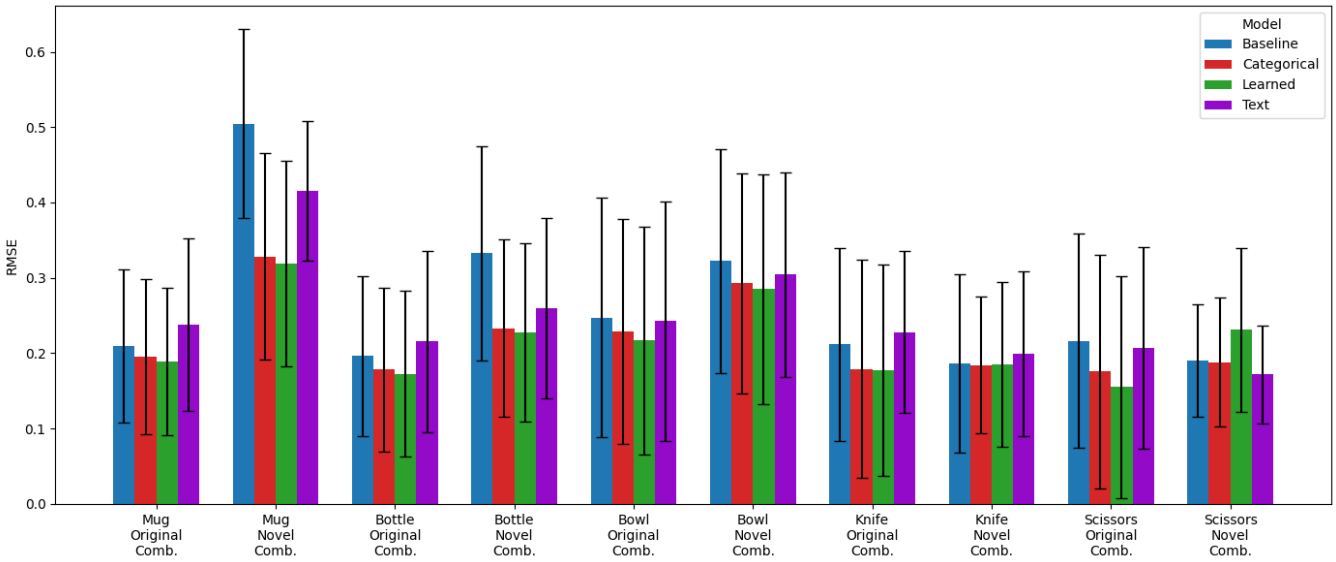

Fig. 3. RMSE by object and original or novel annotation for the baseline and proposed methods. Conditioning on the embodiment enables generalisation to novel combinations of object instance and embodiment.

to the other embodiment. We denote these additional validation samples as "Novel Combination", and the originally generated contact point labels as "Original Combination".

Figure 3 shows the validation RMSE for each object in the dataset, split into "Original Combination" and "Novel Combination" sets, for the baseline and the proposed methods, from one training run. For all sets the Categorical method outperforms the baseline method. In only the Scissors Novel Combination split does the Learned method perform worse than the baseline. This shows that these methods are able to generalise to novel combinations of object instance and embodiment. Whilst the baseline model learns the affordance location solely based on the image features, thereby giving the same prediction for each Original and Novel set, the proposed methods are able to learn how the embodiment changes the affordance location. The Text method performs worse than the baseline for most of the Original sets, but better for most of the Novel sets. This shows that the information from the text encoder is not useful for affordance grounding, but does enable the method to condition on embodiment. This is unsurprising as the text encoder is trained using a contrastive loss, aligning text captions to images [29].

We posit that this difference is most significant for the Mug and Bottle "Novel" sets as the difference in interaction is also the most different for these two embodiments. Due to the geometry, size and weight of these objects, the two-fingered gripper and human hand interact with them differently. Via inspection of the training data, see Appendix A, we argue that for the mug the human hand interaction largely occurs at the handle, whereas for the gripper the interaction occurs at the rim of the mug. This may be due to the two-fingered gripper being unable to pick up a mug at the handle due to the small surface area and large, unbalanced moment. For the bottle, the human hand interaction largely occurs towards the bottom of the bottle, whilst the gripper interaction occurs towards the top. This may be due to the two-fingered grasp pose not balancing moments if it picked up a bottle from the bottom, causing the bottle to rotate and fall out of the gripper.

The difference in performance between the baseline and proposed models is much less pronounced for knives, scissors and bowls. This is due to the affordance location being similarly placed for these objects across the two embodiments. For the knives and scissors, their smaller size and lower weight is likely why the affordance location is not significantly different between embodiments. Inspection, see Appendix A, shows that for the bowls, both embodiments interact at the rim. The geometry of a bowl restricts the locations where interactions are possible, explaining the similarity in affordance location between the embodiments.

D. Data Efficiency Experiment

In this experiment, we aim to show how conditioning on embodiment in our method is a powerful inductive bias. We train from random initialisation a baseline model and a Categorical model on increasing percentages of the Mug training dataset, which consists of 356 images of mugs from both embodiments. We evaluate these models on the Mug validation split of the validation dataset. Figure 4 shows how conditioning on embodiment enables the Categorical model to more efficiently learn from data than the baseline model. The Categorical model has a lower RMSE than the baseline model with only 10% or 20% of the training data. While the baseline model's RMSE improves greatly from 10% to 40% of the training data, it then plateaus, while the Categorical model's RMSE continues to decrease.

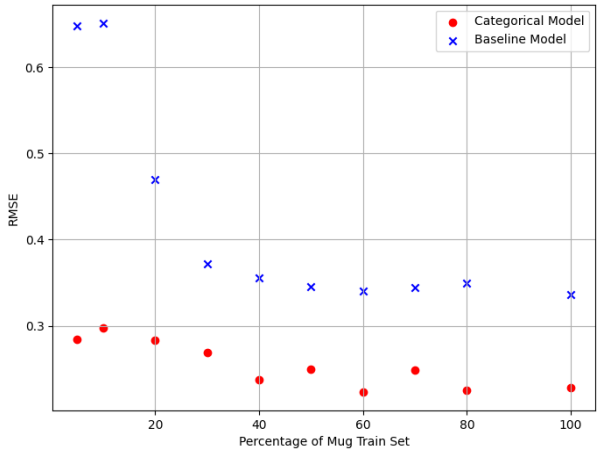

Fig. 4. RMSE for a Categorical and a baseline model trained on increasing percentages of the Mugs training dataset. The Categorical model requires less training samples for a lower RMSE, showing that conditioning on embodiment is an inductive bias that improves learning efficiency.

E. Few-Shot Generalisation Experiment

The purpose of this experiment is to determine the ability of the proposed methods to generalise to a new embodiment given a few examples to learn from. We manually annotate a small amount of data from the Google robot [30] interacting with mugs. This robot has a two-fingered claw gripper, and so interacts with mugs in a different manner than the Franka Panda robot from DROID. We denote these data as the Claw set, shown in Appendix B. Rather than picking up the mug at the rim or handle, it grabs the body of the mug.

We train the baseline and proposed methods on the Combined training dataset, with 0, 1, 2, 4 or 8 Claw samples added. We assign the Google robot a categorical embodiment variable of 2, initialise the Learned embodiment variable to 2 and set the text prompt to "Claw gripper". In Figure 6 we report the RMSE of each method on a test set of data annotated with Claw contact points from the Hand, Gripper and Claw datasets.

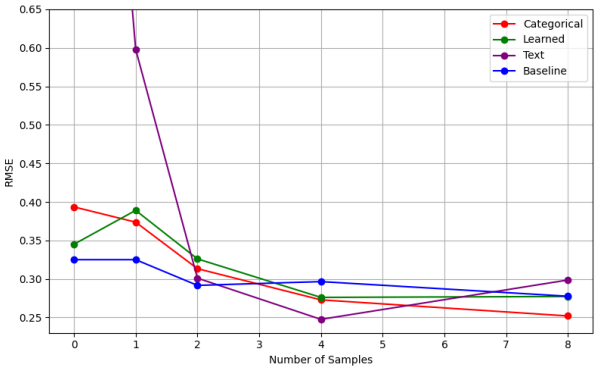

Fig. 6. Embodiment-conditioned affordance grounding enables few-shot generalisation. The Categorical model approaches its overall validation RMSE score for a new embodiment with only 8 training samples.

None of the methods show a strong ability to perform zero-shot generalisation to a new embodiment. However, the Categorical method consistently improves, achieving an RMSE of 0.252 with only 8 samples from the Claw embodiment. The Learned method and baseline perform similarly, indicating that learning the difference between the embodiments requires more data than is available in the few-shot setting. The Text method achieves the lowest RMSE of 0.248 with 4 samples but then performs worse, indicating possibly unstable training.

F. Qualitative Results

We also show that the mechanism of the Categorical model's embodiment conditioning is as hypothesized. We expected that the FiLM parameters γ and β were performing an affine transformation on the image embeddings, shifting the embeddings to create a decision boundary [24]. This would then allow the model to in some sense switch its predictions between the embodiments. We validate this hypothesis by obtaining the Categorical model's embeddings of the Mug validation data. We then use t-SNE [12] to reduce the high-dimensional image embeddings to two dimensions for visualisation. Figure 7 shows how the embodiment variable shifts the image embeddings. The Hand images show a small but significant shift from the categorical variable. Interestingly, the Gripper images are entirely linearly separated by the categorical embodiment variable. The model moves their embeddings towards the Hand image embeddings.

We further validate this hypothesis with qualitative results from our method. Figure 5 shows one example of each object in the dataset, with affordance predictions form the baseline and proposed methods. The top row shows Hand annotations, whilst the bottom row shows Gripper annotations. The baseline method makes the same prediction for both images, whilst the proposed methods make different predictions.

G. Ablations

To validate the design choices in our method, we perform several experiments using the Categorical model to determine the influence of parts of our method.

1) Data Transformations: Our problem setting involves images from two different data distributions. To reduce the domain gap between the Hand and Gripper images we investigate the effects of data transformations. Additionally, applying transformations to images during training has been shown to improve model generalisation [18], [19]. Transformations prevent the model from overfitting spurious correlations present in the training dataset [22], and they inflate the amount of training samples that are seen by the model [23]. We apply transformations of scale, colour, rotation and perspective to determine if they improve model generalisation. We always apply horizontal flip to our methods to ensure our models learn vertical symmetry. We report the mean and standard deviation over five training runs for each transformation combination. Each individual run is reported in Appendix D. Table II shows that large scale jitter [17], rotation and perspective transformations improve performance slightly, whilst colour jitter decreases performance. As the validation set is taken from the same

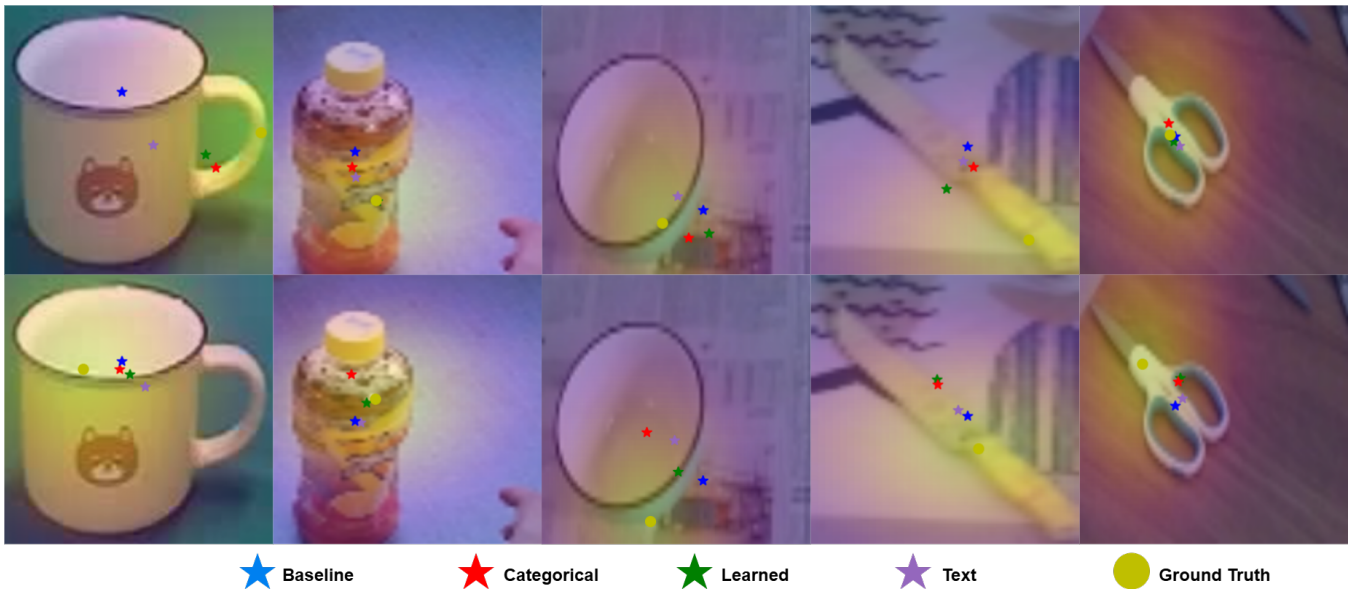

Fig. 5. Visualisation of predicted contact points from the baseline and proposed methods. The top row shows a Hand embodiment label and the bottom row shows a Gripper embodiment label.

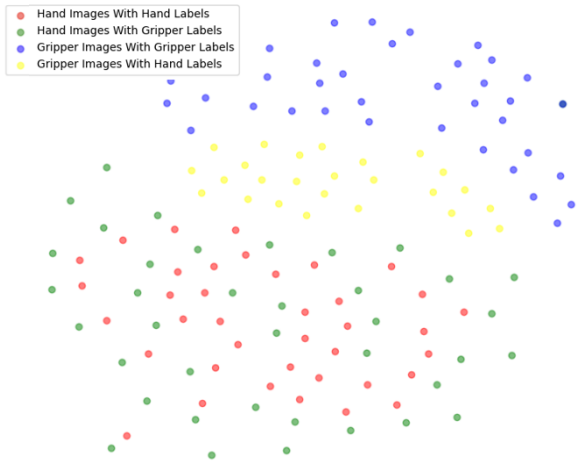

Fig. 7. t-SNE plot of the Categorical model's embeddings of validation mug data with different embodiment variables. The model learns to separate the same image's embeddings based on the embodiment variable.

dataset as the training set, large changes in colour are unlikely, whereas changes in scale, orientation and perspective are more likely. This is why colour jitter decreases generalisation performance, whilst large scale jitter, rotation and perspective transformations improve it. In our reported results in Table I we use horizontal flip, large scale jitter, rotation and perspective transformations when training. Note that our implementation of large scale jitter places the object randomly in the scene, not in the top left.

2) Encoder: To validate our choice of encoder, we tested the method with several state-of-the-art image encoders. A version of the Categorical model was trained using each encoder. We chose four encoders to compare against DINOv2: VGGT, and the three Perception Encoders: Core, Language and Spatial [13]. Previous research has shown that DINOv2 outperformed other image encoders for pixel level tasks [16], but these four encoders were released after that research was performed. The Perception Encoders were chosen as they improved performance in a number of areas and in particular PE Spatial outperformed DINOv2 in several pixel-level tasks such as semantic segmentation [13]. We chose VGGT as literature has shown that the 3D structure of objects is important for affordance grounding [14], [15] and VGGT excels at reconstructing 3D scenes. Table III reports the results over the entire validation dataset, for one training run. DINOv2 shows the best results, with an RMSE of 0.218. Whilst it is unexpected that DINOv2 outperforms PE Spatial, we speculate this is due to the used hyperparameters. It is out of scope of this research to perform a thorough hyperparameter search to determine the cause.

TABLE II

ABLATION OF DATA TRANSFORMATIONS.

Transformation	RMSE ↓ Mean (SD)	NSS ↑ Mean (SD)	SR ↑ Mean (SD)	DTM ↓ Mean (SD)
Crop & Horizontal Flip	0.227 (0.003)	0.757 (0.005)	0.852 (0.007)	0.018 (0.003)
& w/ Large Scale Jitter [17]	0.222 (0.002)	0.770 (0.004)	**0.874** (0.006)	**0.016** (0.002)
& w/ Rotate	**0.219** (0.001)	**0.771** (0.003)	0.869 (0.007)	**0.016** (0.002)
& w/ Colour Jitter	0.222 (0.003)	0.768 (0.005)	0.868 (0.006)	0.017 (0.001)
& w/ Perspective	**0.219** (0.003)	**0.771** (0.004)	0.870 (0.006)	**0.016** (0.002)

TABLE III

ABLATION OF THE IMAGE ENCODER

Model	RMSE ↓ Mean (SD)	NSS ↑ Mean (SD)	SR ↑ Mean (SD)	DTM ↓ Mean (SD)
DINOv2 [10]	**0.218** (0.137)	**0.769** (0.248)	**0.858** (0.349)	**0.016** (0.075)
PE Core [13]	0.378 (0.181)	0.490 (0.333)	0.512 (0.500)	0.026 (0.089)
PE Language [13]	0.277 (0.137)	0.542 (0.321)	0.569 (0.496)	0.026 (0.089)
PE Spatial [13]	0.579 (0.257)	0.658 (0.279)	0.764 (0.425)	0.026 (0.088)
VGGT [9]	0.306 (0.138)	0.647 (0.278)	0.752 (0.432)	0.025 (0.088)

3) Projector: To validate the performance of the projector, we increase its size. In our proposed method, γ and β both have size $(1, 768)$ and are then broadcast across the image embeddings, which have size (257, 768). We test a version where γ and β both have size (257, 768) to see if this enables more fine-grained manipulation of the image embedding space. Table IV shows that the smaller size outperforms the larger size for RMSE and NSS, whilst the larger size performs slightly better on SR and DTM. As RMSE is our primary metric, we choose the size (1,768) for our methods. These results show that the smaller size projector is able to learn all the necessary information, and we speculate that this means that the larger projector is overfitting the data.

TABLE IV
ABLATION OF THE PROJECTOR SIZE

Size	RMSE ↓ Mean (SD)	NSS ↑ Mean (SD)	SR ↑ Mean (SD)	DTM ↓ Mean (SD)
(1,768)	**0.218** (0.137)	**0.769** (0.248)	0.858 (0.349)	0.016 (0.075)
(257, 768)	0.228 (0.137)	0.762 (0.249)	**0.859** (0.249)	**0.015** (0.070)

H. Limitations

While our method shows that there is a noticeable difference in affordance grounding between the two embodiments, further research is needed to confirm that our method improves real-world grasping success rates. We believe that our method may show larger gains for settings where the embodiments show a higher degree of dissimilarity, for example a dexterous hand compared to a pushing rod. However, this research requires generation of new high-quality affordance datasets for these embodiments. As our method is data-driven, it is heavily reliant on the quality of the data. The pseudo-labels from the datasets limit the quality of the predictions possible from the proposed models. Moreover, limitations common to vision models such as perspective apply to our method, see Appendix E.

V. CONCLUSION

In this paper we have presented embodiment-conditioned affordance grounding, by learning affordances from robot and human videos. We proposed a novel method to automatically extract affordances from a large robot dataset. We proposed three methods for conditioning on the embodiment, and evaluated them against a baseline. Our method improves the prediction accuracy compared to the baseline by 25%, and by 68% when compared to the state-of-the-art method. We have thus shown empirically that embodiment does change how affordances are perceived. We also show how our method achieves this performance, by altering the high-dimensional image embedding space. This provides key insights that our method can leverage similarities between embodiments, but also learn when embodiment fundamentally changes affordance. In the future we hope to explore other embodiments, incorporate these ideas into more complex architectures such as VLAs, and perform experiments with real world robots.

ACKNOWLEDGMENT

I would like to thank my supervisors, Ir. Anne Kemmeren, Dr. ir. Gertjan Burghouts, and Dr. ir. Yke Eisma for their time, support and insights during my thesis. In particular I would like to thank Anne for her constant encouragement and help during difficult patches of my thesis. Mum, Dad, Hannah, Joe, and Eve, thank you for your unconditional love and support during the thesis and the entire degree. I couldn't have done it without you. Last, but not least, thank you to all my friends, new and old. Thank you for putting up with my ceaseless complaining and thank you for being there when I needed you.

REFERENCES

[1] J. J. Gibson, *The ecological approach to visual perception: classic edition*. Psychology press, 2014.

[2] J. J. Gibson, 'The senses considered as perceptual systems', 1966.

[3] 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. 13778–13790.

[4] Y. Ju, K. Hu, G. Zhang, G. Zhang, M. Jiang, and H. Xu, 'Robo-abc: Affordance generalization beyond categories via semantic correspondence for robot manipulation', in *European Conference on Computer Vision*, 2024, pp. 222–239.

[5] X. Gao et al., 'Learning 2d invariant affordance knowledge for 3d affordance grounding', in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025, vol. 39, pp. 3095–3103.

[6] A. Kannan, K. Shaw, S. Bahl, P. Mannam, and D. Pathak, 'Deft: Dexterous fine-tuning for real-world hand policies', arXiv preprint arXiv:2310. 19797, 2023.

[7] D. Shan, J. Geng, M. Shu, and D. F. Fouhey, 'Understanding human hands in contact at internet scale', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9869–9878.

[8] A. Khazatsky et al., 'Droid: A large-scale in-the-wild robot manipulation dataset', arXiv preprint arXiv:2403. 12945, 2024.

[9] J. Wang, M. Chen, N. Karaev, A. Vedaldi, C. Rupprecht, and D. Novotny, 'Vggt: Visual geometry grounded transformer', in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 5294–5306.

[10] M. Oquab et al., 'Dinov2: Learning robust visual features without supervision', arXiv preprint arXiv:2304. 07193, 2023.

[11] 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, vol. 32.

[12] L. van der Maaten and G. Hinton, 'Visualizing data using t-SNE', Journal of machine learning research, vol. 9, no. Nov, pp. 2579–2605, 2008.

[13] D. Bolya et al., 'Perception encoder: The best visual embeddings are not at the output of the network', arXiv preprint arXiv:2504. 13181, 2025.

[14] 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.

[15] W. Zhai, H. Luo, J. Zhang, Y. Cao, and D. Tao, 'One-shot object affordance detection in the wild', International Journal of Computer Vision, vol. 130, no. 10, pp. 2472–2500, 2022.

[16] G. Li, D. Sun, L. Sevilla-Lara, and V. Jampani, 'One-Shot Open Affordance Learning with Foundation Models', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 3086–3096.

[17] G. Ghiasi et al., 'Simple copy-paste is a strong data augmentation method for instance segmentation', in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 2918–2928.

[18] S. Wu, H. R. Zhang, G. Valiant, and C. Ré, 'On the Generalization Effects of Linear Transformations in Data Augmentation', arXiv [cs.LG]. 2023.

[19] L. Taylor and G. Nitschke, 'Improving Deep Learning using Generic Data Augmentation', arXiv [cs.LG]. 2017.
[20] 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, 2022, pp. 2252–2261.
[21] K. Fang, T.-L. Wu, D. Yang, S. Savarese, and J. J. Lim, 'Demo2vec: Reasoning object affordances from online videos', in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2139–2147.
[22] M. Ilse, J. M. Tomczak, and P. Forré, 'Selecting Data Augmentation for Simulating Interventions', arXiv [stat.ML]. 2020.
[23] R. Balestriero, I. Misra, and Y. LeCun, 'A Data-Augmentation Is Worth A Thousand Samples: Analytical Moments And Sampling-Free Training', in Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 19631–19644.
[24] C. Keup and M. Helias, 'Origami in N dimensions: How feed-forward networks manufacture linear separability', arXiv [cs.LG]. 2022.
[25] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, 'Computational models of affordance in robotics: a taxonomy and systematic classification', Adaptive Behavior, vol. 25, no. 5, pp. 235–271, 2017.
[26] 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.
[27] M. J. Kim et al., 'OpenVLA: An Open-Source Vision-Language-Action Model', in Conference on Robot Learning, 2025, pp. 2679–2713.
[28] A. O'Neill et al., 'Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0', in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 6892–6903.
[29] C. Jose et al., 'Dinov2 meets text: A unified framework for image- and pixel-level vision-language alignment', in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 24905–24916.
[30] P. Sermanet et al., 'RoboVQA: Multimodal Long-Horizon Reasoning for Robotics', arXiv [cs.RO]. 2023.
[31] I. Loshchilov and F. Hutter, 'Decoupled Weight Decay Regularization', in International Conference on Learning Representations, 2019.
[32] 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.
[33] A. Guo et al., 'Handal: A dataset of real-world manipulable object categories with pose annotations, affordances, and reconstructions', in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 11428–11435.
[34] G. Li et al., 'Learning precise affordances from egocentric videos for robotic manipulation', arXiv preprint arXiv:2408. 10123, 2024.
[35] S. Huang et al., 'A3VLM: Actionable Articulation-Aware Vision Language Model', in Conference on Robot Learning, 2025, pp. 1675–1690.
[36] Y. Li, N. Zhao, J. Xiao, C. Feng, X. Wang, and T.-S. Chua, 'Laso: Language-guided affordance segmentation on 3d object', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 14251–14260.
[37] T. Nguyen et al., 'Language-conditioned affordance-pose detection in 3d point clouds', in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 3071–3078.
[38] T. Nguyen et al., 'Open-vocabulary affordance detection in 3d point clouds', in 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 5692–5698.
[39] S. Ling et al., 'Articulated object manipulation with coarse-to-fine affordance for mitigating the effect of point cloud noise', in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 10895–10901.
[40] A. Kemmeren, G. Burghouts, M. van Bekkum, W. Meijer, and J. van Mil, 'Which objects help me to act effectively? Reasoning about physically-grounded affordances', arXiv preprint arXiv:2407. 13811, 2024.
[41] J. Gao et al., 'Physically grounded vision-language models for robotic manipulation', in 2024 IEEE International Conference on Robotics and Automation (ICRA), 2024, pp. 12462–12469.
[42] 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. 10922–10931.
[43] J. Chen, D. Gao, K. Q. Lin, and M. Z. Shou, 'Affordance grounding from demonstration video to target image', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 6799–6808.
[44] J. Tang, G. Zheng, J. Yu, and S. Yang, 'Cotdet: Affordance knowledge prompting for task driven object detection', in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3068–3078.
[45] H. Chen, B. Sun, A. Zhang, M. Pollefeys, and S. Leutenegger, 'VidBot: Learning Generalizable 3D Actions from In-the-Wild 2D Human Videos for Zero-Shot Robotic Manipulation', in Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 27661–27672.
[46] A. Rai, K. Buettner, and A. Kovashka, 'Strategies to leverage foundational model knowledge in object affordance grounding', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 1714–1723.
[47] C. Cuttano, G. Rosi, G. Trivigno, and G. Averta, 'What does CLIP know about peeling a banana?', in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 2238–2247.
[48] E. Tong, A. Opipari, S. R. Lewis, Z. Zeng, and O. C. Jenkins, 'OVAL-Prompt: Open-Vocabulary Affordance Localization for Robot Manipulation through LLM Affordance-Grounding', in First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024.
[49] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, '3D AffordanceNet: A Benchmark for Visual Object Affordance Understanding', arXiv [cs.CV]. 2021.
[50] 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), 2017, pp. 5908–5915.
[51] L. Mur-Labadia, R. Martinez-Cantin, and J. J. Guerrero, 'Bayesian Deep Learning for Affordance Segmentation in images', arXiv [cs.CV]. 2023.
[52] T.-T. Do, A. Nguyen, and I. Reid, 'AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection', arXiv [cs.CV]. 2018.
[53] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, 'Detecting object affordances with Convolutional Neural Networks', in 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016, pp. 2765–2770.
[54] P. Ardón, È. Pairet, K. S. Lohan, S. Ramamoorthy, and R. Petrick, 'Building affordance relations for robotic agents-a review', arXiv preprint arXiv:2105. 06706, 2021.
[55] C. Chen, Y. Cong, and Z. Kan, 'WorldAfford: Affordance Grounding based on Natural Language Instructions', arXiv [cs.CV]. 2024
[56] W. Qu, L. Guo, J. Cui, and X. Jin, 'Multimodal attention-based instruction-following part-level affordance grounding', Applied Sciences, vol. 14, no. 11, p. 4696, 2024.
[57] M. Ahn et al., 'Do As I Can, Not As I Say: Grounding Language in Robotic Affordances', arXiv [cs.RO]. 2022.
[58] K. Black et al., 'π_0: A Vision-Language-Action Flow Model for General Robot Control', arXiv [cs.LG]. 2024.
[59] M. U. Din, W. Akram, L. S. Saoud, J. Rosell, and I. Hussain, 'Vision Language Action Models in Robotic Manipulation: A Systematic Review', arXiv [cs.RO]. 2025.
[60] S. Nasiriany et al., 'RT-Affordance: Affordances are Versatile Intermediate Representations for Robot Manipulation', arXiv [cs.RO]. 2024.
[61] G. Schiavi, P. Wulkop, G. Rizzi, L. Ott, R. Siegwart, and J. J. Chung, 'Learning Agent-Aware Affordances for Closed-Loop Interaction with Articulated Objects', arXiv [cs.RO]. 2023.

APPENDIX

A. Training Data From The Hand and Gripper Dataset

Figure 8 shows training data from each object category. The top row corresponds to objects from the Hand dataset,

whilst the bottom row corresponds to objects from the Gripper dataset.

Fig. 8. Example training data from the Hand and Gripper datasets.

B. Training Data From the Claw Dataset

Figure 9 shows the training data from the Claw dataset.

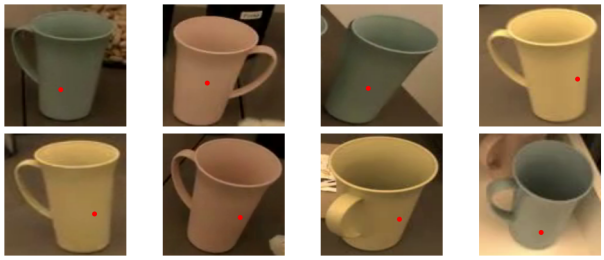

Fig. 9. Training data from the Claw datasets.

C. Proposed and Baseline Loss Curves

Figures 10-17 show the train and validation loss curves for the three proposed models and the baseline model.

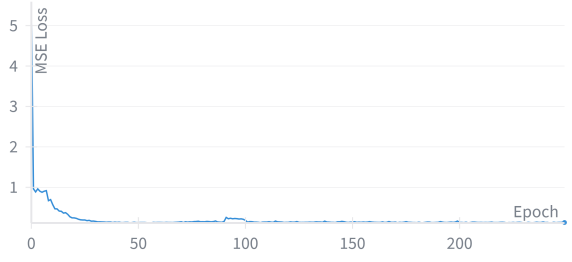

Fig. 10. Baseline model's train loss curve.

D. Individual Run Results

The results of each individual run are reported in Tables V-XIII. The mean and standard deviation for each metric is reported over the validation dataset.

E. Failure Mode

If the region of the object which offers the affordance is not visible, our method fails. This failure mode is shown in Figure 18 for a mug where the handle is not visible. The Categorical model cannot make a good prediction for the Hand embodiment.

Fig. 11. Baseline model's validation loss curve.

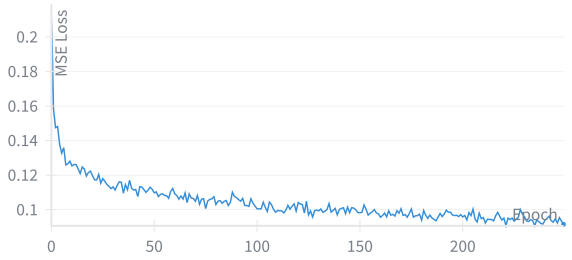

Fig. 12. Categorical model's train loss curve.

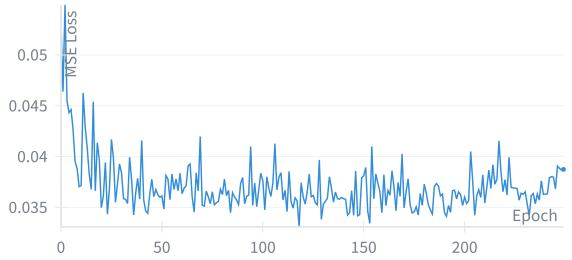

Fig. 13. Categorical model's validation loss curve.

Fig. 14. Learned model's train loss curve.

TABLE V
BASELINE MODEL INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.27076 (0.1601)	0.69827 (0.2816)	0.77703 (0.4166)	0.01906 (0.0772)
2	0.26947 (0.1653)	0.69792 (0.2849)	0.76520 (0.4242)	0.01935 (0.0786)
3	0.26978 (0.1649)	0.69847 (0.2878)	0.76014 (0.4274)	0.01767 (0.0752)
4	0.27268 (0.1595)	0.69868 (0.2726)	0.78378 (0.4120)	0.01661 (0.0689)
5	0.27338 (0.1607)	0.69154 (0.2850)	0.76182 (0.4263)	0.01957 (0.0768)

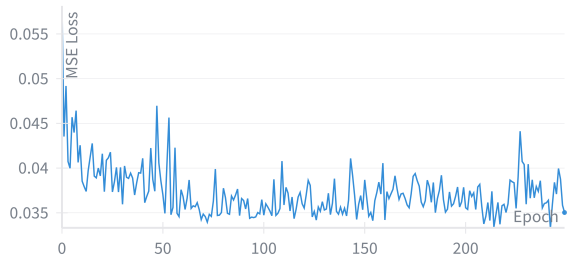

Fig. 15. Learned model's validation loss curve.

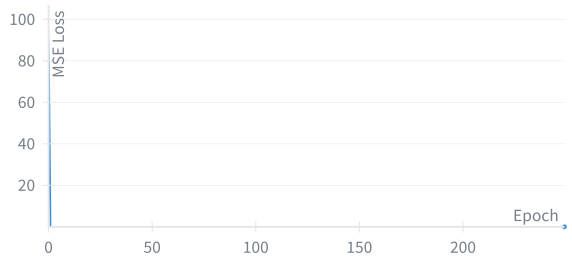

Fig. 16. Text model's train loss curve.

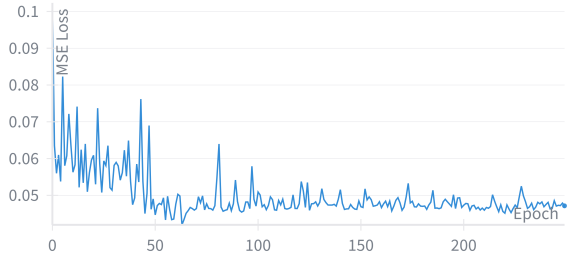

Fig. 17. Text model's validation loss curve.

TABLE VI

CATEGORICAL MODEL INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21812 (0.1365)	0.76923 (0.2475)	0.85811 (0.3492)	0.01681 (0.0754)
2	0.21725 (0.1320)	0.77198 (0.2421)	0.86655 (0.3403)	0.01639 (0.0728)
3	0.21819 (0.1379)	0.77295 (0.2470)	0.87500 (0.3310)	0.01579 (0.0715)
4	0.21783 (0.1295)	0.77231 (0.2425)	0.87838 (0.3271)	0.01620 (0.0723)
5	0.21961 (0.1289)	0.77067 (0.2374)	0.88176 (0.3232)	0.01723 (0.0757)

TABLE VII

LEARNED MODEL INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21325 (0.1302)	0.77947 (0.2389)	0.87669 (0.3291)	0.01586 (0.0679)
2	0.22088 (0.1364)	0.76690 (0.2491)	0.85980 (0.3475)	0.02037 (0.0830)
3	0.21532 (0.1342)	0.77347 (0.2481)	0.86993 (0.3367)	0.01902 (0.0802)
4	0.21533 (0.1366)	0.77730 (0.2428)	0.88176 (0.3232)	0.01509 (0.0697)
5	0.21930 (0.1363)	0.77350 (0.2427)	0.86824 (0.3385)	0.01388 (0.0673)

TABLE VIII

TEXT MODEL INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.25972 (0.1387)	0.70960 (0.2695)	0.81926 (0.3851)	0.01879 (0.0804)
2	0.24511 (0.1375)	0.73565 (0.2523)	0.84459 (0.3626)	0.01640 (0.0734)
3	0.25721 (0.1397)	0.71231 (0.2715)	0.80574 (0.3960)	0.02071 (0.0836)
4	0.25596 (0.1351)	0.71287 (0.2634)	0.81926 (0.3851)	0.02069 (0.0837)
5	0.25585 (0.1362)	0.71789 (0.2639)	0.81757 (0.3865)	0.01907 (0.0805)

TABLE IX

NO JITTER TRANSFORMATION INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.22421 (0.1405)	0.76257 (0.2547)	0.85642 (0.3510)	0.01477 (0.0708)
2	0.22281 (0.1442)	0.76252 (0.2624)	0.84797 (0.3594)	0.01542 (0.0700)
3	0.22993 (0.1349)	0.75498 (0.2540)	0.85811 (0.3492)	0.02049 (0.0834)
4	0.22617 (0.1408)	0.75240 (0.2630)	0.85642 (0.3510)	0.02066 (0.0847)
5	0.23050 (0.1451)	0.75118 (0.2688)	0.84122 (0.3658)	0.01927 (0.0809)

TABLE X

JITTER TRANSFORMATION INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21980 (0.1369)	0.77279 (0.2498)	0.87838 (0.3271)	0.01703 (0.0746)
2	0.21974 (0.1332)	0.77215 (0.2395)	0.87500 (0.3310)	0.01461 (0.0670)
3	0.22205 (0.1385)	0.77237 (0.2422)	0.87838 (0.3271)	0.01378 (0.0651)
4	0.22313 (0.1349)	0.76550 (0.2479)	0.87162 (0.3348)	0.01970 (0.0795)
5	0.22370 (0.1344)	0.76626 (0.2458)	0.86486 (0.3422)	0.01677 (0.0745)

TABLE XI

ROTATION TRANSFORMATION INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21846 (0.1319)	0.77238 (0.2394)	0.87669 (0.3291)	0.01606 (0.0698)
2	0.21781 (0.1351)	0.77382 (0.2455)	0.86993 (0.3367)	0.01701 (0.0745)
3	0.22100 (0.1310)	0.76536 (0.2462)	0.85811 (0.3492)	0.01945 (0.0791)
4	0.22080 (0.1392)	0.77059 (0.2473)	0.87162 (0.3348)	0.01458 (0.0675)
5	0.21854 (0.1354)	0.77353 (0.2437)	0.86655 (0.3403)	0.01492 (0.0698)

TABLE XII

COLOUR TRANSFORMATION INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21755 (0.1322)	0.77433 (0.2428)	0.87669 (0.3291)	0.01552 (0.0708)
2	0.21979 (0.1331)	0.77299 (0.2449)	0.87162 (0.3348)	0.01784 (0.0763)
3	0.22317 (0.1404)	0.76573 (0.2505)	0.86486 (0.3422)	0.01777 (0.0776)
4	0.22612 (0.1371)	0.76304 (0.2442)	0.86149 (0.3457)	0.01672 (0.0737)
5	0.22293 (0.1324)	0.76574 (0.2442)	0.86655 (0.3403)	0.01801 (0.0763)

TABLE XIII

PERSPECTIVE TRANSFORMATION INDIVIDUAL RUN RESULTS

Run	RMSE ↓	NSS ↑	SR ↑	DTM ↓
1	0.21742 (0.1330)	0.77387 (0.2437)	0.87331 (0.3329)	0.01379 (0.0647)
2	0.22288 (0.1349)	0.76713 (0.2451)	0.86149 (0.3457)	0.01596 (0.0704)
3	0.22191 (0.1340)	0.76869 (0.2458)	0.86655 (0.3403)	0.01748 (0.0750)
4	0.21925 (0.1307)	0.77065 (0.2458)	0.87162 (0.3348)	0.01687 (0.0720)
5	0.21491 (0.1338)	0.77714 (0.2461)	0.87838 (0.3271)	0.01475 (0.0699)

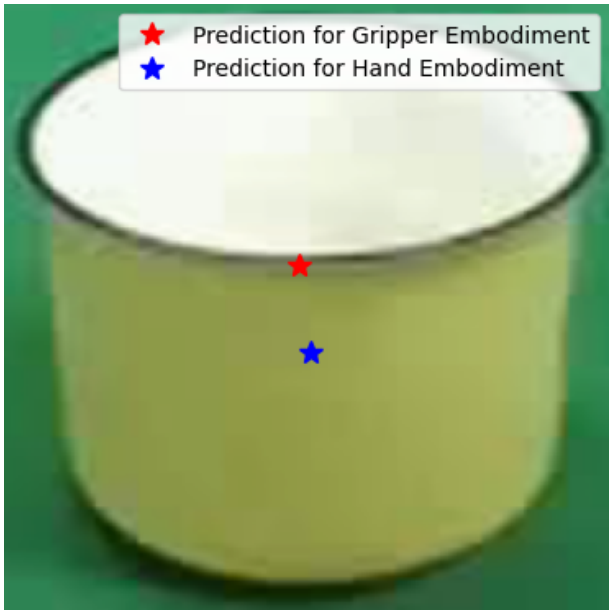

Fig. 18. Failure mode where the mug's handle is not visible.