Semantic Cross-Pose Correspondence from a Single Example

Denis Hadjivelichkov, Sicelukwanda Zwane, Marc Peter Deisenroth, Lourdes Agapito, and Dimitrios Kanoulas

***Abstract*— This article focuses on predicting how an object can be transformed to a semantically meaningful pose relative to another object, given only one or few examples. Current pose correspondence methods rely on vast 3D object datasets and do not actively consider semantic information, which limits the objects to which they can be applied. We present a novel method for learning cross-object pose correspondence. The proposed method detects interacting object parts, performs one-shot part correspondence, and uses geometric and visual-semantic features. Given one example of two objects posed relative to each other, the model can learn how to transfer the demonstrated relations to unseen object instances. Supplementary details can be found at https://sites.google.com/view/semantic-pose-correspondence**

I. Introduction

Many skills humans use to interact in the real world involve object manipulation, for instance, cutting a loaf of bread or hanging a painting. People naturally acquire and transfer this knowledge to novel objects, enabling them to perform a wide range of tasks with ease. Replicating this ability in robots remains challenging. One key challenge is that object manipulation often involves cross-object interactions. Predicting the correct object pose for such interactions requires understanding the relative transformations between the objects, which can vary depending on the specific objects.

Recent approaches to learning object manipulation from demonstrations have focused on identifying "bottleneck" poses, or intermediate configurations relevant to the task [1], [2]. By limiting the search space, robots are able to more easily learn new interactions. Inspired by these works, we focus on identifying and transferring bottleneck poses across different object pairs (i.e. *finding the semantically corresponding cross-poses*). However, this is not trivial – while object geometry can provide some clues, it is insufficient due to variations in appearance and texture. Moreover, obtaining multi-view object data, such as complete point clouds, is particularly difficult in live settings where a robot must operate autonomously with limited sensory information.

Relevant approaches from skill imitation learning do not transfer well to novel out-of-distribution environments [3], [4] or objects [5], or rely on large category-level datasets [6]. Meanwhile, some geometric approaches have shown successful cross-pose correspondences [7], however, they still struggle with occluded single-view observations, which are often the only available data in practical scenarios. Additionally, they also fail to make use of the rich semantic information from the visual observations.

The authors are with University College London, 66 Gower Street, WC1E 6BT, London, UK. Dimitrios Kanoulas is also with Archimedes/Athena RC, Greece. Corresponding author is Denis Hadjivelichkov.

This work was supported by UKRI FLF [MR/V025333/1] and CDT of FAI [EP/S021566/1]. For Open Access, the author has applied a CC BY copyright license to any manuscript version arising from this submission. We thank L. Beddow and M. Stamatopoulou for the useful discussions.

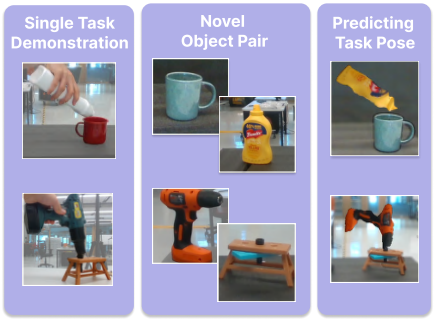

Fig. 1: Given one or few example poses of an object-object interaction, we predict a semantically corresponding pose of a pair of unseen objects with the same relation.

To address these challenges, we propose a novel approach for one-shot semantic correspondence of object poses from single-view examples. Our method leverages semantic information from visual inputs and unsupervised detection of interaction regions to learn correspondences between objects involved in cross-object interactions. This enables us to generalize to novel objects using just one demonstrated example pose (see Figure 1). We demonstrate the ability of the model to transfer object cross-pose relations to novel object pairs, across a plethora of simulated and real example scenes. In this work, we use example poses that can be extracted automatically from an RGB-D video demonstration

We leave the problem of executing high-precision tasks that require learning the interaction dynamics, as future work; for instance by using the cross-pose correspondences to bootstrap the imitation learning of skills with new tools.

Contribution: This work contributes the following:

- A novel method for predicting semantically accurate cross-pose correspondences using one or few demonstrated interactions, without requiring data-intensive pre-training.
- Integration of semantic visual features with unsupervised interaction region detection for robust generalization across novel object pairs.
- Validation through extensive experiments, showing competitive performance and successful deployment.

II. Literature Review

Predicting how novel objects can be used is a challenging problem. Imitation learning works often assume test-time object distribution and struggle with visual variability [1], [3]. Other methods integrate language models for abstract

reasoning and skill transfer across different objects [8], [9], enhancing adaptability to new tasks. These methods highlight the complexity of skill transfer to novel objects and inference from unknown demonstrations. Despite recent advancements in semantic correspondence [10], [11], [12], it is underutilised in learning how to use similar objects.

In computer vision, semantic correspondence is often achieved via proxy representations [13] or cyclic loss [14]. Reliance on pre-trained backbones is common [15], [16], [17], [18], [19]. Most notably, features from pre-trained DINO-ViT [20] are effective in tasks benefiting from semantic reasoning such as zero-shot co-segmentation [21], pose estimation [22] and skill learning [23]. Some works deal with the problem of finding semantically corresponding manipulator poses with respect to novel objects that share similar parts [24], [25], [26]. Bahl et al. [4] detail how human hand-actions are split into pre-, during- and post-interaction primitives. However, direct one-shot transfer to a robot is not possible — the robot needs to fine-tune its policy from safe interaction near the perceived goal. [25] also consider correspondence based on "action" parts. However, only geometric features are considered and part matching is done by simply re-scaling, which limits the method's usability. Inspired by [27], leveraging neural fields has become prevalent due to their continuous spatiotemporal nature beneficial for optimizers, notably in grasping [6], [28], [29], [26]. In recent works, shape models [30], [24], [13] facilitate skill transfers across objects with shared geometries, albeit within constraints of similar poses and geometrically akin parts. Being data-driven, these methods are not directly comparable to our work, as they rely on training sets of objects belonging to some particular category, while several of these works also assume that any test object also belongs to the same distribution as the training set.

Recent approaches propose the deformation of the demonstration objects followed by alignment to novel unseen objects in order to geometrically find corresponding cross-poses [31], [32]. Close to our work, DINObot [33] makes use of DINO-ViT features for manipulation tasks, albeit the method requires robot hand camera demonstrations and treats the problem as camera alignment rather than tool alignment. The closest to our method is TAX-Pose [7]. It takes in a few demonstrations of a desired pose and transfers it successfully to novel object instances. Unlike works that rely on full-object matching for correspondence [13], [7], we propose to limit the correspondence to a small region of the example object where the interaction occurs. As shown in previous works [30], [34], the point distribution in this small region may be reliable and sufficient for cross-pose estimation.

III. Problem Statement

To distinguish between a tool and the object it manipulates, we follow the terminology used in related literature and refer to the instrument the agent manipulates with as the **manipulator**, and to the object that the instrument is applied on as the **manipulandum**. In other works, these may be referred to as the Movable Object/tool and Receiver Object/anchor, respectively. We also define a cross-object **interaction** as the period during which the two objects are affecting each other's states.

Given a set of task demonstrations that depict interactions between pairs of objects, and one novel scene containing objects that could have a similar interaction, our goal is to find a pose for the manipulator in the novel scene corresponding to the demonstrated ones. As in related works [6], [7], we assume the manipulandum is fixed, and only predict the *interaction pose*, not a full trajectory.

Throughout this work, the superscript A is used for the manipulator and B for the manipulandum. Subscript 0 denotes the initial frame in a demonstration. Each demo includes aligned color and depth images (i.e., RGB-D) showing the execution of a single task, and masks of the objects in the first frame, M_0^A and M_0^B. Note that the masks may be provided by annotation or generated without user supervision. In this work, we use language-guided object segmentation for demonstration masks and unsupervised object detection, such as the one used in [19], for the novel scenes. Projecting these masks onto 3D points, the respective point clouds of the object P_0^A and P_0^B are computed, having N_A and N_B points respectively. Novel scenes, similarly, include a single RGB-D frame and masks of the objects. Except for determining the manipulator pose and interaction part at the start of the interaction, we do not use the full demonstration.

IV. Method

We build upon TAX-Pose [7]. Given some demonstrations, we first estimate the 2D object regions which are important for the interaction (Section IV-A). A model is then trained to predict how these regions interact in 3D (Section IV-B). Given a novel pair of objects, we can find the corresponding interaction regions using one-shot part correspondence [19]. Finally, we infer what the semantically corresponding interaction pose is. Refer to the system overview in Figure 2.

A. Estimating the Region of Interaction

We estimate the segmentation mask of each object, which is important for the interaction, denoted as the region of interaction (RoI). During cross-object interaction, the objects often occlude each other, while before the interaction, they are typically less occluded. We take the point clouds P_0^A, P_0^B from the initial frame, and transform them to their interaction poses using transforms obtained by an off-the-shelf pose-tracker [35]. Thus, we can estimate their point clouds despite occlusion, visual artifacts, and motion blur. Transforming the initial manipulator points P_0^A along the trajectory using a transform T_t from the pose tracker, we compute the Euclidean distance between the objects as:

$$d(T_t P_0^A, P_0^B) = \min_{p_t^A \in T_t P_0^A, p^B \in P_0^B} ||p_t^A - p^B||. \quad (1)$$

We define the *interaction frame*, denoted with subscript int, as the frame at which the distance between the two objects' point clouds is the smallest

$$int = \arg\min_t d(T_t P_0^A, P_0^B). \quad (2)$$

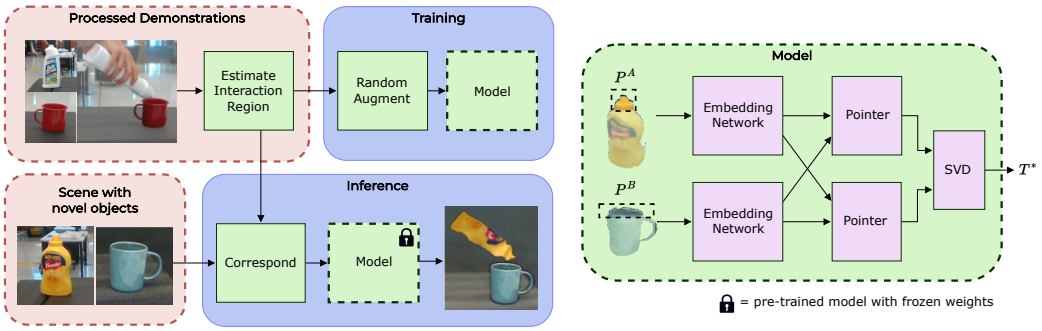

Fig. 2: **Left:** The demonstrations are pre-processed to determine the regions of interaction. The points belonging to those areas are used as training data. Random transforms are applied to the object points at each training step. The model is trained to predict the manipulator transformation that would bring it to the ground truth pose. During inference, the corresponding interaction area is found and fed into the trained model to predict a semantically corresponding pose; **Right:** The model architecture comprises point-wise embedder networks, encoder-decoder transformers (pointers) and differentiable SVD.

The closest pair of points is then noted as p^A_{int} and p^B_{int}. This is the frame in which contact would occur if the manipulation involves physical contact. A subset of points

$$P^A_{int} = \{p^A_t \in T_t P^A_0, \text{ s.t. } ||p^A_{int} - p^A_t|| \leq d_{thres}\}, \quad (3)$$

within some Euclidean threshold d_{thres}, are selected for the region of interaction (see Figure 3). Projecting the 3D points to back 2D, an interaction region mask M^A_{int} can be generated. The RoI M^B_{int} is also approximated similarly. Since the manipulandum B is assumed static, the tracking transform T_t is the identity matrix throughout the demonstration.

B. Goal Pose Learning

Our cross-pose learning model extends TAX-Pose [7], by introducing visual-semantic features and working with object parts instead of full objects. The model used to learn the desired pose of the manipulator is split into two symmetrical streams, one predicting the transformation from A to B; and another the reverse from B to A. For brevity, the method outlined below in (1) and (2) describes one of the streams (see Figure 2-right), while the other is symmetrically identical (i.e. with A and B swapped).

*1) **Embedding**:* To embed semantic information, we use the feature descriptors produced by the DINO-ViT backbone model $\phi(.)$ at the initial RGB frame I_0. Each feature descriptor belonging to the RoI masks M_{int} is projected from its 2D location within the image to the existing point cloud using 3D ray-casting. Any points that are considered invalid or abnormal, due to typical sensor issues with depth cameras, are filtered out via statistical outlier removal. Thus, we obtain features $F^A = \phi(I_0) \odot M_{int}$ associated with 3D points, where $\odot$ is the application of the mask. We opted for using projected 2D features from a pretrained model as that allows us to make use of the vast image datasets without retraining for a specific object. We considered embedding 3D semantic features directly as in [36], [37]. However, such networks cannot easily generalize to unseen objects due to lack of 3D training data. Recently, projected 2D features have proven more useful and scalable across several computer vision tasks [23], [38], [39]. For 2D features, we considered using a supervised network trained on semantic correspondences, or a self-supervised network with emergent semantic capabilities. We chose DINO-ViT features as they performed best as local dense semantic descriptors at the time of developing this method [21].

The points are re-centered via the point cloud mean $\bar{P^A}$ and concatenated with their respective visual features F^A to form point-feature vectors $X^A = [F^A, P^A - \bar{P^A}]$. A point-wise embedding network $g^A_\epsilon(X^A)$ then outputs an embedding for each vector. Note that, unlike DCP [40] and TAX-Pose [7], the embedding network proposed here does not encode anything about the point cloud's local or global geometry. Since these features contain point-wise information, we cannot use the DGCNN point-cloud encoder that TAX-Pose uses. Instead, we opt to off-load point-cloud processing to the following cross-pose correspondence transformer, while point-wise encoding simply downsizes the input features to more manageable dimensions.

*2) **Correspondence**:* The correspondence process follows [40], [7] for a fully differentiable model w.r.t. input points P^A. This is done by first determining a set of virtual points corresponding to where the input points should be, and then finding the optimal transform to bring the point cloud as close as possible to its virtual correspondences. Firstly, for each point $p^A \in P^A$ we assign some point $v^A \in V^A$ which is within the convex hull of P^B via a weighted mean

$$V^A = P^B W^{A \to B}, \quad (4)$$

where $W^{A \to B} \in \mathbb{R}^{N_B \times N_A}$ is a normalized weight matrix such that $\sum_{N_A} W^{A \to B} = 1$. We refer to V^A as "soft" correspondences. Intuitively, this weight should determine the importance of each pair of points from the two point-clouds. To compute weight matrices that incorporate information about both point-clouds, we apply a transformer network $g^{A \to B}_\tau$ (noted as "pointer") which produces cross-object point embeddings

$$\psi^A = g^{A \to B}_\tau(g^A_\epsilon(X^A), g^B_\epsilon(X^B))). \quad (5)$$

The softmax of the last attention layer of the transformer serves as the weights $W^{B \to A}$ used above. Since these soft

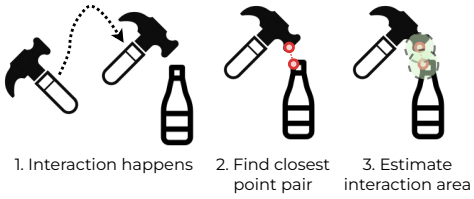

Fig. 3: To find the part of the interacting object which are involved in the interaction (1), the closest 3D point pair is found (2), and all points within some threshold distance from those points are selected (3).

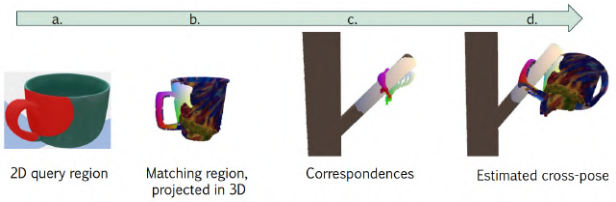

Fig. 4: Correspondence stages: (a) A query region is matched to a target object. (b) Target region is projected to 3D (shown colored) and used as input to our model. (c) For each query point, a virtual correspondence is computed. (d) The optimal transform is determined.

correspondences are constrained within the convex hull of the object, they do not retain the same shape as the initial point cloud — they need to be adjusted to get correspondences that retain a similar shape. For this purpose, a displacement residual vector δ is used. The residual $\delta^A = g_R(\psi^A)$ is learned with a network g_R that uses the point embeddings as input. Finally, the correspondences are defined as

$$\tilde{V}^A = P^B W^{A \to B} + \delta^A. \tag{6}$$

*3) **Optimal Transform**:* The optimal transform between two sets of points is computed using Singular Value Decomposition (SVD). Specifically, we search for one homogeneous transformation matrix that transforms the manipulator points P^A into their virtual correspondences $\tilde{V}^A$, while also transforming the virtual manipulandum points $\tilde{V}^B$ into their correspondences P^B. Using differentiable SVD, we decompose the cross-covariance matrix

$$H = \begin{bmatrix} P^A - \bar{P}^A \\ \tilde{V}^B - \bar{V}^B \end{bmatrix} \begin{bmatrix} \tilde{V}^A - \bar{V}^A \\ P^B - \bar{P}^B \end{bmatrix}^T \tag{7}$$

into $H = USD^T$, where $\bar{P}$ and $\bar{V}$ are point cloud centroids.

The optimal transform T^* (see example in Figure 4) is then comprised of a rotation $R^* = DU^T$ and a translation

$$t^* = \frac{1}{N} \sum_{i=0}^{N} \begin{bmatrix} \tilde{V}^A - \bar{V}^A \\ P^B - \bar{P}^B \end{bmatrix}_i - R^* \begin{bmatrix} P^A - \bar{P}^A \\ \tilde{V}^B - \bar{V}^B \end{bmatrix}_i, \tag{8}$$

where N is the number of points in the concatenated matrices. Note that the model is translation-equivariant.

*4) **Training**:* Since the model's architecture is not rotation-equivariant, we apply random transformations to the input points and re-compute the corresponding ground truth T^{gt}. At each training step, random samples of the input points are used to make the model more robust to different point clouds and reduce the model's reliance on the point order. As [7] we use losses minimizing the distances between virtual points, transformed input points, and ground truth points. Specifically, we use point displacement loss

$$\begin{aligned} \mathbb{L}_{disp.} \quad = \quad & \frac{1}{N_A} \textstyle\sum_A |T^* P^A - T^{gt} P^A| \\ & + \frac{1}{N_B} \textstyle\sum_B |T^{*-1} P^B - T^{gt-1} P^B| \end{aligned}, \tag{9}$$

virtual correspondence loss

$$\begin{aligned} \mathbb{L}_{corr.} \quad = \quad & \frac{1}{N_A} \textstyle\sum_A |V_A - T^{gt} P^A| \\ & + \frac{1}{N_B} \textstyle\sum_B |V_B - T^{gt-1} P^B| \end{aligned}, \tag{10}$$

and a consistency loss

$$\begin{aligned} \mathbb{L}_{cons.} \quad = \quad & \frac{1}{N_A} \textstyle\sum_A |T^* P^A - V^A| \\ & + \frac{1}{N_B} \textstyle\sum_A |T^{*-1} P^B - V^B| \end{aligned}. \tag{11}$$

The final objective with λ mixing coefficients is

$$\mathbb{L} = \lambda_{disp.} \mathbb{L}_{disp.} + \lambda_{corr.} \mathbb{L}_{corr.} + \lambda_{cons.} \mathbb{L}_{cons.}. \tag{12}$$

C. Cross-Poses of Novel Object Pairs

Given some set of demonstrated interactions, each RoI is first estimated. The points within those RoI and their associated DINO-ViT features are used to train the Goal Pose learning model. For inference in novel scenes, our model takes in one of the demonstrated object pairs and finds the 2D semantic correspondences of the interaction regions in the novel object pair using our model AffCorrs [19] (see Figure 4-b). AffCorrs takes as inputs the initial RoI mask $M_{int,demo}$ of either demonstration object, the corresponding RGB image, and the RGB image containing the novel object. The model outputs the corresponding 2D RoI mask $M_{int,novel}$ for either object. These masks are then used as outlined in the training procedure to infer an optimal cross-pose between the two objects. Since the semantic DINO-ViT features belong to the same latent space and represent only the RoI, we can use them directly as inputs for the goal pose learning model to predict cross-poses.

D. Implementation

We use the DINO-ViT-S model with patch size of 8. While these features are derived from a complex model pre-trained on ImageNet, their availability as pre-trained tools allows for their use without requiring additional training on datasets specific to each task. The point embedding networks have four MLP layers with ReLU activations and 512 output dimensions. Each pointer transformer has one encoding and decoding layer, with standard attention. The residual models have three MLP layers with ReLU activations and one final MLP layer downscaling to three dimensions. The coefficients

are $\lambda_{cons.} = \lambda_{corr.} = \lambda_{dist.} = 1$. We used a learning rate of 10^{-4} for 1000 training steps, which takes approximately 10 minutes to train on one NVidia RTX3080 GPU.

V. Evaluation

A. Experimenal Setup

Baseline: We test our model against the TAX-Pose baseline which it extends. For TAX-Pose, we use the official implementation released by the authors. The baseline is initialized with their provided checkpoint weights, which are available for mug-related tasks. For each task outlined, all models are trained for 1000 steps on the same data, with the same training procedure and training augmentations.

Quantitative Tasks: 1) Mug-Pour. We simulate a pouring task in PyBullet. The models are trained and tested on random mug pairs – we use five mugs from GSOD [41] with random image textures for training, and five mugs generated with Shap-E [42] with random textures for testing. This equates to 25 possible training pairs, and 25 testing pairs. The models are trained on a random selection of a few (out of 25 possible) training examples and tested on the 25 unique pairs unseen by the models. The simulated manipulator mug is transformed into the model-predicted pose and 100 balls are simulated inside it. The fraction of balls successfully poured into the manipulandum mug is the task success rate. A mug collision is considered a failure with a 0% success.

2) Mug-Rack. We also simulate mug-rack task, similar to [7], but with textured mugs from the set used for Mug-Pour and only a single-view demonstration instead of full textureless models. We focused on two settings: upright and arbitrarily rotated mugs, defining success as the mug being stable on the rack without falling or collision.

3) Survey. We use single viewpoint examples of three real tasks (drill on bolt, hammer on cup and bottle pour in mug) to train a model on each method. We conduct a survey of 24 people asked to evaluate the predictions on novel objects on a scale from 1 (*"does not resemble the interaction at all"*) to 5 (*"resembles the interaction as closely as possible"*). Three unique object combinations are used for each task.

B. Quantitative Results

1) ***Baseline Comparison****:* We first train five models independently for one-, two- and five-shot tasks on (1/2/5) randomly selected training mug pairs from Mug-Pour. Our method achieves 66%, 68%, and 78% success rates, in comparison to TAX-Pose's 53%, 55%, and 56% success rates (see Figure 5). Similarly on Mug-Rack, both models report a perfect success rate of 100% for the upright scenario, while for arbitrarily rotated mugs, our method achieves 78% in comparison with 41% for TAX-pose. In the survey, we report the mean score of our method is 4.17 ± 1.13, while for TAX-Pose — 2.45 ± 1.07. The outputs of our method are more aligned with the expected interaction. With a mean score of less than 5, we identify that there is still room for improvement.

2) ***Impact of Training Examples****:* Observing Figure 5, we show that having multiple examples that likely contain more different viewpoints of the mugs improves the model's performance. Our model's mean and standard error decreases as the number of demonstrations increases, suggesting that performance scales with data.

3) ***Impact of Semantic Features and Contact Threshold****:* In Figure 5, we show how the model performs with and without semantic features for Mug-Pour, one-shot from a single viewpoint. Our model with semantic features consistently performs better, significantly enhancing pose accuracy. Our ablation over varying the contact threshold, also in Figure 5, shows that for our full model in the Mug-Pour setting, performance improves with a bigger contact threshold. While this result may suggest that simply expanding the contact area is sufficient, the benefit relies on the presence of semantic correspondence. Without it, performance declines as the model fails to relate object parts to their roles in the task. The highest threshold yielding the best results also implies that optimal thresholds may vary across objects, depending on whether global context adds useful task-specific information.

C. Qualitative Results

1) ***Task Variety****:* To test the quality of the pose correspondence, we train our model on a single example of some interaction and show the predicted cross-pose for a novel object pair. This is tested on five real tasks (cup-pour, bottle-pour, drill, place, and stack), as well as five simulated tasks in PyBullet with objects from GSOD (mug on rack, shoe on rack, hat on toy, object in organizer, and plant in pot). See subsample in Figure 6. The real tasks are captured from a single viewpoint with a RS-D435 camera, which has notably noisy point-clouds. Note that the method does not imitate dynamic interactions, but only the pose correspondence. Our model produces good pose correspondences from a single example in many use cases, including when transferring to objects with different scales, geometric shapes, or textures. We observed that the correspondences are not ideal — while the predicted poses are close to what is intuitively expected, collisions between the novel objects are also observed. Moreover, one demonstration from a single viewpoint may be insufficient in providing enough task specificity.

2) ***Task Depth****:* The models are also qualitatively compared in mug pouring. For each mug in a set of $N = 9$ mugs, we provide a single demonstration. A model is trained on each example and tested on all N mugs. Placing more focus on geometric information, the baseline confuses between the top and bottom of the mug in some occurrences. On the other hand, some predicted tool poses from both models are in collision with the manipulanda.

3) ***Real Robot Demonstration****:* The models can predict good cross-poses across a variety of object pairs. However, they are lacking the precision needed for many real robot tasks — the proposed approach does not consider trajectory imitation. It can be directly used for simple manipulation tasks that involve cross-pose estimation. To validate that the presented visual projections can be successfully transferred

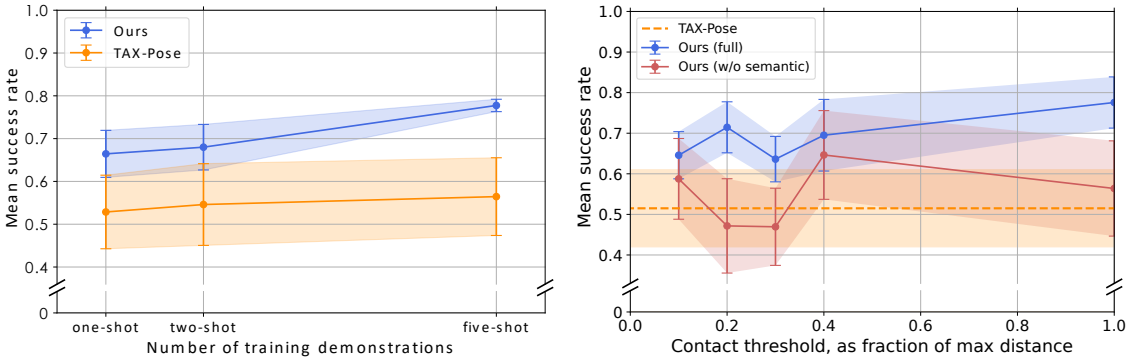

Fig. 5: Success rate evaluated on mug pouring. Each data point represents the mean of five models tested on 25 object pairs each. The shaded area represents the standard error. **Left:** Comparison with baseline for varying number of training demonstrations. **Right:** Ablation of semantic features and contact threshold parameter.

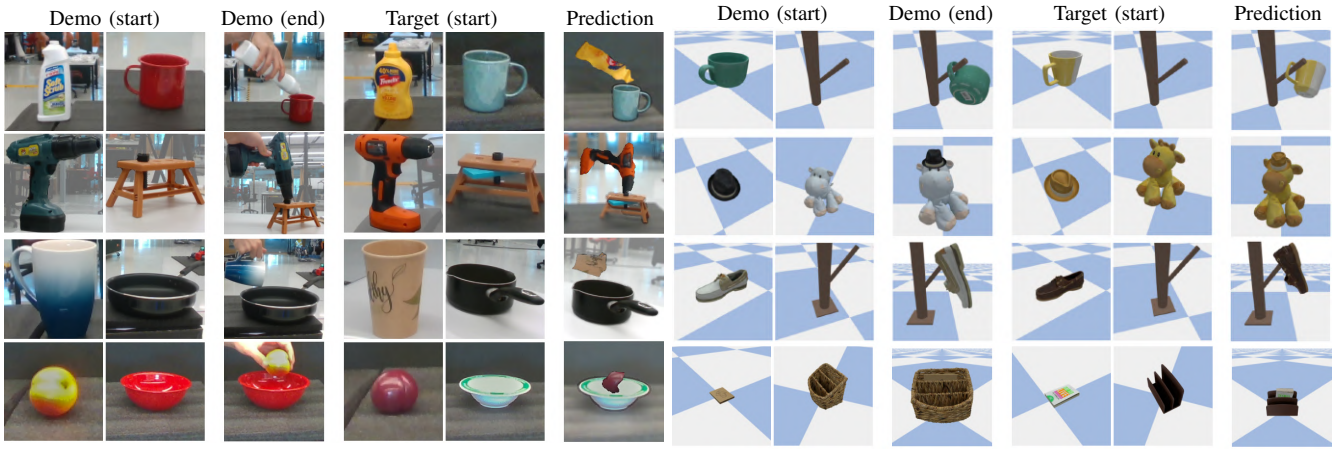

Fig. 6: (left) Real demonstrations and target scenes, with projections of the manipulator single-view mesh in the poses predicted by the model. The tasks in order: bottle pour in mug, drill on bolt, mug pour in saucepan, place fruit in bowl (right) Simulated demonstrations and target scenes, with simulated transformations of the poses predicted by the model. In order: mug on rack, hat on toy, shoe on rack, rectangular object in an organiser

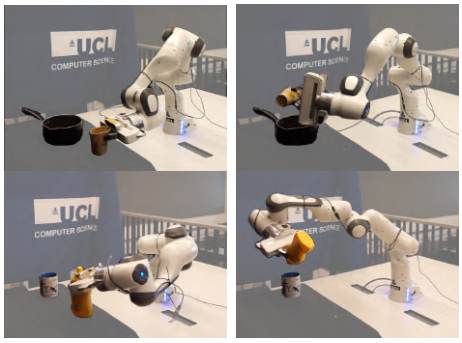

Fig. 7: The robot picks up a cup (top) and a bottle (bottom) and moves it to its predicted pouring pose.

to the real world, we present two of the tasks: pouring with bottle into a mug and pouring with mug into a saucepan, in Figure 7. We predict the interaction pose between the two observed objects, and execute a motion planned with RRT*.

VI. Conclusion

In both our quantitative and qualitative experiments, the predicted cross-poses of our approach were consistently better than the baseline. The proposed architecture alleviates the need for point clouds from multiple viewpoints by making use of semantic priors and the need for large-scale datasets. The method works directly with RGB-D sequences without ground truth object poses by leveraging an off-the-shelf object tracker, though accuracy depends on the tracker's precision. Our approach inherits TAX-Pose's main limitations: the need to train a new model for each task which may require significant memory resources as the number of tasks increases. Future work should explore multi-task approaches to cross-pose correspondence.

The cross-poses predicted by our model may serve as a good initial guess for skill initialization but they are not accurate enough for high-precision tasks. Future work stemming from this work should aim to expand its capabilities, by incorporating object trajectory imitation or learning fine-tuned policies with residual RL [43]; and address the limitations of the approach, e.g., by using neural rendering [44] instead of RGB-D cameras to represent the 3D world better and alleviate issues with representing reflective or transparent objects; and using novel architectures such as [45] to make the model SE(3)-equivariant. Moreover, applications such as assisted teleoperation of novel objects could be explored.

REFERENCES

[1] E. Johns, "Coarse-to-Fine Imitation Learning: Robot Manipulation from a Single Demonstration," in *ICRA*, 2021.

[2] N. Di Palo and E. Johns, "Learning Multi-Stage Tasks with One Demonstration via Self-Replay," in *CoRL*, 2021.

[3] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "VIOLA: Imitation learning for vision-based manipulation with object proposal priors," *CoRL*, 2022.

[4] S. Bahl, A. Gupta, and D. Pathak, "Human-to-Robot imitation in the wild," *RSS*, 2022.

[5] M. Sieb, Z. Xian, A. Huang, O. Kroemer, and K. Fragkiadaki, "Graph-structured visual imitation," in *CoRL*, 2020, pp. 979–989.

[6] A. Simeonov, Y. Du, A. Tagliasacchi, J. B. Tenenbaum, A. Rodriguez, P. Agrawal, and V. Sitzmann, "Neural Descriptor Fields: SE(3)-Equivariant Object Representations for Manipulation," *ICRA*, 2022.

[7] C. Pan, B. Okorn, H. Zhang *et al.*, "TAX-pose: Task-specific cross-pose estimation for robot manipulation," in *CoRL*, 2023.

[8] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *CoRL*, 2021.

[9] M. Ahn *et al.*, "Do as I can, not as I say: Grounding language in robotic affordances," in *arXiv preprint arXiv:2204.01691*, 2022.

[10] C. Michaelis, I. Ustyuzhaninov, M. Bethge, and A. S. Ecker, "One-Shot Instance Segmentation," *arXiv*, vol. abs/1811.11507, 2018.

[11] D. Hadjivelichkov and D. Kanoulas, "Fully Self-Supervised Class Awareness in Dense Object Descriptors," in *CoRL*, 2022.

[12] L. Huang, T. Hodan, L. Ma, L. Zhang, L. Tran *et al.*, "Neural correspondence field for object pose estimation," *ECCV*, 2022.

[13] B. Wen, W. Lian *et al.*, "CaTGrasp: Learning Category-Level Task-Relevant Grasping in Clutter from Simulation," *ICRA*, 2022.

[14] G. Zhang, G. Kang, Y. Yang, and Y. Wei, "Few-Shot Segmentation via Cycle-Consistent Transformer," in *NeurIPS*, 2021, pp. 21 984–21 996.

[15] J. Lee, D. Kim, J. Ponce, and B. Ham, "SFNet: Learning Object-aware Semantic Flow," in *CVPR*, 2019.

[16] S. Choudhury, I. Laina, C. Rupprecht, and A. Vedaldi, "Unsupervised Part Discovery from Contrastive Reconstruction," in *NeurIPS*, 2021.

[17] W.-C. Hung, V. Jampani, S. Liu, P. Molchanov *et al.*, "SCOPS: Self-Supervised Co-Part Segmentation," in *CVPR*, 2019.

[18] Q. Gao, B. Wang, L. Liu, and B. Chen, "Unsupervised Co-part Segmentation through Assembly," in *ICML*, 2021, pp. 3576–3586.

[19] D. Hadjivelichkov *et al.*, "One-shot transfer of affordance regions? AffCorrs!" in *CoRL*, 2023, pp. 550–560.

[20] M. Caron, H. Touvron, I. Misra *et al.*, "Emerging Properties in Self-Supervised Vision Transformers," in *ICCV*, 2021.

[21] S. Amir, Y. Gandelsman, S. Bagon, and T. Dekel, "Deep ViT Features as Dense Visual Descriptors," *arXiv preprint arXiv:2112.05814*, 2021.

[22] W. Goodwin, I. Havoutis, and I. Posner, "You only look at one: category-level object representations for pose estimation from a single example," in *CoRL*, 2023.

[23] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," *Arxiv*, 2024.

[24] W. Liu, J. Mao, J. Hsu, T. Hermans, A. Garg, and J. Wu, "Composable part-based manipulation," in *CoRL 2023*, 2023.

[25] M. Qin, J. Brawer, and B. Scassellati, "Rapidly learning generalizable and robot-agnostic tool-use skills for a wide range of tasks," *Frontiers in Robotics and AI*, vol. 8, 2021.

[26] E. Chun, Y. Du, A. Simeonov, T. Lozano-Perez, and L. Kaelbling, "Local neural descriptor fields: Locally conditioned object representations for manipulation," *arXiv preprint arXiv:2302.03573*, 2023.

[27] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *CVPR*, 2019.

[28] R. Agaram, S. Dewan, R. Sajnani, A. Poulenard, M. Krishna, and S. Sridhar, "Canonical Fields: Self-Supervised Learning of Pose-Canonicalized Neural Fields," in *CVPR*, 2023.

[29] T. Weng, D. Held, F. Meier, and M. Mukadam, "Neural grasp distance fields for robot manipulation," *arXiv preprint arXiv:2211.02647*, 2023.

[30] A. E. Tekden, M. Deisenroth, and Y. Bekiroglu, "Grasp transfer based on self-aligning implicit representations of local surfaces," *IEEE Robotics and Automation Letters*, vol. PP, pp. 1–8, 10 2023.

[31] O. Biza, S. Thompson, K. Pagidi, A. Kumar, E. van der Pol, R. Walters, T. Kipf, J.-W. Meent, L. Wong, and R. Platt, "One-shot imitation learning via interaction warping," 06 2023.

[32] V. Vosylius and E. Johns, "Few-shot in-context imitation learning via implicit graph alignment," in *Conference on Robot Learning*, 2023.

[33] N. D. Palo and E. Johns, "DINOBot: Robot manipulation via retrieval and alignment with vision foundation models," in *ICRA*, 2024.

[34] S. R. Lakani, A. J. Rodríguez-Sánchez, and J. Piater, "Exercising affordances of objects: A part-based approach," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3465–3472, 2018.

[35] B. Wen and K. E. Bekris, "BundleTrack: 6D pose tracking for novel objects without instance or category-level 3D models," in *IROS*, 2021.

[36] C. Park, S. Kim, J. Park, and M. Cho, "Learning so(3)-invariant semantic correspondence via local shape transform," in *CVPR*, 2024.

[37] Z. Chen and H. Xu, "Unsupervised semantic segmentation of 3d point clouds via cross-modal distillation and super-voxel clustering," *arXiv preprint arXiv:2304.08965*, 2023.

[38] N. S. Dutt, S. Muralikrishnan, and N. J. Mitra, "Diffusion 3d features (diff3f): Decorating untextured shapes with distilled semantic features," *arXiv preprint arXiv:2311.17024*, 2023.

[39] M. Qin, W. Li, J. Zhou, H. Wang, and H. Pfister, "Langsplat: 3d language gaussian splatting," *arXiv preprint arXiv:2312.16084*, 2023.

[40] Y. Wang and J. M. Solomon, "Deep Closest Point: Learning representations for point cloud registration," in *ICCV*, 2019.

[41] L. Downs, A. Francis, N. Koenig *et al.*, "Google Scanned Objects: A high-quality dataset of 3D scanned household items," in *ICRA*, 2022.

[42] H. Jun and A. Nichol, "Shap-E: Generating conditional 3D implicit functions," https://github.com/openai/shap-e, 2023.

[43] T. Johannink *et al.*, "Residual reinforcement learning for robot control," in *ICRA*, 2019, pp. 6023–6029.

[44] B. Mildenhall, P. P. Srinivasan, M. Tancik *et al.*, "NeRF: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[45] C. Deng, O. Litany, Y. Duan, A. Poulenard *et al.*, "Vector Neurons: a general framework for SO(3)-equivariant networks," *ICCV*.