Ditto in the House: Building Articulation Models of Indoor Scenes through Interactive Perception

Cheng-Chun Hsu[1] and Zhenyu Jiang[1] and Yuke Zhu[1]

***Abstract*— Virtualizing the physical world into virtual models has been a critical technique for robot navigation and planning in the real world. To foster manipulation with articulated objects in everyday life, this work explores building articulation models of indoor scenes through a robot's purposeful interactions in these scenes. Prior work on articulation reasoning primarily focuses on siloed objects of limited categories. To extend to room-scale environments, the robot has to efficiently and effectively explore a large-scale 3D space, locate articulated objects, and infer their articulations. We introduce an interactive perception approach to this task. Our approach, named *Ditto in the House*, discovers possible articulated objects through affordance prediction, interacts with these objects to produce articulated motions, and infers the articulation properties from the visual observations before and after each interaction. It tightly couples affordance prediction and articulation inference to improve both tasks. We demonstrate the effectiveness of our approach in both simulation and real-world scenes. Code and additional results are available at `https://ut-austin-rpl.github.io/HouseDitto/`**

I. Introduction

Virtualizing the real world into virtual models is a crucial step for robots to operate in everyday environments. Intelligent robots rely on these models to understand the surroundings and plan their actions in unstructured scenes. Recent advances in structure sensors and SLAM algorithms [9, 29, 30] have offered ways to construct static replicas of real-world scenes at unprecedented fidelity. These static replicas facilitate mobile robots to localize themselves and navigate around. Nevertheless, real-world manipulation would require a robot to depart from reconstructing a static scene to unraveling the physical properties of objects. In particular, to physically interact with articulation objects, such as cabinets and doors, commonly seen in daily environments, the robot needs to understand their kinematics and articulation properties. Motivated by this goal, this work aims to enable robots to automatically build articulated 3D models of indoor scenes from their purposeful interactions.

Understanding object articulation has been a long-standing challenge in computer vision and robotics. In recent years, a series of data-driven approaches have attempted to infer articulation from static visual observations by training models on large 3D datasets [22, 40]. These models rely on category-level priors, limiting their generalizations to a handful of preset categories. A parallel thread of research [13, 24] uses interactive perception [2]. They leverage physical interaction to emit visual observations of articulated motions from which they estimate the articulation properties. These works primarily focus on individual objects, whereas scaling to room-sized environments requires the robot to efficiently and effectively explore the large-scale 3D space for meaningful interactions.

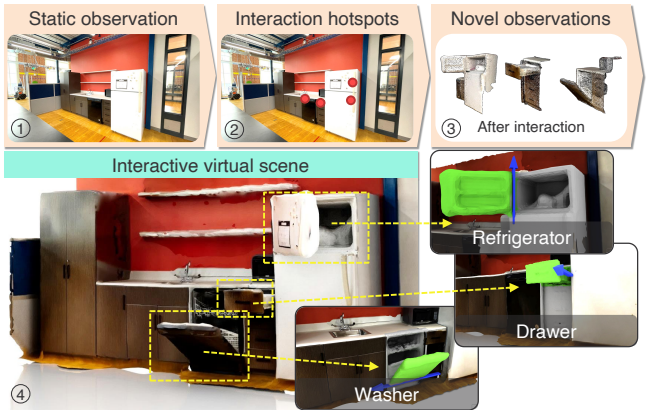

Fig. 1: **Building scene-level articulation models through interactive perception.** From an initial observation of an indoor scene, our approach infers the interaction hotspots, guiding the robot to interact with articulated objects. After that, the robot collects the observations before and after the interactions. Based on the observed articulated motions, it builds the articulation models of individual objects and aggregates them into a scene-level articulation model.

We introduce **Ditto in the House**, an approach to building the articulation model of an indoor environment through a robot's self-directed exploration. The robot discovers and physically interacts with the articulated objects in the environment. Based on the visual observations before and after the interactions, the robot infers the articulation properties of the interacted objects. Our approach requires the robot to identify regions of possible articulations in a large-scale 3D space, manipulate the objects to create articulated motions, and infer the underlying kinematic parameters from partial observations.

The foremost challenge is determining the most effective actions for probing the environment, *i.e.*, those most likely to discover articulations. We cast this problem as inferring scene affordance from visual observations [6, 26, 28]. Specifically, we train a model to predict affordance based on the robot's past interaction experiences with a large collection of procedurally generated 3D scenes in simulation. Furthermore, we introduce an iterative refinement procedure that uses the initial observations of articulated motions as visual cues to refine the affordance predictions. This refined affordance, in turn, guided the robot's subsequent interactions to obtain more informative observations for articulation inference.

Figure 1 illustrates the high-level steps of our approach.

[1] Department of Computer Science, the University of Texas at Austin. Correspondance to `chengchun@utexas.edu`

It discovers the interaction hotspots (*i.e.*, promising locations for probing actions), creates observations of articulated motions through action, and infers articulation from the observations. To locate interaction hotspots, we use an affordance model to predict an affordance map from the point cloud observation of an indoor scene. At the location of each interaction hotspot, the robot tries interacting with the objects and collects egocentric observations of the object before and after its motion induced by the robot's actions. Based on our prior work Ditto [15], we design an articulation estimation model to infer the object articulation from these observations. The articulation estimation model takes the robot's action and visual observations as input and predicts the articulation parameters of the objects. We tightly couple the affordance prediction and articulation estimation in an iterative process. We update the affordance prediction based on the initial estimation of mobile parts of the articulated object since these parts indicate actionable regions. The robot interacts with the object following the updated affordance, creating more visible articulation motions that facilitate articulation inference.

We evaluate our approach on the CubiCasa5K [17] dataset with the iGibson [21] simulator. Quantitative results demonstrate that our approach successfully discovered around 40% more parts than our baseline with higher precision. Our approach also gives a 45% absolute segmentation IoU boost compared with the baseline. Moreover, the ablation studies confirm that the iterative refinement process benefits both affordance prediction and articulation model estimation. Finally, we apply our approach to a real-world kitchen scene and successfully build an articulation model of this environment.

II. Related work

A. Visual Affordance Prediction

Predicting affordance [10] from visual observations is an essential ability for robots to plan their actions for interacting with real-world objects. A series of research learns visual affordance prediction from human videos [3, 7, 12, 27]. These videos reveal strong clues about how humans interact with their environment. However, affordances learned from these videos are often specific to human morphologies and fall short of generalizing to robot hardware. Another line of research acquires active training data with simulated or real robots for affordance learning [16, 19, 26, 28, 41]. In these methods, the robots explore diverse interactions with objects and scenes and learn affordances from their embodied experiences. Following this self-exploratory learning paradigm, we collect exploration data in simulation and train a visual affordance prediction model. This model guides our agent to discover articulated objects in the scene and interact with them for articulation inference.

B. Articulation Model Estimation

Articulation models represent the object parts and the kinematics (and sometimes dynamics) relationships between them. Earlier works model the object partonomy and articulation with probabilistic methods [4, 35–37]. They typically rely on markers or handcrafted features to track the mobile parts, limiting their practical applicability in natural environments. In recent years, deep learning methods [1, 22, 31, 32, 39, 40] have been employed for articulation estimation from raw sensory data. The majority of these works predict articulation models from single observations. They rely on category-level prior to estimating articulation parameters. For this reason, they are category-dependent and cannot generalize to diverse daily objects in indoor scenes. Another line of work leverages physical interaction to create novel sensory stimuli and infer the articulation model based on object state changes. Katz *et al*. [18] is the pioneer work that introduces interactive perception [2] for articulation model estimation. Some following works extend it with hierarchical recursive Bayesian filters [25], probabilistic models [38], and feature tracking [33]. Recent work in this direction utilizes the informative motion created by handcrafted strategies [15] or learned policy [8, 20, 23] and estimate articulation models with geometric deep learning techniques. The methods above are designed for individual objects. They cannot be easily extended to large indoor environments with multiple objects at unknown locations. In contrast, our approach develops a category-independent approach that actively explores the environment and builds scene-level articulation models.

III. Problem Formulation

We explore the problem of building an articulation model of an indoor scene populated with articulated objects. An articulated object consists of multiple parts, and their connecting joints constrain the relative motion between each pair of parts. We assume there exists an initial point cloud observation $P_s \in \mathbb{R}^{N_s \times 6}$ of the scene, where each point is a vector of its (x, y, z) coordinate and (R, G, B) color. N_s is the number of points. This work aims to segment the parts and estimate the articulation models M of the scene through interactive perception. Our approach actively interacts with the objects to estimate their articulation models.

For each object part, we identify the interaction hotspots [27], *i.e.*, the locations where the robot can successfully manipulate the articulated objects, and infer the articulation model. The articulation model represents the kinematic constraints between object parts. We parametrize articulation following ANCSH [22]. Prismatic joint parameters are the direction of the translation axis $u^p \in \mathbb{R}^3$ and the joint state $s^p \in \mathbb{R}$. Revolute joint parameters are the direction of the revolute axis $u^r \in \mathbb{R}^3$, a pivot point $q \in \mathbb{R}^3$ on the revolute axis, and the joint state $s^r \in \mathbb{R}$. The joint state s^p and s^r are the relative changes of the joint state before and after the interaction.

The process starts with finding interaction hotspots $H_s \in \mathbb{R}^{N_h \times 3}$ from an initial scene observation P_s, where N_h denotes the number of interaction hotspots in the scene. From the point cloud observation P_s, an affordance prediction model predicts an affordance map $A_s \in \mathbb{R}^{N_s}$ and samples its peak locations as interaction hotspots.

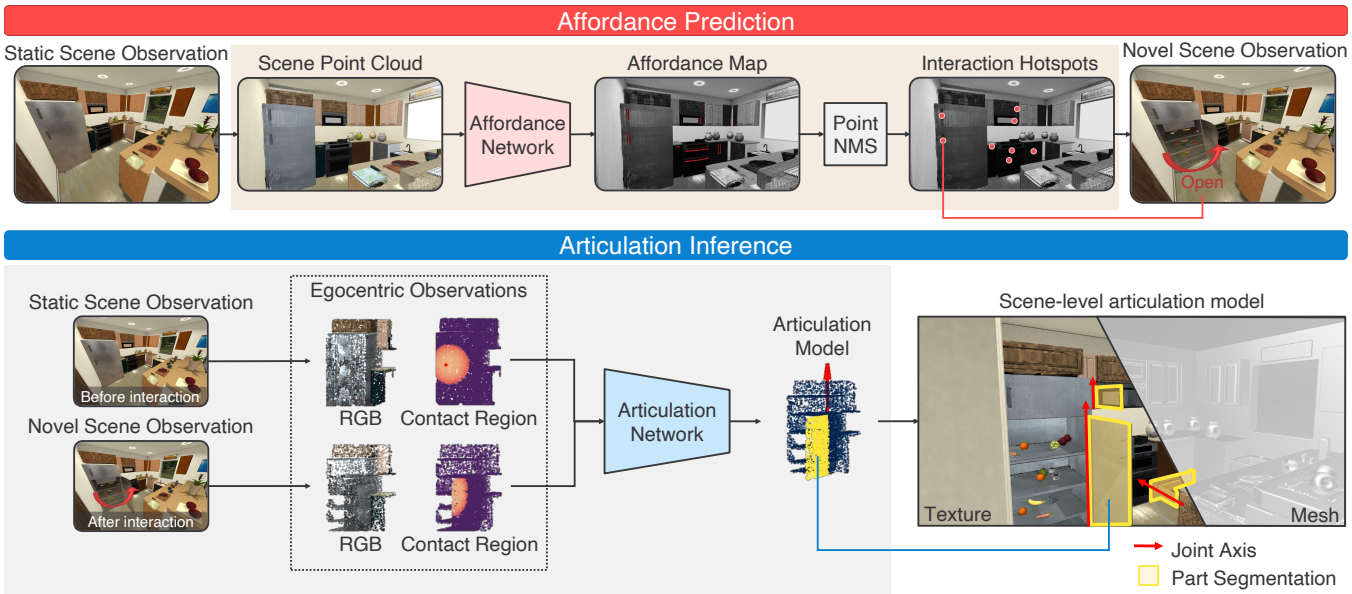

Fig. 2: **Overview of model components.** Our approach consists of two stages — affordance prediction and articulation inference. During affordance prediction, we pass the static scene point cloud into the affordance network and predict the scene-level affordance map. By applying point non-maximum suppression (NMS), we extract the interaction hotspots from the affordance map. Then, the robot interacts with the object based on those contact points. During articulation inference, we feed the point cloud observations before and after each interaction into the articulation model network to obtain articulation estimation. By aggregating the estimated articulation models, we build the articulation models of the entire scene.

At each interaction, the robot applies forces to the object hotspot to produce potential articulated motions. The robot captures two egocentric partial point clouds $P \in \mathbb{R}^{N_o \times 3}$ and $P' \in \mathbb{R}^{N_o' \times 3}$ that center on the interaction hotspot before and after the interaction, where N_o and N_o' denotes the number of points on object point clouds. The robot also records the corresponding contact locations $c, c' \in \mathbb{R}^3$. Given these object point clouds P, P' and contact locations c, c', an articulation inference model segments the point cloud into static and mobile parts and estimates a set of articulation parameters, *i.e.*, $\{u^p, s^p\}$ (prismatic) or $\{u^r, q, s^r\}$ (revolute), of the joints connecting two object parts. Finally, we map the estimated articulation model of each object from the global frame. The set of these articulation models constitutes the scene-level articulation model M.

IV. Approach

We now present our approach to building the articulation model of indoor scenes, as illustrated in Figure 2. In the following subsections, we introduce the three key components of our approach, affordance prediction, articulation estimation, and iterative refinement of both affordance and articulation.

A. Affordance Prediction

At the initial stage, we must identify potential interaction regions and discover the articulated objects. We leverage PointNet++ [34] to estimate scene affordance from an initial point cloud observation of the scene. Formulating affordance prediction as a point-wise binary classification problem, we feed the scene point cloud into PointNet++ and obtain a point-wise affordance map. Based on the map, the robot actively explores the scene and interacts with objects at the corresponding locations. Since dense affordance prediction may introduce too many potential locations to probe, we apply non-maximum suppression (NMS) [11] to extract peaks from each region as the *interaction hotspots*.

Our approach works as follows: first, we select the point with the maximum score and add it to the preserving set. Then we suppress its neighbors by a certain distance threshold. We repeat this process until all points are added to the preserving set or suppressed. The points left in the preserving set are the interaction hotspots. The robot physically interacts with objects at the corresponding location for each interaction hotspot. These interactions create informative motions for further articulation inference.

B. Articulation Inference

After each effective interaction, we observe changes in the articulated configurations of the object. The robot collects egocentric observations of the object before and after the interactions. We develop an articulation network to infer the articulation model from these observations.

The network is built on top of our prior work Ditto [15]. Given a pair of visual observations of an object before and after its articulated motion, Ditto simultaneously predicts the 3D geometry and articulation model of the object. Original Ditto takes the observations as input and does not know the actions that create the motion in the observation. In contrast, our agent physically interacts with the object and then collects the observations. So we incorporate the knowledge of the interaction and use the contact regions of the interaction as part of the input to our network. For each interaction pair, the robot captures the observation point clouds P and P' by interacting at the corresponding contact locations c and c'. We create Gaussian heatmaps centered around the c and c' over the point cloud P and P', respectively. These heatmaps represent the regions where interactions occur.

The locations of contact regions during interaction reveal a vital clue about the mobile part region and the underlying kinematic constraint. Therefore, we feed both the point cloud observations and the heatmaps of contact regions into the network for articulation inference.

Finally, the network outputs the articulation joint parameters, joint state, and part segmentation of each articulated object. To build the final articulation models, we aggregate the estimated object-level articulation models into a scene-level model.

C. Iterative Refinement of Affordance and Articulation

The inference of the articulation model primarily relies on the visual observations captured during the interactions. The estimated articulation model could have a higher accuracy if the observations covered significant articulation motions and a complete view of the object's interior. However, these observations may be partially occluded due to ineffective actions, results from imprecise affordance predictions and manipulation failures. Partially occluded observations lead to inaccurate articulated predictions. Empirically, we find that articulation estimation of a fully opened revolute joint, *e.g.*, $> 30°$, is more accurate in terms of angle error than one with an ajar joint.

In practice, we find that even a coarsely estimated articulation model could reveal useful clues about part-level affordance. The articulation model provides the estimated location of the joint axis and part segmentation, which we can use to predict the possible kinematic motions of the object parts. As we want to produce a larger motion of the object part, we update the affordance based on the motion predicted from the articulation model. Accordingly, we develop an iterative procedure of interacting with partially opened joints and refining the articulated predictions.

As shown in Figure 3, the estimated affordance at the initial stage did not consider the articulation model and spread uniformly over the surface of the mobile part. The extracted interaction hotspot can be close to the axis. This location and the hand-crafted action primitive produce less torque on the revolute axis. As a result, the robot fails to fully open the revolute joint. In our experiments, only 10% of the revolute joints are opened up to $> 30°$ during the initial interaction, which hinders the performance of articulation estimation.

To improve the articulation model inferred at the first stage, we exploit the potential motion information from the previous articulation model and extract the object-level affordance. We refine the affordance prediction by selecting a pair of locations and actions to produce the most significant articulation motion. Given the joint axis and part segment, we select the point in the predicted mobile part farthest from the joint axis as our next interaction hotspot. We set the force direction as the moment of the axis, *i.e.*, the cross-product between the axis direction and the projection from the interaction hotspot to the axis. Finally, we interact with the object following the updated interaction hotspots and force directions and collect new observations. The new observations are less occluded and result in an accurate articulation estimation.

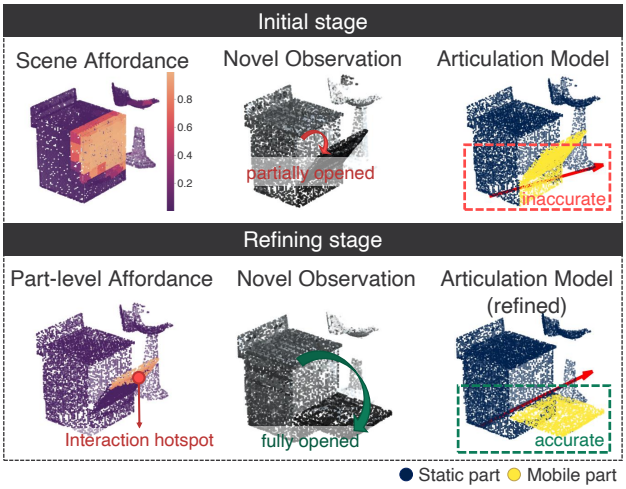

Fig. 3: **Iterative refinement of affordance and articulation.** In the initial stage, the object is partially opened due to the imprecise affordance prediction, which results in an inaccurate articulation estimation. In the refining stage, we refine object affordance based on the previous articulation estimation. The consequent new interaction fully opens the object and reveals the interior surface, thus improving the articulation estimation.

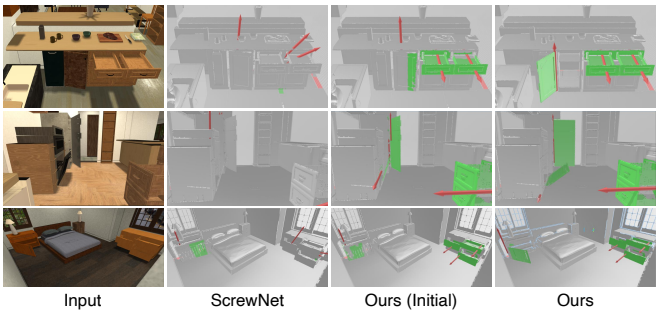

Fig. 4: **Qualitative results on virtual scenes.** Static parts are colored grey, and mobile parts green. The estimated joints are shown as red arrows.

V. Experiments

A. Experimental Setup

Dataset. We train our model with data collected in the simulation environment. Specifically, we conduct experiments on the scenes provided by CubiCasa5K [17] dataset with the iGibson [21] simulator. CubiCasa5K is a large-scale floorplan dataset covering over 5,000 scenes and 80 object categories, such as refrigerators and cabinets. We sample 2,500 scenes from the dataset and split them into (1,500, 500, 500) scenes for training, validation, and testing. We exclude objects with unopenable mobile parts due to collisions or other simulation issues. After the cleaning, our testing set covers 1,030 objects with 1,736 mobile parts.

Exploration-Driven Data Collection. We collect affordance supervision through robots' interactions. We first uniformly sample locations over the surface of both articulated and non-articulated parts, then have the robot interact with them. If the robot successfully moves any articulated part of the objects, we label the corresponding location as positive affordances or negative otherwise. To simulate gripper-based interactions, we perform a collision check at each location

TABLE I: Quantitative results of affordance prediction.

Method	Precision	Coverage	
		Prismatic	Revolute ($> 15°$, $> 30°$)
3D AffordanceNet [5]	0.56	0.23	(0.26, 0.03)
Ours (w/o Refinement)	0.66	0.60	(0.72, 0.10)
Ours	-	0.60	(0.72, 0.55)

to ensure enough space for placing a gripper. After that, we generate a pseudo-link between the object parts and the robot and then let the robot pull toward a certain direction, *i.e.*, left, right, and backward. For simplification, we leverage a simplified virtual robot that can perform similar interactions as the real robot in a simulated environment. The virtual robot is a floating ball that makes single-point contact and can be teleported to any location in the scene. Such simplification lets us focus on the perception and interaction aspects without considering motion planning or navigation. For each successful interaction, we collect the robot's egocentric observations before and after interaction and the object's articulation model as training supervision.

B. Training Details

Affordance Network. We collect affordance data by the aforementioned exploration-driven method. Given the scene point cloud, we randomly sample points that fall on objects as potential interaction hotspots. By physically interacting with them, we label the potential interaction hotspots as *positive* or *negative* affordance depending on the interaction outcomes of *success* versus *failure*. The remaining points that have yet to be interacted with are ignored during training. The network takes the point clouds as input and is supervised under the affordance label. The data distribution is imbalanced due to the large proportion of negative data. To mitigate the imbalance problem, we optimize the network with the combination of the cross-entropy loss and the dice loss, as used in the previous work [5].

Articulation Network. To collect object-centric observations before and after the interaction, the robot captures multi-view observations of the object at the front, right, and left sides of the object at a fixed distance if no collision occurs. After that, we aggregate them as a partial point cloud assuming the ground-truth camera poses are available. The network takes partial point clouds as input and is supervised by the ground truth articulation parameters provided by the simulator. We set up the loss functions and other training details following Ditto [15], except that the occupancy decoder is discarded.

C. Evaluation Metrics

Affordance. To evaluate the affordance network, we report two metrics—coverage and precision. Coverage is the proportion of successfully interacted parts among the total number of interactable parts, and precision is the fraction of successful interactions that the agent attempted. For coverage of the revolute joint, we further define certain open degrees as the thresholds for successful interactions. Note that we do not define additional thresholds for prismatic joints since most prismatic joints are fully opened without refinements.

TABLE II: Quantitative results of articulation inference.

Method	Geometry	Joint		
	Mobile Seg. IoU ↑	Prismatic	Revolute	
		Angle Err. ↓	Angle Err. ↓	Pos Err. ↓
ScrewNet [14]	0.34	0.28	46.19	0.13
Ours (w/o Refinement)	0.78	0.04	49.0	0.06
Ours (w/o Regularity)	0.78	0.05	31.0	0.05
Ours	0.81	0.04	25.2	0.04

Articulation Model. To evaluate the estimated articulation model, *i.e.*, prismatic/revolute joint parameters and part segmentation, we use the same metrics as in Ditto [15]. For both types of joints, we measure point-wise segmentation IoU of the mobile parts (Mobile Seg. IoU) and joint axis orientation error (Angle Err.). For the revolute joint, the position of the joint axis is also important, so we measure the position error (Pos Err.) by the distance between the predicted and ground-truth rotation axis.

D. Baselines

3D AffordanceNet [5]. To validate the efficacy of exploration-driven data collection, we compare the affordance labels collected from the robot's first-hand embodiment experience and manual affordance annotations. 3D AffordanceNet is a benchmark for visual object affordance understanding. It provides 3D affordance maps annotated by humans. We train an affordance model with the same architecture as ours using the ground-truth affordance map from 3D AffordanceNet and evaluate it on our test scenes.

ScrewNet [14]. To validate our joint parametrization choices, we modify our model's output and adopt the screw-based joint parametrization in ScrewNet [14]. ScrewNet uses screw theory to unify the representation of different articulation types and perform category-independent articulation model estimation. While we follow Ditto [15] and predict point-wise dense joints, ScrewNet predicts one global joint.

Ours (w/o Refinement). We test an ablated version of our approach, where we only use the results from the initial stage. Compared with our complete approach, Ours (w/o Refinement) does not refine the affordance based on observed articulated motion and no further refinement of the articulation model. This comparison helps validate the iterative refinement of affordance and articulation.

Ours (w/o Regularity). We leverage action perception regularities by incorporating the contact regions of the interaction into our articulation network. In this ablated version of our approach, we remove the contact region information during training and inference. We present this result to validate the regularities for articulation inference.

E. Affordance Prediction

The quantitative results of affordance prediction are shown in Table I. Ours (w/o Refinement) obtains significantly better results than the model trained on 3D AffordanceNet. Human annotated affordance is biased towards human experience and might not generalize to robot manipulation. The distribution shift between the annotated scenes and the deployment environments also introduces extra challenges

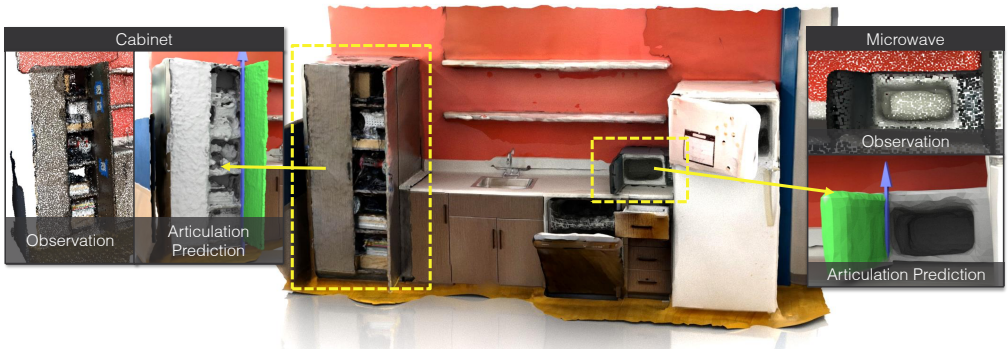

Fig. 5: **Reconstructed articulation model of the real scene.** Static parts are colored grey, and mobile parts are green. The estimated joints are visualized with blue arrows. More results on the same scene are shown in Figure 1.

for generalization. In comparison, we train our affordance model with the data collected through self-exploration. The data is self-annotated based on the outcome of the robot's action executed in the environment. Therefore the trained model can easily adapt to the test scenes and shows much better interaction performance.

Our final model, with the iterative refinement of affordance and articulation, further boosts the coverage performance of revolute joints. The percentage of revolute joints opened over 30° increases from 10% to 55%. This improvement verifies the iterative refinement scheme. The initial estimation of the articulation model provides vital clues about the part-level affordance. The inferred part-level affordance creates more visible articulation motions and opens the parts with revolute joints to a larger degree. Note that our iterative refinement process does not add new or remove existing interaction hotspots. Therefore, we do not report the precision result after the refinement.

F. Articulation Inference

We show the results of articulation estimation in Table II. ScrewNet-based model [14] performs poorly on articulation estimation, especially the part segmentation. Ours (w/o Regularity) does not leverage the interaction information on where the action occurs and thus performs inferior to our complete model in part segmentation and joint axis prediction. In Figure 4, we see that the ScrewNet-based model hardly segments any mobile parts. The joint axis predictions of this baseline are also far from correct. Ours (w/o Refinement) demonstrates much better results on object parts with prismatic joints. However, the result on the cabinet door with a revolute joint is less accurate. Ours (w/o Refinement) only segments less than half of the door, and the position of the estimated revolute joint axis also deviates from the correct location. Our refined model first opens the cabinet to a larger degree and reveals more previously occluded surfaces. With the new observation with more significant object state change, our refined model can predict more accurate part segmentation and joint parameters.

G. Real-World Evaluation

Finally, we evaluate our method in a real-world household scene. We use the LiDAR and camera of an iPhone 12 Pro to recreate the scene in a 3D scan, rather than using a physical robot. We predict interaction hotspots and interact with the objects at these hotspots with our own hands. We then collect novel observations and run our approach to build the scene-level articulation model. The results in Figure 1 and Figure 5 show that our approach can be applied to the real scenario without any modification and reconstruct an accurate articulation model of the scene.

H. Limitations

We use the simulated robot grippers and human interactions to simplify the exploration and object manipulation process in virtual and real-world experiments. Thus, this work has the following limitations: a) *Exploration:* To move between different locations, we directly teleport the robot. We abstract away the navigation and motion planning problems. Moreover, we assume perfect odometry and depth estimation while reconstructing the static model; b) *Interaction:* We perform all interactions by creating pseudo links between the robot and objects or by humans. Issues such as joint constraints or self-collisions of the robot are not taken into consideration during object manipulation.

VI. Conclusion

We develop an interactive perception approach to building scene-level articulation models. The robot physically interacts with articulated objects and infers their articulation properties from visual observations before and after exploratory actions. We further improve the quality of our prediction by coupling affordance prediction and articulation inference in an iterative procedure. Quantitative results demonstrate that our approach outperforms baselines by a substantial margin in both affordance prediction and articulation estimation. The ablation studies confirm that the iterative refinement process improves both tasks. Last, we demonstrate that our approach generalizes to real-world observations for creating an articulation model of a kitchen scene. These results manifest the promise of our approach in building interactive models for robot manipulation in everyday environments.

Acknowledgments We would like to thank Huihan Liu, Mingyo Seo, and Soroush Nasiriany for providing feedback on this manuscript. This work has been partially supported by NSF CNS-1955523 and FRR-2145283, the MLL Research Award from the Machine Learning Laboratory at UT-Austin, and the Amazon Research Awards.

REFERENCES

[1] H. Abdul-Rashid, M. Freeman, B. Abbatematteo, G. Konidaris, and D. Ritchie, "Learning to infer kinematic hierarchies for novel object instances," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8461–8467.

[2] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme, "Interactive perception: Leveraging action in perception and perception in action," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1273–1291, 2017.

[3] S. Brahmbhatt, C. Ham, C. C. Kemp, and J. Hays, "Contactdb: Analyzing and predicting grasp contact via thermal imaging," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 8709–8719.

[4] A. Dearden and Y. Demiris, "Learning forward models for robots," in *IJCAI*, vol. 5, 2005, p. 1440.

[5] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3d affordancenet: A benchmark for visual object affordance understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1778–1787.

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

[7] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic, "People watching: Human actions as a cue for single view geometry," in *European Conference on Computer Vision*, Springer, 2012, pp. 732–745.

[8] S. Y. Gadre, K. Ehsani, and S. Song, "Act the part: Learning interaction strategies for articulated object part discovery," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 752–15 761.

[9] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *2011 IEEE intelligent vehicles symposium (IV)*, Ieee, 2011, pp. 963–968.

[10] J. J. Gibson, "The ecological approach to visual perception: Classic edition," in *Psychology Press*, 1979.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[12] H. Hamer, J. Gall, T. Weise, and L. Van Gool, "An object-dependent hand pose prior from sparse training data," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 671–678.

[13] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme, "Active articulation model estimation through interactive perception," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2015, pp. 3305–3312.

[14] A. Jain, R. Lioutikov, and S. Niekum, "Screwnet: Category-independent articulation model estimation from depth images using screw theory," *arXiv preprint arXiv:2008.10518*, 2020.

[15] Z. Jiang, C.-C. Hsu, and Y. Zhu, "Ditto: Building digital twins of articulated objects from interaction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5616–5626.

[16] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *arXiv preprint arXiv:2104.01542*, 2021.

[17] A. Kalervo, J. Ylioinas, M. Häikiö, A. Karhu, and J. Kannala, "Cubicasa5k: A dataset and an improved multi-task model for floorplan image analysis," in *Scandinavian Conference on Image Analysis*, Springer, 2019, pp. 28–40.

[18] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz, "Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects.," in *ICRA*, 2013, pp. 5003–5010.

[19] M. Khansari, D. Kappler, J. Luo, J. Bingham, and M. Kalakrishnan, "Action image representation: Learning scalable deep grasping policies with zero real world data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 3597–3603.

[20] K. N. Kumar, I. Essa, S. Ha, and C. K. Liu, "Estimating mass distribution of articulated objects using non-prehensile manipulation," *arXiv preprint arXiv:1907.03964*, 2019.

[21] C. Li, F. Xia, R. Martı'n-Martı'n, M. Lingelbach, S. Srivastava, B. Shen, K. Vainio, C. Gokmen, G. Dharan, T. Jain, *et al.*, "Igibson 2.0: Object-centric simulation for robot learning of everyday household tasks," *arXiv preprint arXiv:2108.03272*, 2021.

[22] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song, "Category-level articulated object pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3706–3715.

[23] J. Lv, Q. Yu, L. Shao, W. Liu, W. Xu, and C. Lu, "Sagci-system: Towards sample-efficient, generalizable, compositional, and incremental robot learning," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 98–105.

[24] R. Martı'n-Martı'n and O. Brock, "Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2014, pp. 2494–2501.

[25] R. Martı'n-Martı'n, S. Höfer, and O. Brock, "An integrated approach to visual perception of articulated objects," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 5091–5097.

[26] K. Mo, L. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, "Where2act: From pixels to actions for articulated 3d objects," *arXiv preprint arXiv:2101.02692*, 2021.

[27] T. Nagarajan, C. Feichtenhofer, and K. Grauman, "Grounded human-object interaction hotspots from video," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8688–8697.

[28] T. Nagarajan and K. Grauman, "Learning affordance landscapes for interaction exploration in 3d environments," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2005–2015, 2020.

[29] R. A. Newcombe, D. Fox, and S. M. Seitz, "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 343–352.

[30] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE international symposium on mixed and augmented reality*, IEEE, 2011, pp. 127–136.

[31] A. Noguchi, X. Sun, S. Lin, and T. Harada, "Neural articulated radiance field," *arXiv preprint arXiv:2104.03110*, 2021.

[32] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *2018 international conference on 3D vision (3DV)*, IEEE, 2018, pp. 484–494.

[33] S. Pillai, M. R. Walter, and S. Teller, "Learning articulated motions from visual demonstration," *arXiv preprint arXiv:1502.01659*, 2015.

[34] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[35] J. Sturm, C. Plagemann, and W. Burgard, "Unsupervised body scheme learning through self-perception," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, 2008, pp. 3328–3333.

[36] J. Sturm, C. Plagemann, and W. Burgard, "Adaptive body scheme models for robust robotic manipulation.," in *Robotics: Science and systems*, Zurich, 2008.

[37] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard, "Learning kinematic models for articulated objects," in *Twenty-First International Joint Conference on Artificial Intelligence*, 2009.

[38] J. Sturm, C. Stachniss, and W. Burgard, "A probabilistic framework for learning kinematic models of articulated objects," *Journal of Artificial Intelligence Research*, vol. 41, pp. 477–526, 2011.

[39] X. Wang, B. Zhou, Y. Shi, X. Chen, Q. Zhao, and K. Xu, "Shape2motion: Joint analysis of motion parts and attributes from 3d shapes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8876–8884.

[40] Y. Weng, H. Wang, Q. Zhou, Y. Qin, Y. Duan, Q. Fan, B. Chen, H. Su, and L. J. Guibas, "Captra: Category-level pose tracking for rigid and articulated objects from point clouds," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 13 209–13 218.

[41] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS),
IEEE, 2018, pp. 4238–4245.