3D AffordanceNet: A Benchmark for Visual Object Affordance Understanding

Shengheng Deng[1,*], Xun Xu[2,*], Chaozheng Wu[1], Ke Chen[1,4] and Kui Jia[1,3,4,†]
[1]South China University of Technology, [2]I2R, A-STAR
[3]Pazhou Laboratory, [4]Peng Cheng Laboratory

Abstract

The ability to understand the ways to interact with objects from visual cues, a.k.a. visual affordance, is essential to vision-guided robotic research. This involves categorizing, segmenting and reasoning of visual affordance. Relevant studies in 2D and 2.5D image domains have been made previously, however, a truly functional understanding of object affordance requires learning and prediction in the 3D physical domain, which is still absent in the community. In this work, we present a 3D AffordanceNet dataset, a benchmark of 23k shapes from 23 semantic object categories, annotated with 18 visual affordance categories. Based on this dataset, we provide three benchmarking tasks for evaluating visual affordance understanding, including full-shape, partial-view and rotation-invariant affordance estimations. Three state-of-the-art point cloud deep learning networks are evaluated on all tasks. In addition we also investigate a semi-supervised learning setup to explore the possibility to benefit from unlabeled data. Comprehensive results on our contributed dataset show the promise of visual affordance understanding as a valuable yet challenging benchmark.

1. Introduction

The concept of affordance was first defined as what the environment offers the animal, introduced by [6]. Affordance understanding is concerned with the interactions between human and environment. For instance, human can sit on the chair, grasp a cup or lift a bag. Being able to understand the affordance of objects is crucial for robots to operate in dynamic and complex environments [8]. Many applications are supported by affordance understanding including, anticipating and predicting future actions[12, 10, 13], recognizing agent's activities[21, 4, 26], providing valid functionality of the objects[7], understanding social scene situations[2] and understanding the hidden values of the objects[33]. Tasks including affordance categorization, reasoning, semantic labeling, activity recognition, etc. are defined as specific instantiations of affordance understanding [8]. Among all these we find semantic labeling [22, 33] is of the most importance because the ability to localize the position of possible affordance is highly desired by robotic research. We refer semantic labeling as affordance estimation throughout this paper.

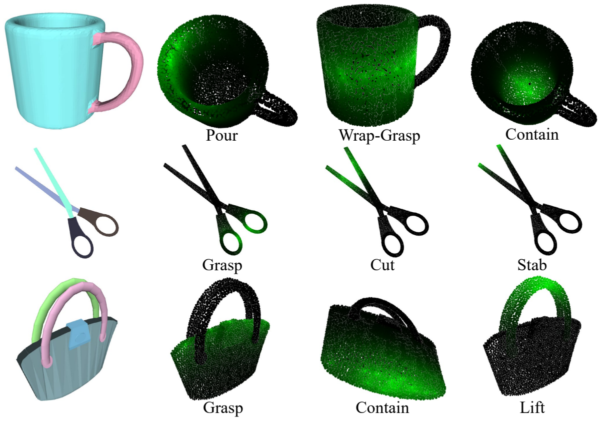

Figure 1. The 3D AffordanceNet dataset. The mesh was first annotated with affordance keypoints. Then we densely sample points and obtain the ground truth data via label propagation.

The most important and proper modality for affordance understanding is through visual sensors [8]. Visual affordance understanding has been extensively studied recently with computer vision techniques. Many algorithms are built upon deep neural networks [19, 3, 23] thus require large labeled affordance dataset for benchmarking. Relevant datasets are developed for these purposes with data collected from 2D (RGB) sensors [32, 24, 22] or 2.5D (RGBD) sensors [18, 19, 23]. Nevertheless, we believe that the affordance understanding requires learning in the 3D domain which conveys the geometric properties. For example, the affordance of grasp is highly correlated with vertical structure with small perimeter and sittable is correlated with flat surface. Unfortunately, such detailed geometry is not captured by the existing 2D datasets while the 2.5 ones [19, 23] are often captured with small depth variation and do not carry enough geometric information.

[*] indicates equal contribution.
[†]Correspondence to Kui Jia <kuijia@scut.edu.cn>.

To encourage research into visual affordance understanding in more realistic scenarios, a benchmark on real 3D dataset is highly desired. Therefore, we are inspired by PartNet[17], a recently proposed dataset containing the fine-grained part hierarchy information of 3D shapes based on the large-scale 3D CAD model dataset ShapeNet[1] and 3D Warehouse. Although PartNet mentioned affordance as potential application, there is still no benchmark purposely established for affordance yet. More importantly, we discover, via user annotations, that the human perceived affordance often do not fully overlap with the individual parts specified in PartNet dataset. For example, In the first row of Fig. 1, the *Pour*, *Wrap-Grasp* and *Contain* affordance from *Mug* do not perfectly match any part indicated by the colored image on the 1st column. Therefore, we believe it is necessary to provide a new set of affordance labels on the PartNet dataset.

Creating 3D visual affordance benchmark is challenging due to the subjective definition. We take into account the affordance definitions from existing research on visual affordance learning in 2D and 2.5D domains [8] and select possible interactions that one can take with 3D shapes from PartNet. Finally, 18 types of affordance were formally defined over 23 semantic objects. Additional challenge associated with annotation on 3D model is the scalability issue. In order to provide highly quality annotation on such a large scale, we use label propagation method to propagate affordance sparsely labeled on individual points. Eventually, we obtain point-wise probabilistic score of affordance for each individual shape in PartNet. We name the new benchmark *3D AffordanceNet* to reflect the focus on visual affordance on 3D point cloud data.

3D AffordanceNet enables benchmarking a diverse set of tasks, in particular, we put forward full-shape, partial-view and rotation-invariant affordance estimations. Three state-of-the-art point cloud deep learning networks are evaluated on all tasks. We also propose a semi-supervised affordance estimation method to take the advantage of large amount of unlabeled data for affordance estimation.

In summary, we make the following contributions:

- We introduce 3D AffordanceNet, consisting of 56307 well-defined affordance information annotations for 22949 shapes covering 18 affordance classes and 23 semantic object categories. To the best of our knowledge, this is the first *large-scale* dataset with *well-defined* probabilistic affordance score annotations;
- We propose three affordance learning tasks which are supported by 3D AfffodanceNet to demonstrate the value of annotated data: full-shape affordance estimation, partial-view affordance estimation and rotation-invariant affordance estimation.
- We benchmark three baseline methods for proposed affordance learning tasks and further propose a semi-supervised affordance estimation method to take advantage of unlabeled data for affordance estimation.

2. Related Work

Affordance refers to the possible action an agent could make to interact with the environment [6]. Examples include a cup can afford '*pouring*', a bed is '*sittable*' and '*layable*', *etc*. Affordance understanding is the core function in developing autonomous systems. In particular, the visual affordance understanding is the most promising way due to the rich information carried by visual sensors. We mainly review the recent development in visual affordance dataset and approaches, a detailed review can be found in [8]. Recent advances in visual affordance are mostly demonstrated on affordance recognition [7, 3], detection [3, 18] and segmentation [3, 18]. Beyond the low-level visual tasks, there is substantial attention on-the-rise paid to affordance reasoning, affordance-based activity recognition and social affordances. In this work we are interested in providing a benchmark for affordance segmentation, a.k.a. prediction, due to the clear definition and high demand in robotic applications.

To benchmark visual affordance segmentation, UMD [18], CAD120 [23] and IIT-AFF [19] are respectively developed recently. All datasets feature affordance segmentation on RGBD images covering from 10-20 objects, 6-9 affordance types and 3k-10k labelled images. In particular, IAF-IIT and CAD120 capture complex scenes while UMD mainly focuses on well-controlled scenes. Nevertheless, none of these datasets carry the rich geometric properties of objects that robotic application would expect and only a single view-point is present. As a result, these datasets no longer pose challenges to modern computer vision techniques.

With the easy access to 3D point cloud data, *e.g*. collected from LiDAR and SFM, and potential application in robotics, atunomous driving, *etc*., there is a recent surge in research towards 3D point cloud. ShapeNet [1] collected 3D CAD models from open-sourced 3D repositories, with more than 3,000,000 models and 3,135 object categories. It was further developed by [29] for shape part segmentation. Partially motivated by affordance understanding, the subsequent PartNet dataset [17] was proposed with 26k objects, featuring fine-grained semantic segmentation task. Hierarchical segmentation was also addressed by a recursive part decomposition [31]. Though affordance is briefly mentioned as the motivation for creating above 3D shape datasets, to the best of our knowledge, there is no existing dataset which explicitly addresses the task of visual affordance prediction. The only known attempt on 3D shape functionality understanding [9] is still limited to a few types

https://3dwarehouse.sketchup.com

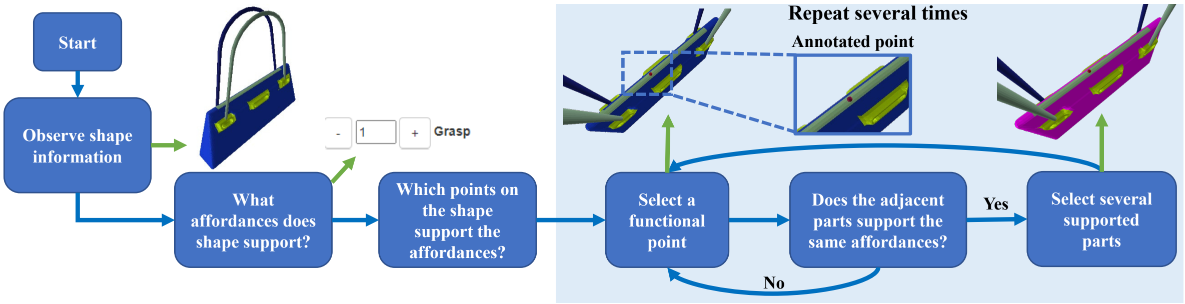
Start
Observe shape information
What affordances does shape support?
- 1 + Grasp
Which points on the shape support the affordances?
Repeat several times
Annotated point
Select a functional point
Does the adjacent parts support the same affordances?
Yes
No
Select several supported parts

Figure 2. The annotation workflow. The blue arrows indicate the annotation procedures, the green arrows refer to the corresponding 3D GUI actions. The annotators are first asked to determine the supported affordance classes and then select the functional points. The annotators need to confirm whether the adjacent parts support the same affordances.

of objects and did not make connection to the well-studied visual affordance understanding in 2D image domains. In contrast, we created a new benchmark for visual affordance estimation on 3D point cloud. The affordance types are selectively inherited from a summary of existing works and annotations are made on 3D point cloud data directly.

3. Dataset Construction

We present 3D AffordanceNet as a dataset for affordance estimation 3D point cloud. To construct this dataset, we first define a set of affordance types by referring to the existing visual affordance works [8]. Raw 3D shape data are collected from the shapes in PartNet [17] which covers common object types in typical indoor scenes. A question-answering 3D GUI is developed to collect raw point-wise annotation on mesh shapes. In total, we hired 42 professional annotators for annotations, the average annotation time per shape is 2 minutes and each shape is annotated by 3 annotators.Finally, label propagation is employed to obtain probabilistic ground-truth for the shape point cloud.

3.1. Affordance Type

We refer to [8] for a full review of affordance types adopted in visual affordance research. From the full list of possible affordances, we select those suitable for 3D objects present in PartNet [17] and remove the irrelevant ones, e.g. ‘reachable’ and ‘carryable’. Overall, we filter out 18 categories of affordances, namely ‘grasp’, ‘lift’, ‘contain’, ‘open’, ‘lay’, ‘sit’, ‘support’, ‘wrap-grasp’, ‘pour’, ‘display’, ‘push’, ‘pull’, ‘listen’, ‘wear’, ‘press’, ‘cut’, ‘stab’, and ‘move’. Then, we associate the affordance types to each category of object in PartNet according to its attributes and functionality that it can afford to interact with human or robot. For example, a chair is ‘sittable’ but not ‘layable’, a table can afford ‘support’ but not ‘contain’, *etc*. The affordances of each category are shown in Tab. 2. The annotators are allowed to freely determine where the affordance locates on the object, *e.g*. ‘grasp’ of bag can be annotated at its handle, webbing or straps. Notice that we allow the annotators to select the supported affordance for each shape, thus some shapes may not have all affordances defined for its own shape category.

3.2. Annotation Interface

We created a web-based 3D GUI to collect raw annotations. The process of annotation is designed to be a question-answering workflow as illustrated in Fig. 2. A user is given one shape at a time visualized in 3D. Each individual parts are colored according to the pre-defined colormap in PartNet dataset [17]. Annotators are allowed to freely rotate, translate and change the scale of the shape using mouse, which allow the annotators to observe the shape from more angles. After observation, annotators are first asked to determine the supported affordances by choosing from a list (‘What affordances does this shape support?’). Considering that some annotators may not understand the affordances, we provide the explanation of each affordance on the interface. Annotators are then asked to select keypoints that support the specified affordance (‘What points on the shape support the affordance?’). At least 3 keypoints will be labeled by one annotator for each affordance. Annotators will also decide whether the selected affordance will propagate beyond the part which the labeled keypoint sits on. If yes, annotators are asked to select eligible parts that the affordance can propagate to, otherwise, more annotations will be made on the same part until enough keypoints are collected.

The questions that the annotation interface proposes for each affordance directly determine how the annotators perceive the affordances. Therefore, we define questions carefully tailored for each affordance. Some questions are shown in Tab. 1. A complete list of affordance question is given in the supplementary material.

3.3. Ground-Truth Construction

After obtaining affordance keypoints, we propagate labels to all points on the shape to create ground-truth for downstream learning tasks. We first record the coordinates of the selected keypoints. We then propagate the labels to N points densely sampled on the shape mesh surface, note

Affordance	Object	Question
Lay	Bed	If you were to lie on this bed, which points would you lie on the bed?
Grasp	Earphone	If you want to grab this earphone, where will your palm position be?
Lift	Bag	If you want to lift this bag, at which points are your finger most likely to carry the bag?
Sit	Chair	If you were sitting on this chair, on which points would you sit?
Move	Table	If you want to move this table, at which points on this table will you exert your strength?
Open	Trash Can	If you want to open the lid of this trash can, from which points on the trash can you open it?
Pour	Bottle	Suppose there is water in the bottle, and you want to pour the water out of the bottle. From which points on the bottle will the water flow out?
Press	Laptop	If you want to press keys on a computer keyboard, which points on the keyboard would you press?
Contain	Microwave	If you put something in the microwave, at which points in the microwave would you put the object?
Support	Table	If you want to put something on the table, at which points on the table would you put the object?

Table 1. Some examples of the proposed questions for affordance annotation.

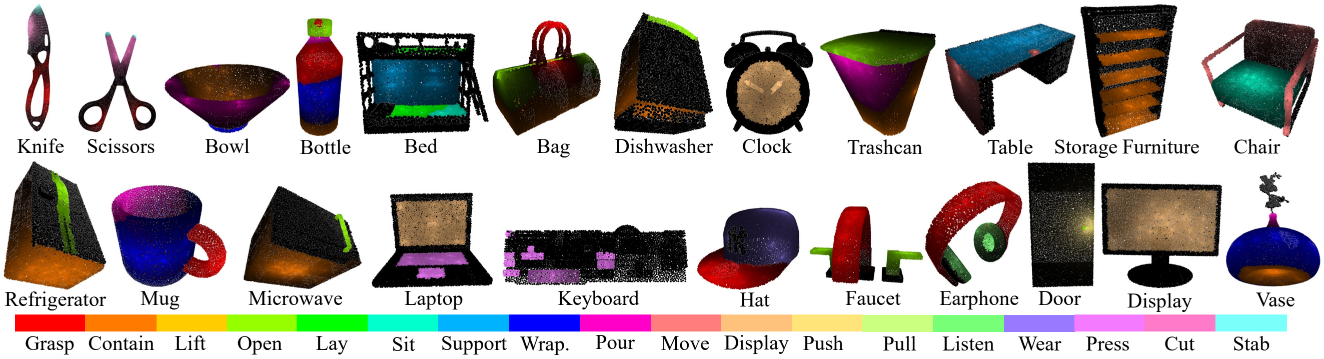

Figure 3. The example of annotated data. Different affordances are shown in different colors, points annotated with multiple affordances are colored by the affordance that has the highest scores. The brighter the color, the higher the score.

that we only propagate on the parts that support the specific affordance that are recorded during user annotation. Formally, we construct a kNN graph on sampled points where the adjacency matrix $\mathbf{A}$ writes as,

$$a_{ij} = \begin{cases} \|\mathbf{v}_i - \mathbf{v}_j\|_2 \,, \mathbf{v}_j \in NN_k(\mathbf{v}_i) \\ \quad 0, \quad otherwise \end{cases} \tag{1}$$

where $\mathbf{v}$ is the xyz spatial coordinate of point and NN_k denotes the set of k nearest neighbors. The adjacency matrix is symmetrized by $\mathbf{W} = 1/2(\mathbf{A} + \mathbf{A}^\top)$. Then we normalize the adjacency matrix by $\widetilde{\mathbf{W}} = \mathbf{D}^{-0.5}\mathbf{W}\mathbf{D}^{-0.5}$. where $\mathbf{D}$ is the degree matrix. Finally the scores $\mathbf{S}$ for all points is propagated by the closed-form solution $\mathbf{S} = (\mathbf{I} - \alpha\widetilde{\mathbf{W}^{-1}})\mathbf{Y}$. where $\mathbf{Y} \in \{0, 1\}^{N \times 18}$ is the one-hot label vector and 1 indicates positive label. α is a hyper-parameter controlling the decreasing speed of $\mathbf{S}$, we empirically set α to 0.998 throughout the experiments. Finally we linearly normalize $\mathbf{S}$ to the range between 0 and 1 so that it is a probabilistic score. Example shapes with propagated affordance groundtruth are shown in Fig. 3.

3.4. Statistics

The final dataset provides well-defined visual affordance score map annotations for 22949 shapes from 23 shape categories with at most 5 affordance types defined for each category. From the perspective of affordance categories, 3D AfffodanceNet contains 56307 affordance annotations from 18 affordance classes. It is worth noting that due to the multi-label nature, each point could be labeled with multiple affordances. More details of the dataset are presented in Tab. 2 and Tab. 3.

Object	Affordance	Num
Bag	grasp, lift, contain, open	125
Bed	lay, sit, support	181
Bowl	contain, wrap-grasp, pour	187
Clock	display	524
Dishwasher	open, contain	166
Display	display	887
Door	open, push, pull	220
Earphone	grasp, listen	223
Faucet	grasp, open	628
Hat	grasp, wear	222
Storage Furniture	contain, open	2186
Keyboard	press	156
Knife	grasp, cut, stab	314
Laptop	display, press	421
Microwave	open, contain, support	184
Mug	contain, pour, wrap-grasp, grasp	190
Refrigerator	contain, open	185
Chair	sit, support, move	6113
Scissors	grasp, cut, stab	68
Table	support, move	7990
Trash Can	contain, pour, open	315
Vase	contain, pour, wrap-grasp	1048
Bottle	contain, open, wrap-grasp, grasp, pour	411

Table 2. 3D AffordanceNet statistics. The first column shows the object category. The second column shows the defined affordance classes for each category.The third column shows the amount of each shape semantic category.

4. Tasks and Benchmarks

In this section, we benchmark three tasks to demonstrate the 3D AfffodanceNet dataset, namely, full-shape, partial-view and rotation-invariant affordance estimation. The 3D AffordanceNet dataset is split into train, validation and test

	Support	**Move**	**Sit**	**Contain**	**Open**	**Grasp**	**Pour**	**Display**	**Wrap-Grasp**
#Annot	14848	14540	6516	5155	4506	2253	2086	1914	1889
	Press	**Cut**	**Stab**	**Wear**	**Listen**	**Pull**	**Push**	**Lay**	**Lift**
#Annot	588	393	393	231	228	225	225	194	123

Table 3. The number of shapes that are positive for each category of affordance.

sets with a ratio of 70%, 10% and 20%, respectively according to the shape semantic category. The first experiment estimates point-wise affordance given full 3D point cloud as input. The second experiment estimates the affordance of partially visible objects observed from different viewpoints. The last experiment estimates the affordance of rotated 3D objects under two different rotation settings. We also create a semi-supervised affordance estimation benchmark to explore the opportunity of exploiting unlabeled data for affordance estimation.

4.1. Full-Shape Affordance Estimation

Given an object as 3D point cloud without knowing the affordances supported by the object, the full-shape affordance estimation task aims to estimate the supported affordance type and predict the point-wise probabilistic score of affordance. We show that state-of-the-art 3D point cloud segmentation networks predict reasonable results on 3D AffordanceNet.

Network and Training. We evaluate three network architecture, namely PointNet++[20], DGCNN[27] and U-Net[28] for this task. To obtain the point-wise score, we utilized the segmentation branch of PointNet++ and DGCNN as shared backbones to extract features for each point, then for each affordance type, we pass the features through multiple classification heads and used a sigmoid function to obtain the posterior scores. The classification heads were set up for each affordance category individually while the backbone networks were shared. We use cross-entropy loss L_{CE} for training the network as below,

$$l_{CE} = \frac{1}{N}\sum_{i}^{M}\sum_{j}^{N} -(1-t_{ij})log(1-p_{ij}) - t_{ij}log(p_{ij}) \tag{2}$$

where M is the total number of affordance types, N is the number of points within each shape, s_{ij} is the ground truth score of jth point of ith affordance category and p_{ij} is the predicted score.

Since the points with zero score account for a relatively large proportion of the total dataset, we further propose to use dice loss [15] to mitigate the imbalance issue. The dice loss l_{DICE} is defined as:

$$\begin{aligned} l_{DICE} = \sum_{i}^{M} 1 - \frac{\sum_{j}^{N} s_{i,j}p_{i,j} + \epsilon}{\sum_{j}^{N} s_{i,j} + p_{i,j} + \epsilon} \\ - \frac{\sum_{j}^{N}(1-s_{i,j})(1-p_{i,j}) + \epsilon}{\sum_{j}^{N} 2 - s_{i,j} - p_{i,j} + \epsilon} \end{aligned} \tag{3}$$

Finally the loss function is defined as $l = l_{CE} + l_{DICE}$. We train PointNet++ and DGCNN on the 3D AffordanceNet using the default training strategies and hyper-parameters described in respective papers[20, 27]. For Unet, we fine-tune the network initialized by the pre-trained weight provided by PointContrast[28].

Evaluation and Results. We evaluate four metrics for affordance estimation, including mean Average Precision (mAP) scores, mean squared error (MSE), Area Under ROC Curve (AUC) and average Intersection Over Union (aIoU). For AP, we calculate the Precision-Recall Curve and AP is calculated for each affordance. For AUC, we report the area under ROC Curve. For MSE, we calculate mean squared error of each affordance category and sum up the results from all affordance categories. For aIoU, we gradually tune up the threshold from 0 to 0.99 with 0.01 step to binarize the prediction, and the aIoU is the arithmetic average of all IoUs at each threshold. Except for the MSE, all the other metrics for each category are averaged over all shapes, a.k.a. macro-average. For each affordance category, the ground-truth map is binarized with 0.5 threshold before evaluation. The results are reported in Tab. 4 under the Full-Shape section and some qualitative examples from PointNet++ are selected and visualized in Fig. 4.

As shown in Tab. 4, the performances of three networks are close and all achieve a relatively low aIOU score, which indicates that affordance estimation is still a challenging task. Comparing the second row of Fig. 4 to corresponding ground truth, we found that PointNet++ produces some reasonable results. For example, the estimations of *grasp* on a *bag* are successfully localized on both the handles and the webbing. However, the results of *pour* on a *bottle* fail since the network predicts the scores mainly on the lid of the bottle rather than the body edge of the bottle where is the place that the water flow out. More qualitative examples are given in the supplementary.

Room for Performance Improvement The performances of PointNet++ and DGCNN on the tasks mentioned above are relatively weak. Hence, we evaluate the trained network on full-shape affordance estimation task over the training, validation and testing sets to investigate the room for performance improvement with results reported in Tab. 5. From the results we observe that both two networks still underfit, meaning that the proposed affordance estimation task is very challenging for existing point cloud analysis networks.

4.2. Partial-View Affordance Estimation

Although complete point clouds or meshes can provide detailed geometric information for affordance estimation, in real-world application scenarios, we can only expect partial view of 3D shapes, represented as partial point cloud. Therefore, another important task we are concerned with is to estimate the affordance from partial point cloud.

Network and Training. To obtain partial point clouds, we follow [11] to synthesize point cloud observed from certain

Full-shape	Avg	Grasp	Lift	Contain	Open	Lay	Sit	Support	Wrap.	Pour	Display	Push	Pull	Listen	Wear	Press	Move	Cut	Stab
P_mAP	48.0	43.4	75.1	56.9	46.6	63.0	81.1	52.5	19.0	46.9	59.3	20.5	37.9	41.6	20.4	31.4	35.3	41.1	90.9
P_AUC	87.4	82.8	97.1	89.3	90.6	92.6	96.0	89.7	72.6	89.2	90.6	83.1	85.3	85.9	67.9	90.9	79.0	91.4	98.8
P_aIOU	19.3	15.7	41.2	22.2	20.2	30.0	38.1	17.5	4.0	18.2	25.6	6.5	12.7	14.0	6.5	11.2	8.5	15.2	40.9
P_MSE	0.059	0.003	0.0001	0.006	0.003	0.0005	0.005	0.012	0.002	0.002	0.002	0.0007	0.0002	0.0006	0.0005	0.0006	0.021	0.0003	0.0001
D_mAP	46.4	43.9	85.2	57.6	51.8	12.3	80.9	54.0	20.7	47.7	65.5	20.5	40.5	36.0	18.3	34.2	35.5	40.2	91.4
D_AUC	85.5	82.5	98.7	89.9	91.6	50.1	96.1	90.2	74.6	89.2	92.1	85.0	89.7	86.1	61.9	91.8	78.9	91.7	98.7
D_aIOU	17.8	13.9	40.2	21.6	25.4	1.0	34.9	18.8	5.6	17.7	32.1	5.5	11.8	11.9	5.9	14.8	9.9	14.5	35.4
D_MSE	0.08	0.003	0.0001	0.007	0.003	0.0006	0.006	0.013	0.007	0.005	0.002	0.002	0.0006	0.002	0.002	0.0007	0.025	0.0002	0.0001
U_mAP	47.4	42.6	75.7	56.6	45.9	60.6	80.8	53.7	19.1	45.0	61.5	19.7	36.5	37.4	20.4	33.9	35.2	39.8	89.0
U_AUC	86.3	79.8	94.3	88.8	88.5	88.2	95.8	89.7	72.9	87.1	90.2	81.4	84.0	82.9	70.3	91.6	79.2	90.8	98.5
U_aIOU	19.7	13.7	41.2	22.4	20.8	29.4	37.3	18.6	4.7	18.3	32.4	6.2	13.0	13.1	4.4	14.5	9.2	14.2	41.5
U_MSE	0.063	0.003	0.0003	0.006	0.003	0.0006	0.005	0.014	0.002	0.002	0.002	0.0006	0.0002	0.0007	0.0003	0.0008	0.021	0.0002	0.0001
Partial	**Avg**	**Grasp**	**Lift**	**Contain**	**Open**	**Lay**	**Sit**	**Support**	**Wrap.**	**Pour**	**Display**	**Push**	**Pull**	**Listen**	**Wear**	**Press**	**Move**	**Cut**	**Stab**
P_mAP	45.7	43.2	80.6	41.9	48.5	52.6	69.8	45.5	20.0	47.0	52.5	24.1	36.5	42.1	15.3	30.7	37.8	43.3	92.6
P_AUC	85.2	81.2	96.2	83.3	87.9	86.7	95.0	86.5	71.3	88.4	85.2	84.9	86.1	84.2	64.1	84.7	79.6	90.2	98.8
P_aIOU	16.9	14.4	45.6	13.2	21.6	25.2	31.0	11.2	3.6	17.8	19.3	5.7	11.5	13.4	2.4	12.4	6.2	13.5	37.8
P_MSE	0.062	0.003	0.0001	0.005	0.003	0.0006	0.004	0.013	0.002	0.002	0.002	0.0002	0.0001	0.0007	0.0004	0.0006	0.025	0.0003	0.0001
D_mAP	42.2	40.1	83.8	38.5	44.5	46.6	67.3	43.9	19.6	44.8	50.3	15.7	28.6	19.9	17.7	26.9	35.6	43.3	92.1
D_AUC	83.7	80.3	97.0	80.8	86.8	82.6	94.6	87.0	70	86.8	81.7	83.6	86.0	75	64.1	83.0	78.5	90.0	99.1
D_aIOU	13.8	13.3	37.9	8.6	16.0	15.5	27.2	13.5	3.2	13.9	13.8	3.9	4.3	8.4	5.2	5.7	8.2	13.3	36.8
D_MSE	0.069	0.005	0.0002	0.005	0.004	0.0006	0.004	0.013	0.003	0.002	0.002	0.001	0.001	0.002	0.003	0.0004	0.022	0.0004	0.0001
U_mAP	43.0	40.8	73.2	41.2	47.6	44.6	68.9	45.3	18.5	45.2	53.2	19.5	29.6	38.9	11.1	29.6	36.7	39.1	90.8
U_AUC	83.2	78.9	94.6	81.7	87.1	80.4	94.5	87.0	69.3	87.5	84.4	80.9	79.2	84.2	56.4	84.9	77.7	89.3	98.7
U_aIOU	16.8	14.7	38.1	15.0	21.1	20.8	33.5	14.4	4.3	18.9	20.4	5.3	8.2	10.9	1.0	14.7	9.0	15.9	35.8
U_MSE	0.065	0.003	0.0002	0.006	0.003	0.001	0.006	0.012	0.003	0.003	0.002	0.0003	0.0002	0.0003	0.0006	0.0008	0.022	0.0007	0.0002
Rotate z	**Avg**	**Grasp**	**Lift**	**Contain**	**Open**	**Lay**	**Sit**	**Support**	**Wrap.**	**Pour**	**Display**	**Push**	**Pull**	**Listen**	**Wear**	**Press**	**Move**	**Cut**	**Stab**
P_mAP	47.3	43.1	85.7	58.1	39.6	62.7	80.6	53.8	20.4	47.5	47.2	21.8	34.8	39.7	19.0	28.1	36.3	40.4	91.9
P_AUC	87.0	82.0	97.9	89.5	86.2	91.1	95.9	90.1	74.2	89.4	87.1	85.4	87.9	84.3	67.0	88.5	80.3	91.5	98.5
P_aIOU	18.7	15.5	45.4	22.4	17.6	26.0	38.0	18.3	4.6	19.5	17.4	7.0	10.5	13.9	6.9	9.2	8.8	14.8	40.6
P_MSE	0.06	0.003	0.0001	0.006	0.003	0.0006	0.005	0.012	0.002	0.002	0.003	0.001	0.0003	0.0007	0.0007	0.0005	0.02	0.0003	0.0001
D_mAP	44.8	42.2	82.9	58.2	45.2	17.3	78.8	52.4	20.3	46.7	58.5	21.6	45.2	28.5	16.8	29.6	35.0	36.4	90.8
D_AUC	84.9	80.8	98.2	89.9	87.9	54.9	95.8	89.9	74.0	89.5	90.6	84.7	89.4	81.0	64.4	89.2	79.7	90.6	98.4
D_aIOU	16.1	13.6	36.1	19.6	21.2	1.0	29.2	18.5	3.4	13.6	25.3	5.7	13.6	11.3	5.7	13.1	9.7	13.6	36.3
D_MSE	0.074	0.003	0.0001	0.007	0.003	0.0006	0.007	0.013	0.005	0.004	0.002	0.003	0.0008	0.002	0.002	0.0009	0.021	0.0004	0.0001
U_mAP	46.1	42.7	74.4	56.1	40.4	58.4	81.2	54.9	18.5	44.7	56.4	20.7	35.3	36.8	17.4	31.6	35.6	36.2	88.8
U_AUC	86.1	81.2	95.8	87.5	85.9	88.4	95.8	90.2	72.2	87.6	87.9	85.1	87.9	83.2	63.1	90.0	78.9	90.3	98.4
U_aIOU	18.9	15.5	39.6	21.8	18.4	24.7	38.4	18.9	4.5	18.6	26.9	6.6	12.2	14.5	5.1	14.1	9.9	12.1	37.7
U_MSE	0.06	0.003	0.0001	0.006	0.003	0.0004	0.005	0.013	0.002	0.002	0.002	0.0007	0.0002	0.0008	0.0004	0.0008	0.021	0.0003	0.0001
Rotate SO(3)	**Avg**	**Grasp**	**Lift**	**Contain**	**Open**	**Lay**	**Sit**	**Support**	**Wrap.**	**Pour**	**Display**	**Push**	**Pull**	**Listen**	**Wear**	**Press**	**Move**	**Cut**	**Stab**
P_mAP	41.8	40.5	78.5	42.4	32.7	38.4	74.5	48.3	19.4	41.7	41.1	19.7	30.3	39.4	17.6	21.5	34.6	41.1	90.6
P_AUC	83.3	79.0	93.6	81.1	81.3	79.6	93.9	87.4	71.6	85.4	83.7	83.1	84.0	84.2	64.8	78.2	78.6	91.0	99.4
P_aIOU	15.2	12.8	38.3	12.2	13.1	9.6	33.6	16.5	3.8	16.1	13.7	3.0	11.1	14.8	5.5	8.8	8.9	14.4	37.2
P_MSE	0.072	0.003	0.0001	0.008	0.003	0.0007	0.007	0.015	0.003	0.003	0.003	0.0002	0.0001	0.0006	0.002	0.0009	0.022	0.0006	0.0001
D_mAP	37.3	37.9	70.7	37.3	34.2	9.7	73.9	46.8	17.6	40.2	46.8	19.1	37.4	6.9	11.8	22.3	30.9	41.4	86.4
D_AUC	78.9	79.1	95.6	79.4	81.4	36.3	93.9	87.2	71.2	85.6	85.7	82.3	88.8	43.5	60.3	82.7	76.9	91.9	99.1
D_aIOU	12.8	13.6	32.5	7.8	13.9	1.0	35.4	14.4	4.9	16.4	19.3	4.5	11.0	0.003	1.0	8.4	7.1	15.7	23.3
D_MSE	0.08	0.004	0.0002	0.007	0.003	0.0006	0.007	0.015	0.005	0.004	0.005	0.002	0.0005	0.0004	0.0006	0.0009	0.022	0.0007	0.0002
U_mAP	37.9	37.0	61.2	38.0	29.8	34.0	77.4	49.9	16.4	39.3	42.6	14.8	24.7	35.7	8.6	20.1	31.8	36.9	83.3
U_AUC	80.9	76.9	90.5	79.5	77.7	78.1	94.1	87.8	67.9	82.7	81.9	78.1	83.5	83.7	52.5	76.6	77.1	89.6	99.0
U_aIOU	12.0	15.3	8.2	10.8	10.7	5.4	35.8	16.2	2.7	12.7	15.8	1.0	4.3	12.3	1.0	7.3	7.9	11.9	35.8
U_MSE	0.07	0.005	0.0001	0.008	0.003	0.0005	0.006	0.013	0.003	0.003	0.004	0.0002	0.0001	0.0004	0.0004	0.0008	0.022	0.0003	0.0001

Table 4. Affordance Estimation Results. Except for the MSE results, others are shown in percentage, the higher the scores the higher the results. Algorithm P, D and U represent PointNet++[20], DGCNN[27] and U-Net[28] respectively. The words *Full-Shape*, *Partial*, *Rotate z* and *Rotate SO(3)* represent the full-shape, partial, z/z and $SO(3)/SO(3)$ rotation-invariant affordance estimation, respectively.

	mAP	AUC	aIOU	MSE	Loss		mAP	AUC	aIOU	MSE	Loss
P Train	52.3	89.8	21.7	0.054	8.75	**D Train**	51.7	89.1	21.2	0.061	8.83
P Val	48.2	88.0	19.2	0.057	8.81	**D Val**	47.8	85.8	17.5	0.075	8.91
P Test	48.0	87.4	19.3	0.059	8.83	**D Test**	46.4	85.5	17.8	0.08	8.93

Table 5. The performances of two different networks on full-shape affordance estimation task over train, validate and test sets. P represents PointNet++ and D refers to DGCNN.

camera viewpoints. Only points directly facing the camera will be preserved as visible points and each point is assigned a radius to create occlusion effect. In specific, because all shapes are well aligned within the (-1,-1,-1) to (1,1,1) cube, we set up 4 affine cameras located at (1,1,1), (-1,-1,1), (1,-1,-1), (-1,1,-1) in Cartesian coordinate system, facing towards the origin. After obtaining the partial point clouds, we sample 2048 points from each viewpoint via furthest point sampling, if the number of points of the point cloud is fewer than 2048, we utilize the point cloud up-sampling method proposed in [30] to up-sample the data. We use exactly the same backbone networks and training strategies described in previous sections.

Evaluation and Results. During testing stage, we estimate the affordance on the visible partial point cloud only. The evaluation protocol follows the one described in Section 4.1. All evaluation metrics are reported in Tab. 4 with qualitative results from PointNet++ shown in Fig. 4.

Unsurprisingly, the quantitative performances of three networks have decreased due to the loss of geometric information of partial point cloud relative to complete point cloud. Nevertheless, we still observe reasonable qualitative results even though only a partial view is observed. For instance, the network produces high prediction for *move* on the upside of the legs of a *table* despite the unseen parts of the legs. The *grasp* for *bag*, *hear* for *earphone*, *sit* for *chair*, etc., are all more-or-less correctly predicted. In contrast, the estimation for *contain* on *storage furniture* are par-

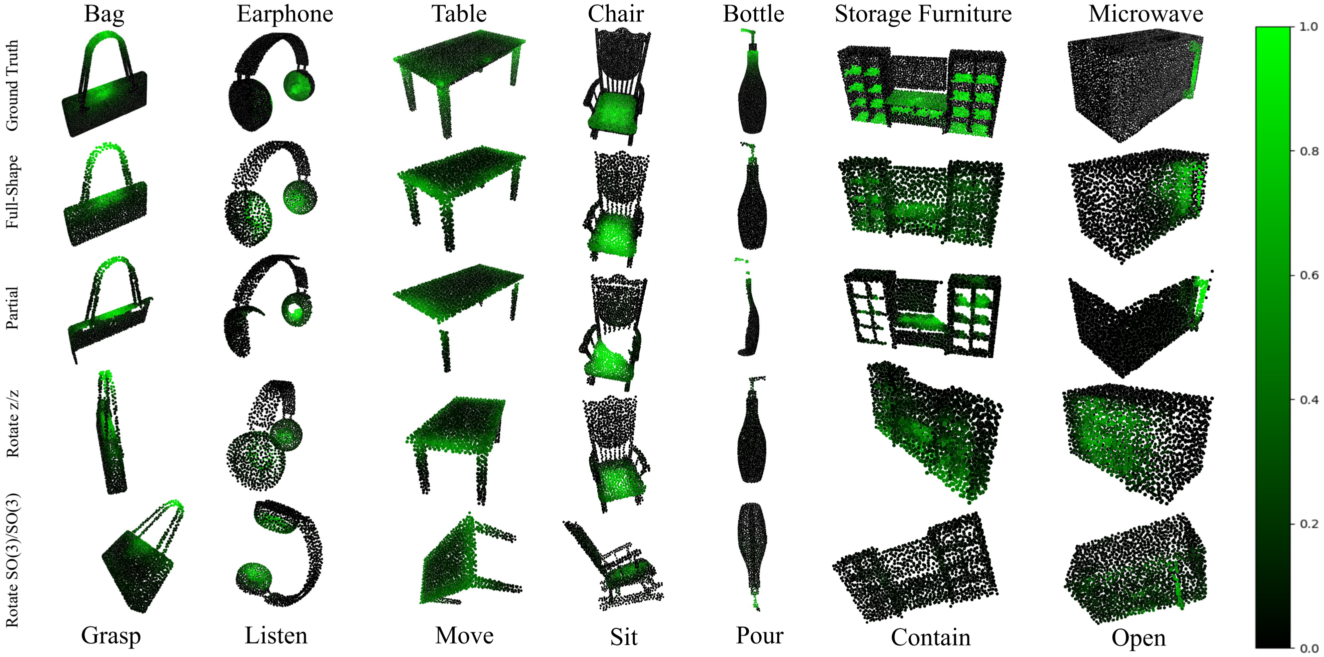

Figure 4. Qualitative results for affordance estimation. The top row shows the ground truth. The second row shows the full-shape estimated results, the third row shows the partial-view estimated result, the fourth and the bottom row show the z/z and $SO(3)/SO(3)$ rotation-invariant estimated results, respectively. All results come from PointNet++. The top words indicate the semantic category of each column and the bottom words indicate the affordance category. The greener the color of the points, the higher the confidence about specific affordance types. *Wrap.* is the abbreviation of *Wrap-Grasp.*

tially missing since it predicts the scores on the top of the furniture which is not fully observed.

4.3. Rotation-Invariant Affordance Estimation

The shapes in 3D AffordanceNet are all aligned in canonical poses, however, the data observed by sensors in real world are not always in canonical poses. The difference in rotation between real data and training data will lead to a performance drop in real-world usage which inspired research into rotation equivariant network [5]. Hence, it is critical to train the algorithms to estimate affordance on rotated objects. In this section, we provide a benchmark for affordance estimation subject to two types of rotations.

Network and Training. We used the same backbone networks, training strategies and hyper-parameters described in the Sect. 4.1. We propose two different rotation settings for experiment: z/z and $SO(3)/SO(3)$ where z/z means rotation is applied along z axis only for both training and inference stages while $SO(3)/SO(3)$ refers to $SO(3)$ rotation, i.e. freely rotation along x, y and z axes. During training session, we randomly sample rotation poses between $[0,2\pi]$ for each shape in the training mini-batch on-the-fly. We train proposed methods on complete point cloud. For testing phase, we randomly sample 5 rotation poses for each shape for both rotation settings and fix the sampled rotations for testing data.

Evaluation and Results. We calculate mAP, AUC, aIOU on the proposed methods. Quantitative results are presented in Tab. 4 and qualitative results are shown in the fourth and fifth rows of Fig. 4 for z/z and $SO(3)/SO(3)$ settings with PointNet++ as backbone. We observe that the performances on both z/z and $SO(3)/SO(3)$ settings dropped compared to canonical view experiments. In particular, for z/z setting, the performance dropped around 1% in all metrics for all networks as backbone. While more significant loss of performance is observed for $SO(3)/SO(3)$ setting with $5-10\%$ drop in all metrics. This is aligned with our expectation that $SO(3)/SO(3)$ is a much more challenging task. We further make observations from the qualitative results in Fig. 4. First, under z/z rotation scheme, despite the consistent performance drop, affordance estimations are largely correct across most categories. Obvious mistakes are made in *bottle* and *microwave* where the former missed the tip which supports *pour* while the latter mistakenly predict the while door of *microwave* for *open*. Under the more challenging $SO(3)/SO(3)$ scheme, there is still a visually satisfying results for most shapes. The most prominent error is made on *storage furniture* where *contain* is totally missed, probably because the complex geometric structure (many concave shapes) renders *contain* a hard affordance to learn under arbitrary rotation. In general, we believe affordance estimation under $SO(3)/SO(3)$ a very challenging task and it deserves further investigation.

VAT	**Avg**	**Grasp**	**Lift**	**Contain**	**Open**	**Lay**	**Sit**	**Support**	**Wrap.**	**Pour**	**Display**	**Push**	**Pull**	**Listen**	**Wear**	**Press**	**Move**	**Cut**	**Stab**
mAP	36.6	34.8	78.8	44.3	22.6	34.9	64.8	41.4	13.3	41.5	58.9	13.5	8.3	20.5	8.5	29.5	19.4	33.1	90.2
AUC	78.8	75.8	95.3	78.4	79.2	71.2	92.4	86.7	56.5	83.7	89.9	69.0	74.8	74.1	51.4	88.8	62.7	89.3	99.1
aIOU	11.2	12.7	20.3	14.2	4.8	6.4	16.9	7.7	5.1	17.6	30.2	2.5	1.1	8.4	3.6	12.8	2.9	9.5	24.3
MSE	0.155	0.007	0.0001	0.02	0.006	0.0007	0.012	0.023	0.023	0.007	0.008	0.0003	0.0001	0.002	0.002	0.005	0.034	0.004	0.0003
Full-Shape	**Avg**	**Grasp**	**Lift**	**Contain**	**Open**	**Lay**	**Sit**	**Support**	**Wrap.**	**Pour**	**Display**	**Push**	**Pull**	**Listen**	**Wear**	**Press**	**Move**	**Cut**	**Stab**
mAP	34.3	34.5	77.9	42.3	18.2	36.5	62.9	38.8	11.2	38.9	53.3	13.4	7.3	9	6.7	24.5	17.4	35.8	89.5
AUC	77.5	75.4	94.8	78.1	75.7	73.3	92.1	85.8	54.4	82.1	87.5	74.5	77.4	61.6	46.7	83.4	64.1	89.8	98.9
aIOU	9.8	11.2	28.1	14.3	2.5	10.0	23.4	9.8	2.2	7.5	19.9	1.9	1.0	1.6	1.6	6.8	2.3	5.6	26.5
MSE	0.105	0.009	0.0002	0.013	0.003	0.001	0.013	0.021	0.004	0.003	0.003	0.0002	0.0001	0.0004	0.001	0.0008	0.031	0.0007	0.0001

Table 6. The Results of Semi-Supervised Affordance Estimation. All numbers are in % except for MSE. We only implement semi-supervised affordance estimation on DGCNN. The words *Full-Shape* and *VAT* represent full-shape estimation and semi-supervised affordance estimation with virtual adversarial training. *Wrap.* is the abbreviation of *Wrap-Grasp*.

4.4. Semi-Supervised Affordance Estimation

Although the label propagation procedure allows the annotators to only annotate a few keypoints on the object surface, affordance annotation still remains as an expensive and labor intensive procedure. Inspired by the recent success in semi-supervised learning (SSL) [14, 25, 16] we establish a benchmark for semi-supervised affordance estimation. We synthesize a semi-supervised setting by randomly sampling 1% training data, assumed to be labeled, and the rest are assumed to be unlabeled data. The validation and testing sets are kept the same with standard benchmarks.

Network and Training. We utilize DGCNN[27] as our backbone. During every mini-batch in the training stage, we randomly sample a equal number of labeled data $\mathbf{X}_l$ and unlabeled data $\mathbf{X}_{ul}$. To fully exploit the unlabeled data, we employ a state-of-the-art semi-supervised learning framework, namely Virtual Adversarial Training (VAT) [16]. It encourages the consistency between the posterior of unlabeled sample and its augmentation, measured by mean square error,

$$l_{mse} = \frac{1}{N} \sum_{i}^{M} \sum_{j}^{N} ||p_{i,j} - \hat{p}_{i,j}||_2^2 \quad (4)$$

where $\hat{p}_{i,j}$ is the posterior prediction for augmented sample. To best exploit the consistency power, the augmentation is obtained by first applying a one step adversarial attack, the corresponding adversarial perturbation is then added to the original point cloud to produce the augmentation. Finally, the total loss for semi-supervised affordance estimation combines both losses defined for labeled data and unlabeled data.

$$l = l_{CE} + l_{DICE} + l^{l}_{mse} + l^{u}_{mse} \quad (5)$$

where l^{l}_{mse} and l^{u}_{mse} is the mean square error (MSE) calculated between labeled and unlabeled data, respectively. We compare the semi-supervised approach against a fully supervised baseline which is trained on the 1% labeled data alone with cross-entropy and dice loss. We use a mini-batch of 16, 8 for labeled data and 8 for unlabeled data, and follow the same training strategies and hyper-parameters described in [16]. We train a full-shape affordance estimation method based on DGCNN only on the labelled data following the description in Sect. 4.1.

Evaluation and Results We evaluated the methods following the metrics described in Sect. 4.1 with results reported in Tab. 6. Comparing the performance of semi-supervised affordance estimation to the full-shape one, we found that semi-supervised affordance estimation outperforms the fully supervised baseline on all three metrics. Specifically, the gains for some affordance categories (*e.g.* open) that have low metrics on full-shape affordance estimation are high, which indicates that unlabeled data can provide useful information for affordance learning. In conclusion, we believe that exploiting unlabeled data to improve the performance has practical value and should receive more attention in the future.

5. Conclusion

In this work, we proposed 3D AffordanceNet, a 3D point cloud benchmark consisting of 22949 shapes from 23 semantic object categories, annotated with 56307 affordance annotations and covering 18 visual affordance categories. Based on this dataset, we define three individual affordance estimation tasks and benchmarked three state-of-the-art point cloud deep learning networks. The results suggested future research is required to achieve better performance on difficult affordance categories and under SO(3) rotation. Furthermore, we proposed a semi-supervised affordance estimation method to take advantage of large amount of unlabeled data. The proposed dataset encourage the community to focus on affordance estimation research.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China (Grant No.: 61771201, 61902131), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No.: 2017ZT07X183), and the Guangdong R&D key project of China (Grant No.: 2019B010155001). Xun Xu acknowledges the A*STAR Career Development Award (CDA) Funding for providing financial support (Grant No. 202D8243).

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet:

An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[2] Ching-Yao Chuang, Jiaman Li, Antonio Torralba, and Sanja Fidler. Learning to act properly: Predicting and explaining affordances from images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[3] Thanh-Toan Do, Anh Nguyen, and Ian Reid. Affordancenet: An end-to-end deep learning approach for object affordance detection. In *IEEE International Conference on Robotics and Automation*, 2018.

[4] Jay Earley. An efficient context-free parsing algorithm. *Communications of the ACM*, 1970.

[5] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision*, 2018.

[6] J Gibson James. The ecological approach to visual perception, 1979.

[7] Helmut Grabner, Juergen Gall, and Luc Van Gool. What makes a chair a chair? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[8] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual affordance and function understanding: A survey. *arXiv preprint arXiv:1807.06775*, 2018.

[9] Ruizhen Hu, Oliver van Kaick, Bojian Wu, Hui Huang, Ariel Shamir, and Hao Zhang. Learning how objects function via co-analysis of interactions. *ACM Transactions on Graphics*, 2016.

[10] Ashesh Jain, Amir R Zamir, Silvio Savarese, and Ashutosh Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[11] Sagi Katz, Ayellet Tal, and Ronen Basri. Direct visibility of point sets. In *ACM Special Interest Group on GRAPHics and Interactive Techniques 2007 papers*. 2007.

[12] Hema Swetha Koppula, Rudhir Gupta, and Ashutosh Saxena. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research*, 2013.

[13] Hema S Koppula and Ashutosh Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[14] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations*, 2017.

[15] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision*, 2016.

[16] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[17] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[18] Austin Myers, Ching L Teo, Cornelia Fermüller, and Yiannis Aloimonos. Affordance detection of tool parts from geometric features. In *IEEE International Conference on Robotics and Automation*, 2015.

[19] Anh Nguyen, Dimitrios Kanoulas, Darwin G Caldwell, and Nikos G Tsagarakis. Object-based affordances detection with convolutional neural networks and dense conditional random fields. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017.

[20] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, 2017.

[21] Siyuan Qi, Siyuan Huang, Ping Wei, and Song-Chun Zhu. Predicting human activities using stochastic grammar. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[22] Anirban Roy and Sinisa Todorovic. A multi-scale cnn for affordance segmentation in rgb images. In *Proceedings of the European Conference on Computer Vision*, 2016.

[23] Johann Sawatzky, Abhilash Srikantha, and Juergen Gall. Weakly supervised affordance detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[24] Hyun Oh Song, Mario Fritz, Daniel Goehring, and Trevor Darrell. Learning to detect visual grasp affordance. *IEEE Transactions on Automation Science and Engineering*, 2015.

[25] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, 2017.

[26] Tuan-Hung Vu, Catherine Olsson, Ivan Laptev, Aude Oliva, and Josef Sivic. Predicting actions from static scenes. In *Proceedings of the European Conference on Computer Vision*, 2014.

[27] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 2019.

[28] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Proceedings of the European Conference on Computer Vision*, 2020.

[29] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics*, 2016.

[30] Wang Yifan, Shihao Wu, Hui Huang, Daniel Cohen-Or, and Olga Sorkine-Hornung. Patch-based progressive 3d point set upsampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[31] Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for

fine-grained and hierarchical shape segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[32] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 2019.

[33] Yixin Zhu, Yibiao Zhao, and Song Chun Zhu. Understanding tools: Task-oriented object modeling, learning and recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.