Grounding 3D Object Affordance from 2D Interactions in Images Supplementary Materials

Yuhang Yang[1], Wei Zhai[1,*], Hongchen Luo[1], Yang Cao[1,2], Jiebo Luo[3], Zheng-Jun Zha[1]
[1] University of Science and Technology of China [3] University of Rochester
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{yyuhang@mail., wzhai056@mail., lhc12@mail., forrest@}ustc.edu.cn
jluo@cs.rochester.edu zhazj@ustc.edu.cn

A. Implementation Details

A.1. Method Details

For the image branch, we chose ResNet18 as the extractor, and the input images are randomly cropped and resized to 224×224. To prevent the interactive subject and object in the image from being cropped out, we set the crop area outside the bounding box of the interactive subject and object, shown in Fig. 1. The image extractor output the image feature $\mathbf{F_I} \in \mathbb{R}^{512\times7\times7}$. Then, utilizing the bounding boxes B_{obj}, B_{sub} to calculate the scene mask M_{sce} (outside these two boxes), taking them to locate the object, subject, and scene feature in $\mathbf{F_I}$, next, apply Roi-Align to obtain the same size object, subject, scene feature $\mathbf{F}_i, \mathbf{F}_s, \mathbf{F}_e \in \mathbb{R}^{512\times4\times4}$, reshape them to $\mathbb{R}^{512\times16}$. For the point cloud branch, the number of points for each input point cloud is fixed to 2048, we take 3 set abstraction (SA) layers with multi-scale grouping to extract the point-wise feature. Each SA layer uses Farthest Point Strategy (FPS) to sample points, and the number of sampling points for each layer is set to 512, 128, and 64 respectively. Finally, this branch outputs the point-wise feature $\mathbf{F}_p \in \mathbb{R}^{512\times64}$. We show the dimension of tensors in the whole pipeline in Tab. 1. In the implementation, the joint denotes combining the image and point cloud feature sequence at the last dimension, like joint $\bar{\mathbf{P}}$ and $\bar{\mathbf{I}}$ as the object representation $\mathbf{F}_j$.

Furthermore, here we make a more detailed explanation for the KLD loss $\mathcal{L}_{KL}$. $\mathbf{F}_j$ denotes the joint object representation, and $\mathbf{F}_\alpha$ denotes the joint affordance representation. Since the order of the sequence does not change in the calculation process, we split them back into image and point cloud sequences. $\mathbf{F}_{i\alpha}$ contains the affordance feature distribution of each region in $\hat{\mathbf{F}}_i$, the regions with high correspondence to affordance possess more significant features. This relative difference is reflected in the region feature distribution, the insight is to make the distribution of $\hat{\mathbf{F}}_i$ also keep the distribution characteristics of the $\mathbf{F}_\alpha$, so as to implicitly enhance the affordance region features in the object representation, shown in Fig. 2. And with the establishment of the correspondence between the image and point cloud regions in the alignment process, this property

*Corresponding Author.

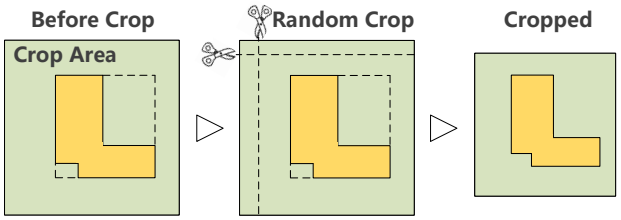

Figure 1. **Random Crop.** Cropping in image augmentation, we only do it outside the object and subject bounding boxes.

Table 1. **Tensors.** The dimension and meaning of the tensors in the pipeline.

Tensor	Dimension	Meaning
$\mathbf{F_I}$	$512 \times 7 \times 7$	image extractor output
$\mathbf{F}_p$	512×64	point cloud extractor output
$\mathbf{F}_{i,s,e}$	512×16	features output by roi-align
$\mathbf{P}$	512×64	project $\mathbf{F}_p$ to a feature space
$\mathbf{I}$	512×16	project $\mathbf{F}_i$ to a feature space
φ	64×16	dense similarity between $\mathbf{P}$ and $\mathbf{I}$
$\bar{\mathbf{P}}$	512×64	point feature with structural relevance
$\bar{\mathbf{I}}$	512×16	image feature with structural relevance
$\mathbf{F}_j$	512×80	joint object representation
$\mathbf{Q}$	512×80	the query projected by $\mathbf{F}_j$
$\mathbf{K}_1, \mathbf{K}_2$	512×16	keys projected by $\mathbf{F}_s, \mathbf{F}_e$
$\mathbf{V}_1, \mathbf{V}_2$	512×16	values projected by $\mathbf{F}_s, \mathbf{F}_e$
$\mathbf{\Theta}_1, \mathbf{\Theta}_2$	512×80	interaction contexts
$\mathbf{F}_\alpha$	512×80	joint affordance representation
$\hat{\mathbf{F}}_p, \mathbf{F}_{p\alpha}$	512×64	split from $\mathbf{F}_j, \mathbf{F}_\alpha$
$\hat{\mathbf{F}}_i, \mathbf{F}_{i\alpha}$	512×16	split from $\mathbf{F}_j, \mathbf{F}_\alpha$
$\hat{\phi}$	2048×1	3D object affordance

tends to be shown in $\hat{\mathbf{F}}_p$. This makes the region alignment and affordance extraction exhibit a mutual mechanism, the affordance representation could better correspond the object affordance regions, and the object region features with stronger correspondences could assist to extract more explicit affordance features in the optimization process. The computation of $\mathcal{L}_{KL}$ is expressed as:

$$\mathcal{L}_{KL} = KLD(\hat{\mathbf{F}}_i, \mathbf{F}_{i\alpha}) = \sum_n \mathbf{F}_{i\alpha_n} log(\epsilon + \frac{\mathbf{F}_{i\alpha_n}}{\epsilon + \hat{\mathbf{F}}_{i_n}}), \quad (1)$$

where ϵ is a regularization constant, n denotes the regions. Since $\mathbf{F}_{i\alpha}$ is split from $\mathbf{F}_\alpha$ and $\hat{\mathbf{F}}_i$ is split from $\mathbf{F}_j$, $\mathcal{L}_{KL}$ could optimize the layers for alignment in JRA, also the layers for affordance extraction in ARM, expressed as:

$$\theta_1^{t+1} \leftarrow \theta_1^t - \eta \nabla \frac{\partial \mathcal{L}_{KL}(\theta_1^{t+1})}{\partial \theta_1^{t+1}}, \quad (2)$$

where t is the number of steps, η is the learning rate, θ_1 denotes the parameters of layers that tend to be optimized.

In JRA, for the learnable layers, we give the following explanations. f_δ contains two 1×1 convolution layers, which is used to project $\mathbf{F}_i$ and $\mathbf{F}_p$ into a feature space. f_i, f_p are utilized to map the region relevance of object feature from images and point clouds. After the feature extraction, the $\mathbf{F}_p$ and $\mathbf{F}_i$ represent sub-regions of the raw objects, our aim is to map the spatial correlation among these sub-regions. There are several ways to satisfy this mapping: 1) Transformer-based. In this method, each sub-region feature can be regarded as a patch, then take the self-attention technique to model the correlation among the patches. 2) Expectation-Maximization Attention [10]. This method initializes a group of bases μ, and then performs alternating "E-step" and "M-step" to mine the correspondence between object features from the image and point cloud. 3) Multi-layer perception. This is the most direct way, which regards all regions as a whole one and directly maps the intra-relation among the region features. We conduct a comparative experiment to see their performance and finally chose the self-attention technique (see Tab. 5).

In ARM, for the extraction of affordance representation $\mathbf{F}_\alpha$, we take the feature $\mathbf{F}_j$ as the object representation to model the interaction contexts and reveal affordance. This process could be regarded as fusing the image and point cloud feature as the multi-modal representation and completing certain downstream tasks like affordance extraction. And the feature alignment is performed implicitly during optimizing the extraction, which is a learning-based way [3]. Meanwhile, theoretically, the multi-modal feature contains more information, for example, structures, colors, and textures, which is also beneficial to the downstream task like the extraction of affordance (see clarification in [9]).

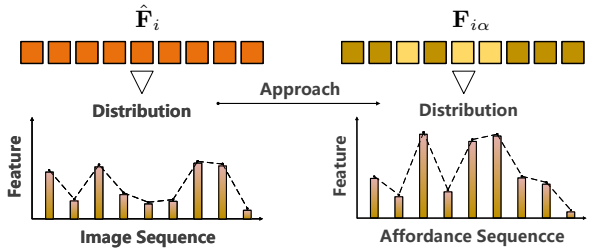

Figure 2. **The Distribution.** The right denotes feature distribution of each region in $\mathbf{F}_{i\alpha}$, regions with higher correspondence to affordance keep more significant feature. And the goal is to let this property exists in $\hat{\mathbf{F}}_i$, so we narrow the distribution discrepancy between $\hat{\mathbf{F}}_i$ and $\mathbf{F}_{i\alpha}$ to tune the feature distribution of $\hat{\mathbf{F}}_i$.

A.2. Evaluation Metrics

We use four evaluation metrics to benchmark the PIAD. **AUC** [11] is used to evaluate the predicted saliency map on the point cloud. The **a**verage **I**ntersection **O**ver **U**nion (**aIoU**) [18] is aimed at evaluating the overlap between the affordance region predicted in the point cloud and the labeled region. **SIM**ilarity (**SIM**) [21] is used to measure the similarity between the prediction map and the ground truth. **M**ean **A**bsolute **E**rror (**MAE**) [23] is the absolute difference between the prediction map and ground truth for point-wise measurement.

- **AUC** [11]: The Area under the ROC curve, referred to as AUC, is the most widely used metric for evaluating saliency maps. The saliency map is treated as a binary classifier of fixations at various threshold values (level sets), and a ROC curve is swept out by measuring the true and false positive rates under each binary classifier (level set).

- **aIOU** [18]: IoU is the most commonly used metric for comparing the similarity between two arbitrary shapes. The IoU measure gives the similarity between the predicted region and the ground-truth region, and is defined as the size of the intersection divided by the union of the two regions. It can be formulated as:

$$IoU = \frac{TP}{TP + FP + FN}, \quad (3)$$

  where TP, FP, and FN denote the true positive, false positive, and false negative counts, respectively.

- **SIM** [21]: The similarity metric (SIM) measures the similarity between the prediction map and the ground truth map. Given a prediction map P and a continuous ground truth map Q^D, $SIM(\cdot)$ is computed as the sum of the minimum values at each element, after

normalizing the input maps:

$$SIM(P, Q^D) = \sum_i min(P_i, Q_i^D),$$
$$where \quad \sum_i P_i = \sum_i Q_i^D = 1. \tag{4}$$

– **MAE** [23]: The Mean Absolute Error (MAE) is a useful measure widely used in model evaluations. The calculation of MAE is relatively simple. It involves summing the magnitudes (absolute values) of the errors to obtain the "total error" and then dividing the total error by n:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |e_i|, \tag{5}$$

where e_i is the calculated model error.

A.3. Training Details

Our model is implemented in PyTorch and trained with the Adam optimizer. The training epoch is set to 80. All training processes are on a single NVIDIA 3090 Ti GPU with an initial learning rate of 0.0001. The loss balance hyper-parameters $\lambda_1, \lambda_2, \lambda_3$ are set to 1, 0.3, and 0.5 respectively, and the training batch size is set to 16. The image extractor uses the pre-trained parameters on ImageNet, while the point cloud extractor is trained from scratch. In addition, since images and point clouds do not need strict one-to-one pairing, we pair images and point clouds online during training. An image could be paired with n point clouds in one training step, which is equivalent to expanding training samples. In the training process, the loss of an image and all paired point clouds is accumulated, and the gradient is calculated according to the accumulated loss. We set $n = 2$ in our implementation.

B. Dataset

B.1. Data Samples

According to our collection principles, the object in the image and point cloud belongs to the same category and the image demonstrates the way in which the 3D object could interact. For the interactive subject in the image, it could be a human or just a human body part, this is in line with the way humans learn from demonstrations, which usually only needs to observe the specific parts that occur interactions. Here, we give more data pairs in **P**oint-Image **A**ffordance **D**ataset (**PIAD**), shown in Fig. 3.

B.2. Dataset Partitions

Here, we explain how the dataset is divided and the reason for doing so. PIAD includes 23 object categories and 17

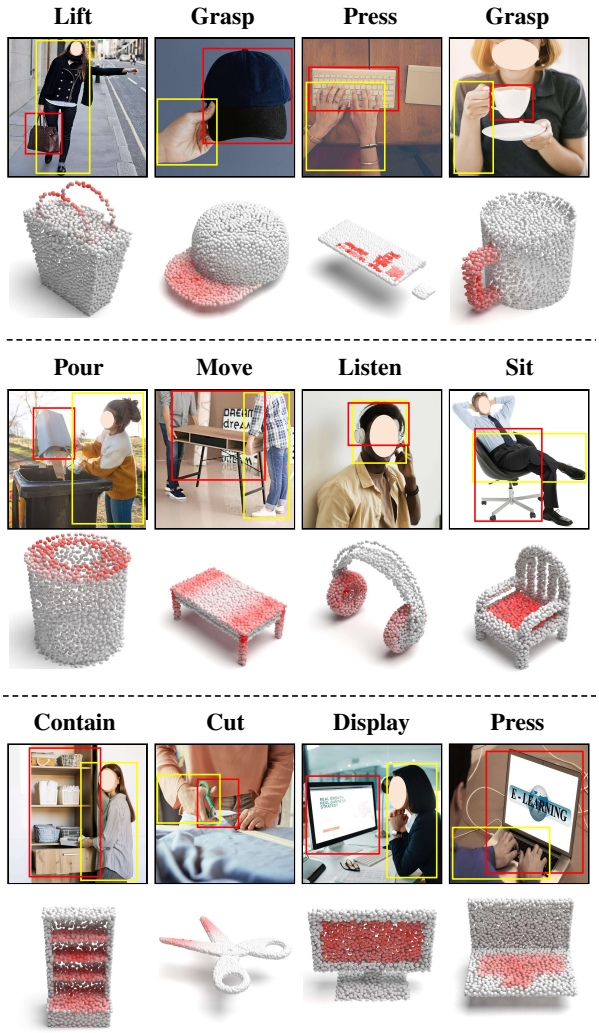

Figure 3. **Examples of PIAD.** Some paired images and point clouds in PIAD. The "yellow" box in the image is the bounding box of the interactive subject, and the "red" box is the bounding box of the interactive object.

affordance classes, we divide it into two partitions: **`Seen`** and **`Unseen`**. **`Seen`** includes all objects and affordances, this partition is utilized to verify whether it is feasible to ground 3D object affordance in such a learning paradigm. The category of affordance corresponding to each object and the number of images and point clouds in the training set and testing set are shown in Tab. 2. A series of experiments in the main paper has proved this kind of learning paradigm is achievable. However, embodied agents commonly face the human space, and meet novel objects, to facilitate the agents' capabilities for anticipating novel objects' affordance, we make the **`Unseen`** partition. In **`Unseen`**, serval objects do not exist in the training set. The principle of the partition is that the affordance of unseen objects should have corresponding objects in the training set. Eventually, we chose the following objects to the testing set

Table 2. **Data Distribution.** The affordance corresponding to each object and the number of its corresponding images and point clouds.

Objects	Affordance	Image		Point	
		Train	Test	Train	Test
Vase	wrapgrasp,contain	186	46	209	46
Display	display	210	52	253	52
Bed	lay,sit	117	28	127	28
Microwave	contain,open	111	27	130	27
Door	push,open	105	25	108	25
Earphone	listen,grasp	144	35	157	35
Bottle	wrapgrasp,contain,open,pour	252	61	288	61
Bowl	wrapgrasp,contain,pour	117	28	132	28
Laptop	display,press	238	58	295	58
Clock	display	38	9	205	9
Scissors	cut,stab,grasp	43	10	49	10
Mug	wrapgrasp,contain,grasp,pour	142	35	152	35
Faucet	open,grasp	196	48	210	48
StorageFurniture	contain,open	217	53	329	53
Bag	contain,open,grasp,lift	74	15	88	15
Chair	sit,move	797	199	1352	199
Dishwasher	contain,open	84	20	117	20
Refrigerator	contain,open	119	28	130	28
Table	move,support	401	100	957	100
Hat	wear,grasp	143	35	156	35
Keyboard	press	100	25	110	25
Knife	cut,stab,grasp	196	47	225	47
TrashCan	contain,open,pour	120	28	221	28

as the unseen category: "Dishwasher", "Microwave", "Scissors", "Laptop", "Bed", "Vase". It can be seen from Tab. 2 that the number of images and point clouds is not consistent. They do not need a fixed one-to-one pair, an image can be paired with multiple point clouds.

C. Experiments

C.1. Details of Modular Baselines

Since there is no previous work research grounding 3D object affordance in such a cross-modal fashion, we select some advanced studies in the image-point cloud cross-modal learning area to make the comparison experiment. The common aspect of these works is that they extract the feature of images and point clouds separately, and then align or fuse the extracted features. These methods are divided into two types: one is to use the camera intrinsic parameter in the alignment or fusion module for obtaining spatial correspondence, and the other is to align or fusion multi-modal features directly in the feature space without using the intrinsics. For methods using the camera intrinsic parameter, we remove the step that uses intrinsic parameters to verify the effectiveness of such methods on the proposed task setting, in which images and point clouds originate from different physical instances and it is infeasible to obtain spatial correspondence through the assistance of camera priors. All comparative methods share the same extractors with our method, and the difference occurs subsequent to the extraction of $\mathbf{F}_p$ and $\mathbf{F}_i$.

- **`Baseline`**: For the design of baseline, we directly concatenate the features that are output from the image and point cloud extractors, let the concatenation be a fusion block and there are no intermediate steps to correspond the region features from different sources.
- **`MBDF-Net (MBDF)`** [22]: This work focus on 3D object detection work. It has three branches: image branch, lidar branch, and fusion branch. An Adaptive Attention Fusion Module (AAF) is proposed in this work to fuse the features from the image and point cloud. We take its AAF as the cross-modal block to fuse the image and point cloud feature. And in AAF, we remove the step that utilizes the inner parameters of the camera to project the coordinates $p(x, y, z)$ of each point in the point cloud into the corresponding image coordinate $p'(x^{'}, y^{'})$.
- **`PMF`** [27]: For 3D point cloud segmentation, this paper designs a two-stream network to fuse the rich semantic

Table 3. **Evaluation Metrics in `Seen`.** Objective results of each affordance type for all comparison methods in the **`Seen`**. "cont." denotes "contain", "supp." denotes "support", "wrap." denotes "wrapgrasp", and "disp." denotes "display".

Method	Metrics	grasp	cont.	lift	open	lay	sit	supp.	wrap.	pour	move	disp.	push	listen	wear	press	cut	stab
`Baseline`	AUC	63.31	51.94	91.1	75.17	59.84	76.43	71.24	52.83	83.79	60.23	73.01	62.16	51.20	53.99	72.82	80.46	61.13
	aIOU	3.89	4.65	9.55	4.17	6.69	8.70	7.33	3.88	7.04	4.69	7.66	4.20	5.21	4.16	4.93	5.32	5.57
	SIM	0.387	0.352	0.155	0.146	0.305	0.363	0.524	0.577	0.418	0.387	0.288	0.557	0.356	0.539	0.102	0.478	0.209
	MAE	0.157	0.154	0.129	0.121	0.214	0.177	0.164	0.145	0.125	0.158	0.199	0.089	0.192	0.154	0.140	0.129	0.137
`MBDF` [22]	AUC	58.18	76.21	76.70	69.35	86.71	94.72	84.84	58.00	73.60	50.25	78.99	63.73	68.09	65.60	87.25	80.99	64.75
	aIOU	6.10	8.60	12.06	5.20	10.79	24.11	10.51	3.97	8.07	5.83	10.70	5.96	4.19	4.49	6.59	5.63	5.87
	SIM	0.397	0.386	0.162	0.154	0.466	0.561	0.631	0.590	0.392	0.389	0.449	0.553	0.405	0.561	0.222	0.430	0.306
	MAE	0.162	0.150	0.129	0.139	0.151	0.109	0.128	0.147	0.148	0.191	0.147	0.121	0.166	0.144	0.131	0.126	0.142
`PMF` [27]	AUC	60.90	73.90	78.03	70.73	87.48	95.22	82.66	57.56	82.71	50.14	80.22	63.94	61.34	61.53	87.96	82.09	65.44
	aIOU	7.12	8.41	11.49	6.94	14.52	25.50	7.83	4.21	8.92	5.91	15.86	6.18	3.15	2.87	6.76	3.03	4.09
	SIM	0.426	0.385	0.193	0.180	0.486	0.598	0.653	0.565	0.397	0.380	0.470	0.535	0.412	0.548	0.237	0.421	0.356
	MAE	0.161	0.148	0.164	0.143	0.143	0.098	0.126	0.152	0.147	0.186	0.137	0.115	0.122	0.154	0.121	0.132	0.148
`FRCNN` [25]	AUC	62.56	77.88	77.71	74.41	88.69	95.72	82.59	53.73	77.38	53.73	79.53	62.51	69.55	68.83	87.46	80.24	68.84
	aIOU	7.36	9.12	11.51	7.39	15.03	25.17	7.76	4.05	7.05	6.29	15.93	5.17	4.70	2.79	6.58	4.97	4.41
	SIM	0.423	0.387	0.182	0.189	0.494	0.591	0.644	0.559	0.399	0.376	0.475	0.543	0.431	0.560	0.234	0.421	0.385
	MAE	0.148	0.144	0.160	0.137	0.137	0.094	0.124	0.132	0.136	0.173	0.136	0.116	0.162	0.148	0.119	0.131	0.132
`ILN` [4]	AUC	62.45	76.43	77.17	74.84	88.08	94.99	82.77	51.84	82.59	52.15	80.08	62.93	68.68	64.34	87.84	82.59	66.51
	aIOU	8.02	8.13	10.42	6.69	16.35	26.72	7.90	4.97	8.18	5.97	17.25	4.84	5.10	3.46	6.93	5.51	4.22
	SIM	0.393	0.388	0.212	0.183	0.512	0.613	0.654	0.557	0.394	0.386	0.497	0.547	0.410	0.548	0.231	0.440	0.379
	MAE	0.144	0.148	0.158	0.141	0.133	0.095	0.127	0.136	0.147	0.172	0.132	0.107	0.173	0.146	0.108	0.123	0.139
`PFusion` [24]	AUC	65.12	78.43	84.52	74.12	89.50	95.47	82.88	51.00	82.02	52.37	84.69	67.45	69.98	67.08	87.63	85.94	67.30
	aIOU	9.25	9.86	11.43	8.48	12.07	25.39	7.59	4.95	6.07	6.22	18.14	3.83	6.63	4.68	7.42	6.22	4.26
	SIM	0.387	0.392	0.223	0.156	0.445	0.616	0.651	0.573	0.396	0.387	0.505	0.569	0.490	0.554	0.230	0.402	0.385
	MAE	0.139	0.138	0.133	0.132	0.138	0.095	0.125	0.131	0.157	0.196	0.128	0.096	0.173	0.148	0.103	0.142	0.155
`XMF` [1]	AUC	62.98	79.97	78.93	79.74	88.93	94.91	84.56	56.47	82.15	54.97	82.91	63.74	74.93	70.57	88.18	79.17	69.97
	aIOU	12.82	10.41	13.24	9.17	17.00	26.65	7.89	5.24	7.81	6.46	17.25	2.68	5.31	5.34	7.87	6.35	5.88
	SIM	0.415	0.401	0.334	0.184	0.427	0.619	0.659	0.569	0.399	0.391	0.508	0.535	0.433	0.565	0.237	0.427	0.394
	MAE	0.121	0.131	0.138	0.131	0.129	0.093	0.119	0.128	0.159	0.192	0.127	0.083	0.164	0.129	0.113	0.135	0.152
`Ours`	AUC	77.53	83.84	95.05	90.89	93.54	95.94	84.58	66.71	86.02	63.09	89.29	84.71	87.13	71.16	89.46	86.69	76.41
	aIOU	16.83	17.12	31.95	28.39	31.80	37.72	12.04	6.02	20.33	5.57	30.57	1.79	15.59	6.55	14.42	12.95	9.48
	SIM	0.530	0.534	0.368	0.401	0.685	0.723	0.716	0.571	0.525	0.443	0.657	0.418	0.671	0.563	0.402	0.507	0.280
	MAE	0.108	0.093	0.030	0.044	0.081	0.066	0.100	0.143	0.096	0.174	0.084	0.085	0.090	0.129	0.059	0.085	0.099

information provided by images with point cloud features to obtain more fine-grained results. For the multi-modal feature fusion, it devises a residual-based fusion model to concatenate image and point cloud features and uses convolution and attention to calculate the fusion feature, and finally, residual connection with the point cloud features. We use the RF module to fuse features that are from different sources and remove the step which projects the point cloud to the camera coordinate system by perspective projection.

– **`FusionRCNN (FRCNN)`** [25]: This work first extracts proposals in the image and point cloud respectively and then fuses these proposals by cross-attention and self-attention. Specifically, for the extracted point cloud and image features, performing a self-attention on both features, then, the point cloud feature is regarded as a query, and the image feature is regarded as key and value, using a cross-attention to fuse them. In the whole pipeline, this operation is performed twice. We fuse the image and point cloud feature with the aforementioned scheme.

– **`ImloveNet (ILN)`** [4]: This research uses images to support the registration of low-overlap point cloud pairs. Its purpose is to use images to provide information for the low-overlap regions of point cloud pairs, so as to support the registration, it is the SOTA on low-overlap point cloud pairs registration task. It also extracts the image and point cloud features separately, and in the fusion module, it projects image features into the 3D feature space through a learnable map-

Table 4. **Evaluation Metrics in `Unseen`.** Objective results of each affordance type for all comparison methods in the **`Useen`**. "cont." denotes "contain", "wrap." denotes "wrapgrasp"

Method	Metrics	cont.	lay	sit	wrap.	open	display	stab	grasp	press	cut
`Baseline`	**AUC**	60.15	76.08	63.89	40.15	72.41	37.84	51.37	43.77	61.51	69.55
	aIOU	4.49	10.31	5.16	1.30	3.30	2.88	3.06	2.41	4.18	5.39
	SIM	0.351	0.450	0.370	0.448	0.126	0.061	0.267	0.166	0.228	0.379
	MAE	0.157	0.153	0.157	0.181	0.117	0.385	0.147	0.146	0.141	0.112
`MBDF` [22]	**AUC**	62.81	76.80	64.30	41.47	73.25	45.26	61.75	46.62	65.57	75.69
	aIOU	5.09	11.28	5.63	1.52	3.68	3.94	3.35	2.42	4.64	6.36
	SIM	0.364	0.464	0.379	0.418	0.133	0.148	0.279	0.168	0.236	0.382
	MAE	0.151	0.149	0.152	0.177	0.108	0.286	0.124	0.150	0.137	0.098
`PMF` [27]	**AUC**	64.10	80.54	64.89	42.02	74.86	51.62	68.93	48.44	65.98	79.05
	aIOU	5.15	13.16	5.83	1.59	3.85	3.19	4.31	2.83	4.97	7.93
	SIM	0.368	0.465	0.381	0.448	0.139	0.123	0.313	0.174	0.245	0.387
	MAE	0.148	0.147	0.153	0.172	0.102	0.262	0.113	0.144	0.135	0.094
`FRCNN` [25]	**AUC**	64.11	84.18	66.37	44.27	74.77	48.12	71.58	49.32	67.58	82.46
	aIOU	5.54	13.72	6.28	1.56	3.96	4.64	4.47	3.13	5.13	8.74
	SIM	0.384	0.481	0.389	0.463	0.143	0.131	0.341	0.189	0.263	0.394
	MAE	0.142	0.142	0.147	0.173	0.099	0.258	0.107	0.137	0.130	0.085
`ILN` [4]	**AUC**	66.76	83.17	65.87	42.21	73.75	54.5	68.48	48.97	66.51	81.42
	aIOU	5.87	13.39	5.78	1.71	3.87	4.73	4.39	3.05	5.05	8.17
	SIM	0.382	0.474	0.385	0.419	0.140	0.118	0.312	0.187	0.259	0.392
	MAE	0.145	0.145	0.151	0.167	0.096	0.274	0.112	0.129	0.132	0.091
`PFusion` [24]	**AUC**	65.56	83.35	67.54	42.70	76.03	56.93	69.05	52.23	68.81	82.39
	aIOU	5.92	13.56	6.83	1.63	4.14	5.14	4.27	3.79	5.25	9.64
	SIM	0.396	0.483	0.391	0.47	0.151	0.072	0.334	0.234	0.262	0.413
	MAE	0.140	0.139	0.145	0.163	0.096	0.255	0.091	0.123	0.128	0.083
`XMF` [1]	**AUC**	67.98	84.02	68.45	45.74	78.53	62.2	76.92	59.19	69.32	85.87
	aIOU	6.29	15.10	7.29	1.42	4.32	6.20	6.12	3.97	5.71	13.95
	SIM	0.412	0.503	0.403	0.451	0.156	0.075	0.351	0.278	0.270	0.435
	MAE	0.137	0.135	0.144	0.156	0.094	0.240	0.087	0.117	0.124	0.078
`Ours`	**AUC**	67.96	84.82	71.10	56.39	90.91	85.51	98.83	78.60	68.07	95.95
	aIOU	7.24	18.12	8.47	1.89	12.28	16.28	10.39	4.79	4.22	21.47
	SIM	0.430	0.525	0.407	0.556	0.227	0.393	0.437	0.533	0.194	0.599
	MAE	0.125	0.130	0.143	0.150	0.050	0.130	0.044	0.102	0.122	0.057

ping. Then, applying the attention mechanism fuses point cloud feature, image feature, and the projected 3D feature in turn. We take this mechanism to fuse image and point cloud features in implementation.

– **`PointFusion (PFusion)`** [24]: This is an early work towards 3D object detection. It also extracts the features of the point cloud and image respectively. For the fusion of different modal features, its processing method is relatively simple. The image branch eventually outputs a global feature, while the point cloud branch outputs a global feature and point-wise feature, the two global features and the point-wise feature do dense fusion to get the fusion feature finally. And we apply this operation to implement the fusion of image and point cloud features.

– **`XMFnet (XMF)`** [1]: This study focus on point cloud completion, it is the SOTA on cross-modal point cloud completion task. It proposes XMFnet, which is composed of two modality-specific feature extractors that capture localized features of the input point cloud and image, then, it uses the combined cross-attention and self-attention to fuse the features of the two modalities. And we apply this block to compute the image and point cloud feature in the pipeline.

Table 5. **Techniques.** Results of different techniques for projecting the region relevance in **Seen** and **Unseen**. S-Atten. denotes self-attention, EM-Atten. denotes Expectation-Maximization Attention, and MLP denotes multilayer perception.

Setting	Metrics	S-Atten.	EM-Atten.	MLP
Seen	**AUC**	85.16	82.37	82.93
	aIOU	21.20	16.03	17.56
	SIM	0.564	0.521	0.527
	MAE	0.088	0.102	0.948
Unseen	**AUC**	73.69	67.45	68.12
	aIOU	8.70	6.78	7.04
	SIM	0.383	0.429	0.432
	MAE	0.117	0.174	0.159

Table 6. **Different Backbones.** Results of **Seen** and **Unseen** settings in different backbone networks. En. P indicates the extractor of the point cloud, En. I indicates the extractor of the image. PN is PointNet++ [17], PM is PointMLP [13] and Res is ResNet [8].

Setting	En. P	En. I	AUC	aIOU	SIM	MAE
Seen	**PN**	Res18	85.16	21.20	0.564	0.088
		Res34	85.45	21.32	0.569	0.086
		Res50	85.52	21.40	0.569	0.082
	PM	Res18	84.89	19.47	0.543	0.095
		Res34	84.98	19.66	0.548	0.088
		Res50	85.31	19.93	0.554	0.084
Unseen	**PN**	Res18	73.69	8.70	0.383	0.117
		Res34	73.78	8.70	0.387	0.107
		Res50	73.83	8.82	0.393	0.101
	PM	Res18	70.11	8.29	0.436	0.146
		Res34	70.76	8.67	0.440	0.142
		Res50	71.05	8.83	0.447	0.136

C.2. Metrics of Each Affordance

We give the overall results of each method in the main paper. Here, we display the results of each affordance respectively. The experimental results of all methods in **Seen** are shown in Tab. 3. And **Unseen** results are shown in Tab. 4. As can be seen, our method achieves the best results under most affordance categories, which demonstrates the superiority of our method in grounding 3D object affordance. These results indicate that our model has great performance whether it is for unseen objects or structures that have not been mapped to a certain affordance, which also indicates the stability and the generalization of our model. At the same time, other methods also achieve considerable objective results under our setting, which proves the rationality of the setting.

C.3. Techniques for Establishing Relevance

To investigate the way of mapping the region relevance, we conduct a comparative experiment to explore the performance of the aforementioned three techniques. They are transformer-based (self-attention), expectation-maximization attention (EMA), and multilayer perception (MLP). The results of metrics in **Seen** and **Unseen** are shown in Tab. 5. As can be seen from the table, MLP and EMA get sub-optimal performance. We analyze the possible reason is that the region features are derived from different sources, so there are gaps among corresponding region features, MLP cannot effectively establish the mapping. Similarly, EMA is also difficult to excavate the region correlation with a group of bases. While self-attention calculates the correlation between every two regions from different sources, there exists a certain relative difference in these correlations, and this relative difference could make it match the corresponding regions of different sources. Based on the above results, we finally choose the self-attention.

C.4. Different Backbones

To verify the effectiveness of the framework and the influence of the backbone, we test another backbone network. We use a recently advanced network PointMLP [13] as the point cloud backbone, it also extracts features of the point cloud hierarchically. In addition, we also test the impact of the model scale on performance, for each point cloud backbone, we take ResNet18, ResNet34 and ResNet50 [8] as the image feature extractor respectively. The evaluation results are shown in Tab. 6, as can be seen from the results, the backbone network does not have a significant impact on the final performance. The larger backbone network could improve the performance, but the improvement is relatively limited. To make the model effective and keep it lightweight, we select PointNet++ and ResNet18 as the final backbone networks.

C.5. Different Hyper-parameters

To explore the impact of each hyper-parameter on the total loss, we conduct a comparative experiment of these hyper-parameters. The experimental results are shown in Tab. 7. λ_1 is the coefficient of $\mathcal{L}_{HM}$, and it accounts for the highest proportion of the total loss, the reduction of λ_1 will have a greater impact on the performance. λ_2 is the coefficient of affordance category loss $\mathcal{L}_{CE}$, and λ_3 is the coefficient of the KL loss $\mathcal{L}_{KL}$. The best result is to set λ_2 to 0.3 and λ_3 to 0.5. Whether they increase or decrease, the performance of the model is affected. In addition, we remove $\mathcal{L}_{KL}$ to test the performance, from the result, we can see that lacking $\mathcal{L}_{KL}$ degrades the model performance.

Table 7. **Hyper-Parameters.** The influence of hyper-parameters that balance three losses in the total loss. The last row represents the performance of the model when removing $\mathcal{L}_{KL}$.

λ_1	λ_2	λ_3	Seen				Unseen			
			AUC	aIOU	SIM	MAE	AUC	aIOU	SIM	MAE
1	0.3	0.5	**85.16**	**21.20**	**0.564**	**0.088**	**73.69**	**8.70**	**0.383**	**0.117**
0.8	0.3	0.5	83.16	18.25	0.532	0.116	69.98	7.25	0.421	0.175
1	0.3	0.7	83.78	18.93	0.537	0.112	70.13	7.52	0.429	0.168
1	0.3	0.3	83.49	18.88	0.533	0.114	70.09	7.47	0.426	0.173
1	0.5	0.5	83.93	19.25	0.552	0.104	70.95	7.85	0.432	0.156
1	0.1	0.5	83.42	19.12	0.546	0.115	70.17	7.60	0.425	0.163
1	0.3	0	82.42	16.94	0.528	0.183	68.23	6.92	0.402	0.273

Table 8. **Pairing Count.** One image could be paired with multiple point clouds for training. Different pairing counts have an influence on the model performance.

Setting	Metrics	1	2	4	6
Seen	AUC	84.75	85.16	85.44	84.82
	aIOU	19.45	21.20	21.83	20.15
	SIM	0.540	0.564	0.571	0.558
	MAE	0.095	0.088	0.083	0.090
Unseen	AUC	69.98	73.69	73.72	71.32
	aIOU	8.37	8.70	8.72	8.58
	SIM	0.375	0.383	0.391	0.380
	MAE	0.130	0.117	0.106	0.121

C.6. Different Pairing

Since images and point clouds originate from different physical instances, one image could be paired with multiple point clouds for training, which can increase the diversity of training data pairs and make the trained model more robust. We test the difference in pairing count and the results are in Tab. 8. When the number of pairings is set to 2, the model performs well, and when it is set to 4, it achieves better results, but the training time is doubled. If the number of pairs is 6, due to the limitation of computing resources, the batch size has to be reduced, so the performance drops instead. Considering the above situations, we finally chose to set the pairing count to 2 in the implementation.

C.7. More Visual Results

We show more visual results of comparative methods and our method in **Seen** and **Unseen** partitions. Fig. 5 shows the result in the **Seen** and Fig. 6 shows the result in the **Unseen**. In addition, we also provide more visual results of our method in all partitions, which shows in Fig. 7. It can be seen from the visual results that our model is able to anticipate the accurate affordance of object functional components from diverse interactions and across multiple object categories, reflecting its stability, robustness, and generalization. Plus, for methods that directly map affordance to specific structures, we present a visual comparison of results between it and our method, shown in Fig. 8. It can be seen from the visual results that this type of method is limited in generalization to the unseen structure.

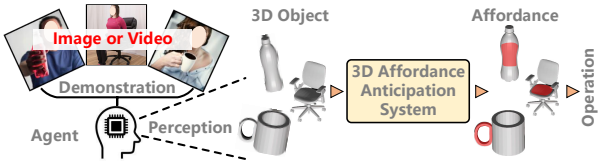

Figure 4. **Potential Applications.** This work has the potential to bridge the gap between perception and operation, serving areas like demonstration learning [2, 19], and may be a part of human-assistant agent system *e.g.* Tesla Bot, Boston Dynamics Atlas [16].

C.8. Partialness and Rotation

Following the setting proposed by 3D-AffordanceNet [7]. We make experiments to test the model performance on partial and freely rotating point clouds, which also simulate the occlusion and rotation of objects in the daily environment. The detailed sampling methods of the partial and freely rotating point clouds could be found in 3D-AffordanceNet [7]. Fig. 9 shows the test results. It can be seen from the results that even if the point cloud only has a partial structure or is randomly rotated, our model can still anticipate 3D object affordance on the corresponding geometric structure. This shows the 2D interactions provide various clues for the model to learn the correlation between geometric structure and affordance. And it could be generated for variable object situations.

D. Potential Applications

Object affordance grounding could serve as a link in the embodied system, supporting many down-stream potential applications, as shown in Fig. 4.

- **Embodied Artificial Intelligence.** Embodied AI [20] is to enable agents to interact with the world from passive perception to active reasoning. The key step to actively understanding the physical world is to know how to interact with the surrounding environment, which is a fundamental skill for embodied agents. And the precondition for the agent to interact with the environment is perceptive the object. Currently, there are some sensors like LIDAR and depth cameras that could sample 3D scene data, and obtain point clouds or depth maps. Meanwhile, many techniques support obtaining a representation of the object from such data. The most direct is to segment or detect the object. Some methods generate the 3D object representation from 2D sources [15]. The above works ensure obtaining

the objects' representation in the scene and support the affordance anticipation system. Anticipating the affordance makes the agent know what action could be done and which location supports the corresponding action on the object representation, which bridges the gap between perception and operation. Such an ability has applications in navigation and manipulation for embodied agents [14, 26].

- **Imitation Learning.** Imitation learning is a common approach to training intelligent agents to perform tasks by observing demonstrations from humans or other agents. However, a key challenge in imitation learning is the ability to generalize to new scenarios and environments that the agent has not encountered during training. This challenge arises because the agent infers the intentions and goals of the demonstrator from a limited set of observed objects or scenes, and must determine how to adapt those actions to the new context. Affordances are the perceived properties of objects or environments that suggest how they can be used or interacted with. In the context of imitation learning, affordances could help the agent to better understand the demonstrator's intentions and goals by identifying the relevant objects and actions in the environment that the demonstrator is using. By recognizing the affordances, the agent can infer the most likely next action, and can therefore learn to imitate their behaviors more accurately [12].

- **Augement Reality.** Augmented reality (AR) is currently considered as having potential for daily applications. By anticipating the 3D affordance of objects in the 3D physical world, more practical functions can be brought to AR devices. For example, if an object needs to be repaired, just send a demo image or video to the user, and then the AR device anticipates the corresponding 3D affordance according to the demo to provide operational guidance. It has high application value in such fields as after-sale service, device maintenance, installation industry, and so on. [5].

- **Virtual Reality.** Nowadays, virtual reality (VR) is more and more widely used in the entertainment, online games, and education industries. It provides a virtual environment for people to interact with the three-dimensional virtual scenario. Some online games or entertainment projects will provide some novel interaction scenes to interact with players. An affordance system can play the role of an NPC to provide interaction guidance for users and improve the user experience. [6].

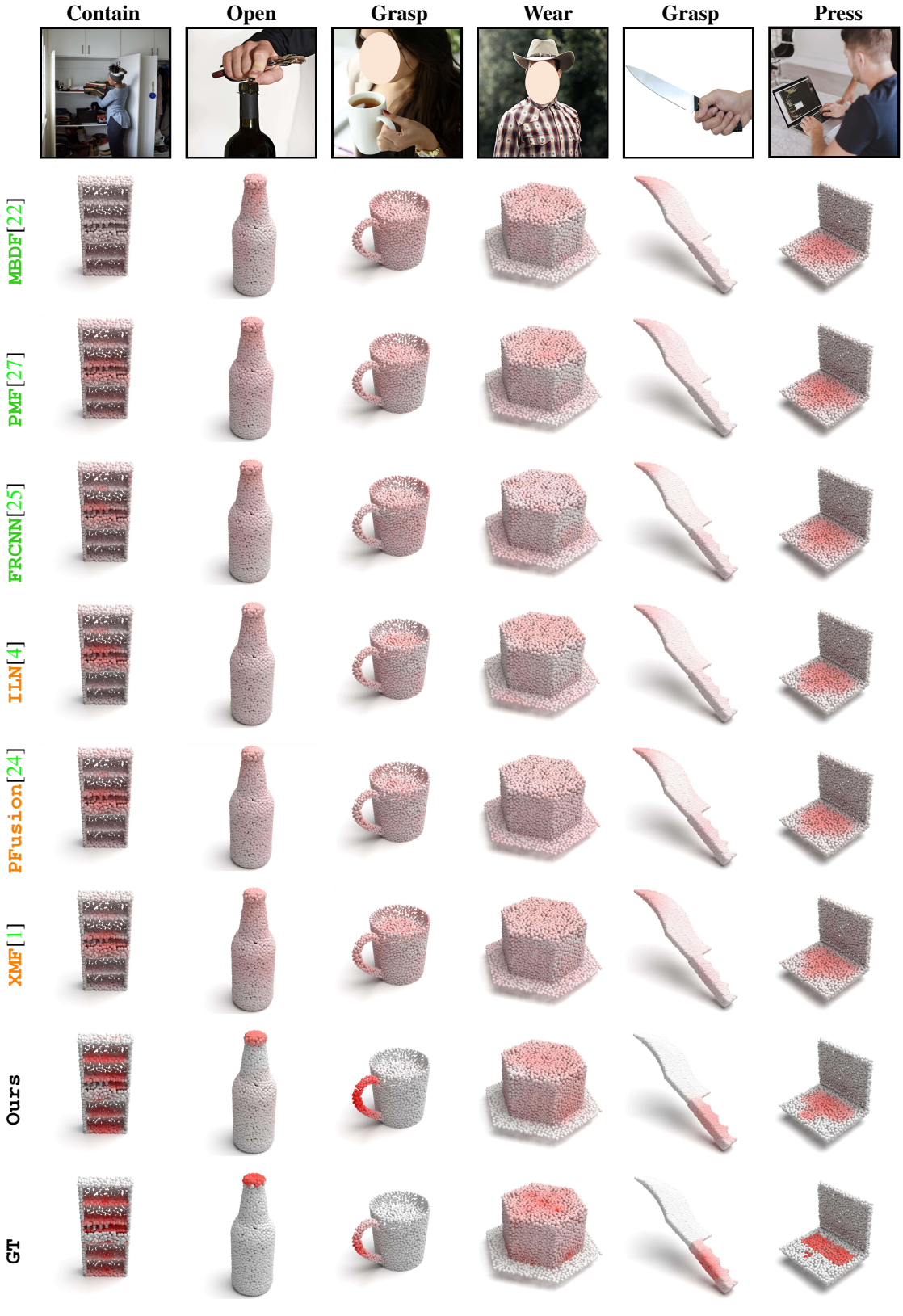

Figure 5. **Visual results in Seen.** We give some visual results of all comparison methods and our method in the **Seen**.

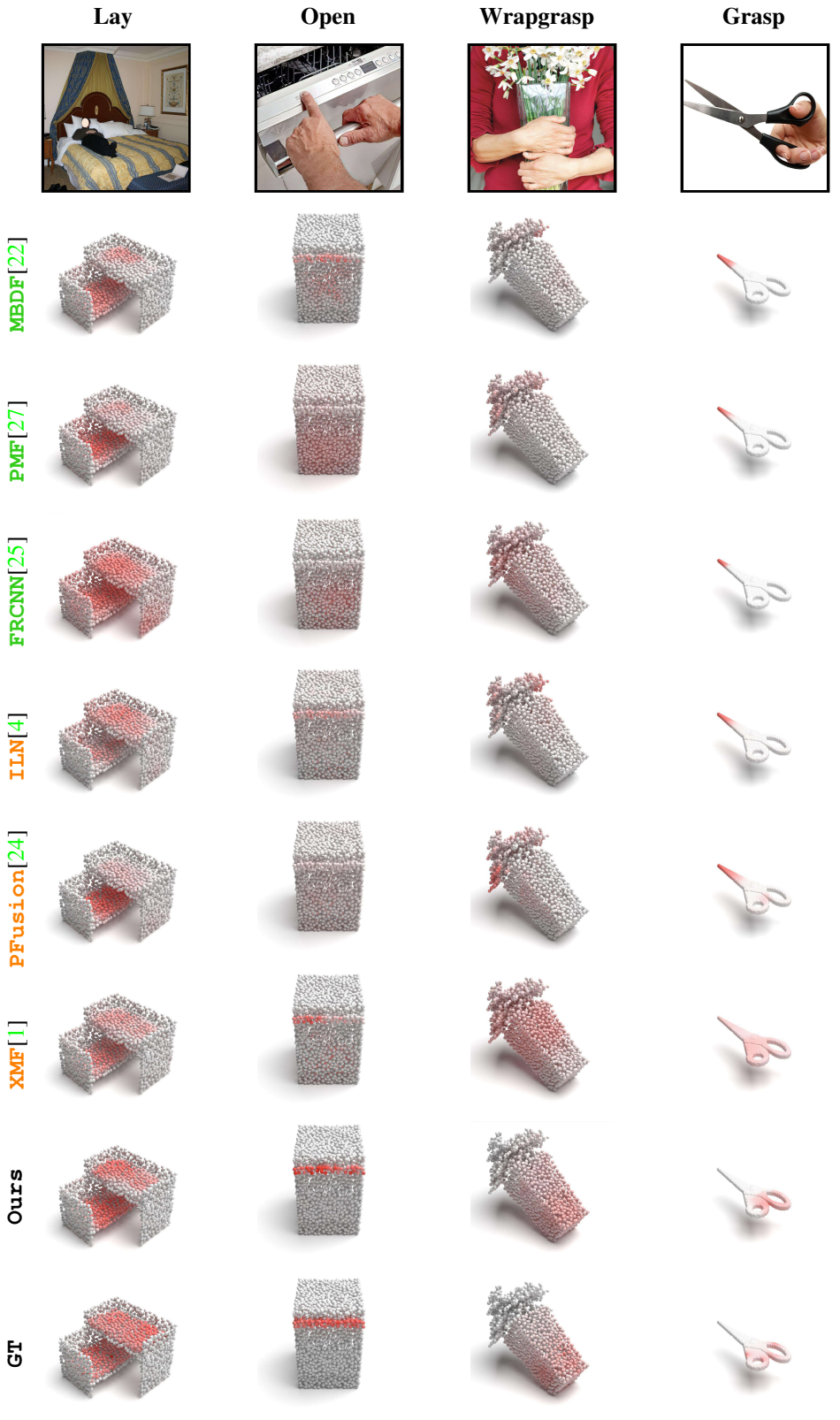

Figure 6. **Visual results.** We give some visual results of all comparison methods and our method in the **Unseen**.

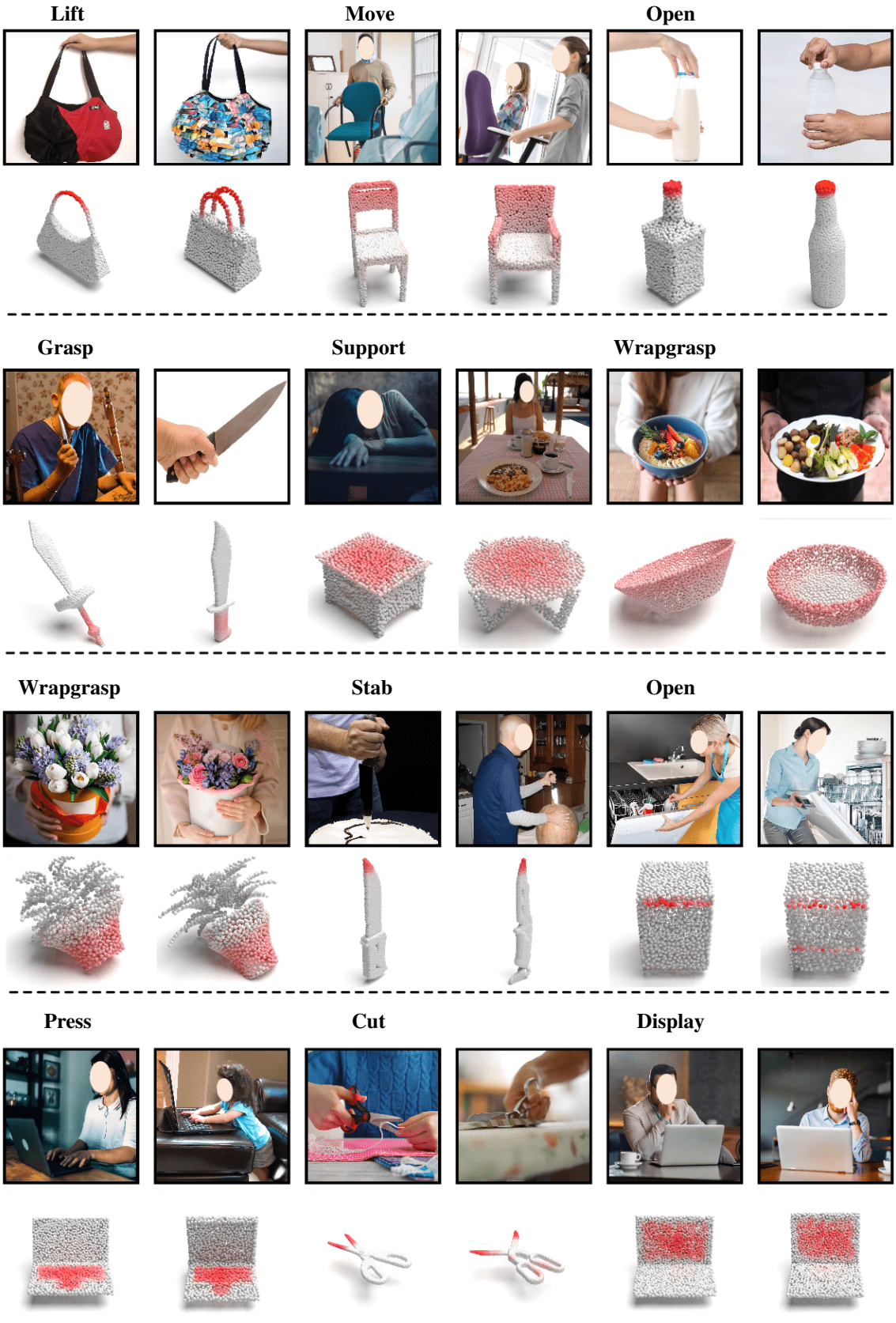

Figure 7. **Visual results of Our Method.** We give more results of our method in **Seen**, **Unseen**.

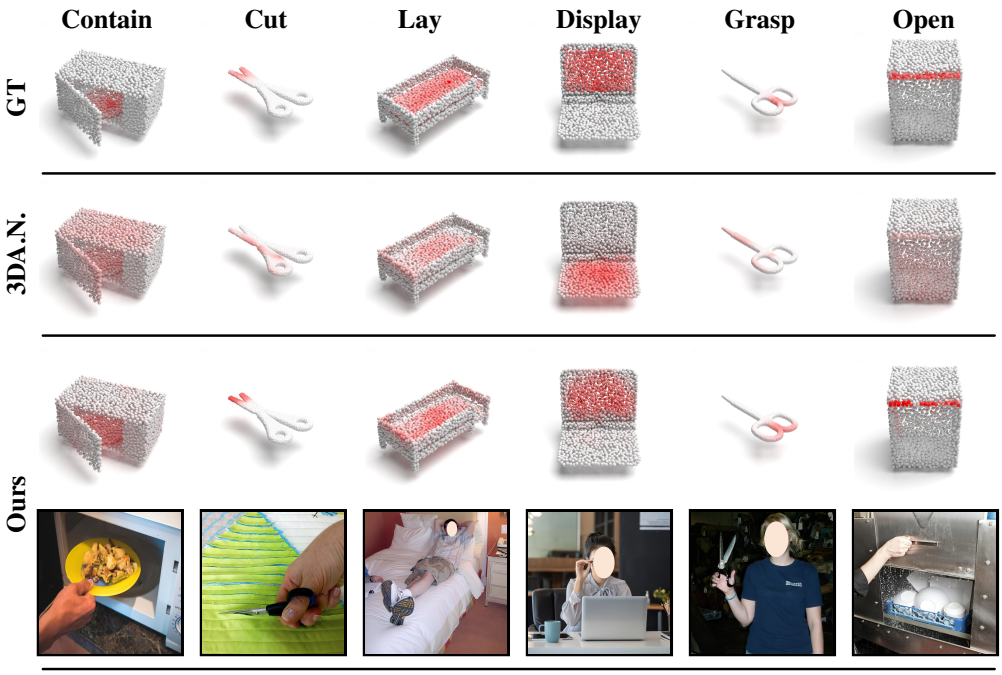

Figure 8. **Generalization in `Unseen`**. 3D.A.N is 3D-AffordanceNet [7]. Visualization results of this method and ours in **`Unseen`**.

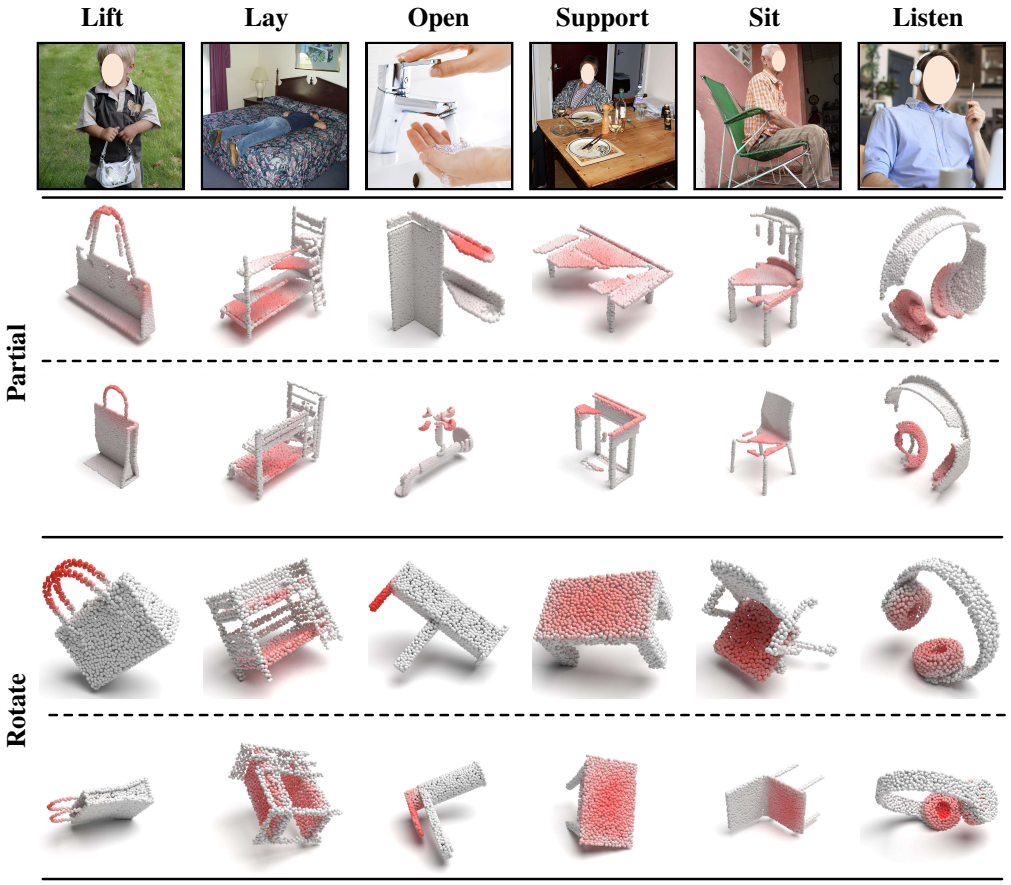

Figure 9. **Partial and Rotate.** Visualization results of partial point cloud and rotated point cloud.

References

[1] Emanuele Aiello, Diego Valsesia, and Enrico Magli. Cross-modal learning for image-guided point cloud shape completion. *arXiv preprint arXiv:2209.09552*, 2022. 5, 6, 10, 11

[2] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009. 8

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018. 2

[4] Honghua Chen, Zeyong Wei, Yabin Xu, Mingqiang Wei, and Jun Wang. Imlovenet: Misaligned image-supported registration network for low-overlap point cloud pairs. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 5, 6, 10, 11

[5] Kun-Hung Cheng and Chin-Chung Tsai. Affordances of augmented reality in science learning: Suggestions for future research. *Journal of science education and technology*, 22(4):449–462, 2013. 9

[6] Barney Dalgarno and Mark JW Lee. What are the learning affordances of 3-d virtual environments? *British Journal of Educational Technology*, 41(1):10–32, 2010. 9

[7] Shengheng Deng, Xun Xu, Chaozheng Wu, Ke Chen, and Kui Jia. 3d affordancenet: A benchmark for visual object affordance understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2021. 8, 13

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[9] Yu Huang, Chenzhuang Du, Zihui Xue, Xuanyao Chen, Hang Zhao, and Longbo Huang. What makes multi-modal learning better than single (provably). *Advances in Neural Information Processing Systems*, 34:10944–10956, 2021. 2

[10] Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9167–9176, 2019. 2

[11] Jorge M Lobo, Alberto Jiménez-Valverde, and Raimundo Real. Auc: a misleading measure of the performance of predictive distribution models. *Global ecology and Biogeography*, 17(2):145–151, 2008. 2

[12] Manuel Lopes, Francisco S Melo, and Luis Montesano. Affordance-based imitation learning in robots. In *2007 IEEE/RSJ international conference on intelligent robots and systems*, pages 1015–1021. IEEE, 2007. 9

[13] Xu Ma, Can Qin, Haoxuan You, Haoxi Ran, and Yun Fu. Rethinking network design and local geometry in point cloud: A simple residual mlp framework. *arXiv preprint arXiv:2202.07123*, 2022. 7

[14] Priyanka Mandikal and Kristen Grauman. Learning dexterous grasping with object-centric visual affordances. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6169–6176. IEEE, 2021. 9

[15] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. *arXiv e-prints*, pages arXiv–2302, 2023. 8

[16] Gabe Nelson, Aaron Saunders, and Robert Playter. The petman and atlas robots at boston dynamics. *Humanoid Robotics: A Reference*, 169:186, 2019. 8

[17] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 7

[18] Md Atiqur Rahman and Yang Wang. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International symposium on visual computing*, pages 234–244. Springer, 2016. 2

[19] Rouhollah Rahmatizadeh, Pooya Abolghasemi, Aman Behal, and Ladislau Bölöni. From virtual demonstration to real-world manipulation using lstm and mdn. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 8

[20] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019. 8

[21] Michael J Swain and Dana H Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991. 2

[22] Xun Tan, Xingyu Chen, Guowei Zhang, Jishiyu Ding, and Xuguang Lan. Mbdf-net: Multi-branch deep fusion network for 3d object detection. In *Proceedings of the 1st International Workshop on Multimedia Computing for Urban Data*, pages 9–17, 2021. 4, 5, 6, 10, 11

[23] Cort J Willmott and Kenji Matsuura. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate research*, 30(1):79–82, 2005. 2, 3

[24] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 244–253, 2018. 5, 6, 10, 11

[25] Xinli Xu, Shaocong Dong, Lihe Ding, Jie Wang, Tingfa Xu, and Jianan Li. Fusionrcnn: Lidar-camera fusion for two-stage 3d object detection. *arXiv preprint arXiv:2209.10733*, 2022. 5, 6, 10, 11

[26] Yuxiang Yang, Zhihao Ni, Mingyu Gao, Jing Zhang, and Dacheng Tao. Collaborative pushing and grasping of tightly stacked objects via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 9(1):135–145, 2021. 9

[27] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16280–16290, 2021. 4, 5, 6, 10, 11