Open-Vocabulary Affordance Detection in 3D Point Clouds

Toan Nguyen[1,2], Minh Nhat Vu[3,4], An Vuong[1], Dzung Nguyen[1], Thieu Vo[5], Ngan Le[6], Anh Nguyen[7]

***Abstract*— Affordance detection is a challenging problem with a wide variety of robotic applications. Traditional affordance detection methods are limited to a predefined set of affordance labels, hence potentially restricting the adaptability of intelligent robots in complex and dynamic environments. In this paper, we present the Open-Vocabulary Affordance Detection (OpenAD) method, which is capable of detecting an unbounded number of affordances in 3D point clouds. By simultaneously learning the affordance text and the point feature, OpenAD successfully exploits the semantic relationships between affordances. Therefore, our proposed method enables zero-shot detection and can be able to detect previously unseen affordances without a single annotation example. Intensive experimental results show that OpenAD works effectively on a wide range of affordance detection setups and outperforms other baselines by a large margin. Additionally, we demonstrate the practicality of the proposed OpenAD in real-world robotic applications with a fast inference speed. Our project is available at https://openad2023.github.io.**

I. INTRODUCTION

The concept of affordance, proposed by the ecological psychologist James Gibson [1], plays an important role in various robotic applications, such as object recognition [2], [3], action anticipation [4], [5], agent's activity recognition [6]–[8], and object functionality understanding [9], [10]. In these applications, affordances are used to illustrate the potential interactions between the robot and its surrounding environment. For instance, with a general cutting task, the knife's affordances can guide the robot to use the knife's blade to achieve requirements such as mincing meat or carving wood. Detecting object affordances, however, is not a trivial task since the robots need to understand in real-time the arbitrary correlations between objects, actions, and effects in complex and dynamic environments [11].

Traditional methods for affordance detection utilize classical machine learning methods on the images, such as Support Vector Machine (SVM) based affordance prediction [12], texture-based and object-level monocular appearance cues [13], relational affordance model [14], and human-object interactions [15]. With the rise of deep learning, several works have employed Convolutional Neural Networks (CNN) [16] for different affordance-related tasks, such as affordance reasoning [17], [18], pixel-based affordance detection [19]–[22], and functional scene understanding [23], [24]. The key challenge in detecting object affordances via the imagery data is that object affordances may differ in terms of visual information such as shape, size, or geometry while being similar in object functionality [25]. In practice, object affordance detection from the images requires an additional step to be applied to downstream robotic tasks, as we need to transform the detected results from 2D to 3D using the depth information [26].

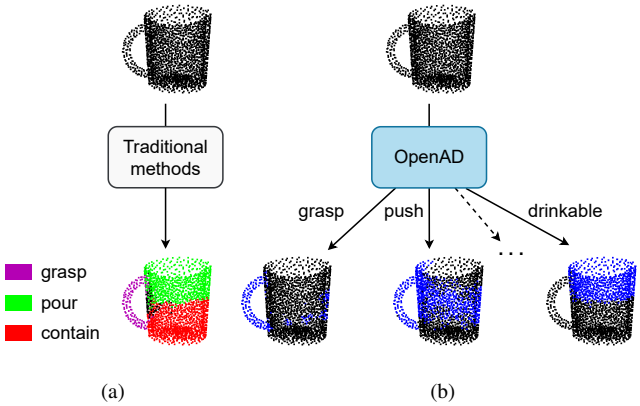

Fig. 1. The comparison between traditional affordance detection methods (a) and our method (b). Traditional methods are restricted to predefined affordance label sets, while our OpenAD enables open-set affordance labels.

With the increasing availability of advanced depth cameras, 3D point cloud has become a popular modality for robotic applications [27]. Compared to conventional images, 3D point clouds directly provide the robot with 3D information about surrounding objects and the environment. Consequently, several recent works directly utilize the 3D point clouds for affordance detection [26], [28]–[30]. For instance, Kim *et al.* [31] detected affordances by dividing point clouds into segments and classifying them using logistic regression. The authors in [32] proposed a new grasp detection method in point cloud data. More recently, Mo *et al.* [26] predicted affordance heat maps from human-object interaction via the scene point cloud. In this work, we address the task of affordance detection in 3D point clouds to directly apply the results to robotic application tasks. More specifically, we consider the affordances of point-level objects and propose a new method to generalize the affordance understanding using the open-vocabulary setting.

While several works have been proposed for affordance understanding using 2D images or 3D point clouds, they are

[1] FPT Software AI Center, Vietnam {toannt28, anvd2, dungnt20}@fpt.com
[2] Faculty of Information Technology, University of Science, Ho Chi Minh City, Viet Nam
[3] Automation & Control Institute (ACIN), TU Wien, Vienna, Austria vu@acin.tuwien.ac.at
[4] Center for Vision, Automation & Control, AIT Austrian Institute of Technology (GmbH), Vienna, Austria minh.vu@ait.ac.at
[5] Faculty of Mathematics and Statistics, Ton Duc Thang University, Ho Chi Minh City, Vietnam vongocthieu@tdtu.edu.vn
[6] Department of Computer Science & Computer Engineering, University of Arkansas thile@uark.edu
[7] Department of Computer Science, University of Liverpool, UK anh.nguyen@liverpool.ac.uk

mostly restricted to a *predefined affordance label set*. This limitation prevents robots from quickly adapting to a wide range of real-world scenarios or responding to changes in the operating environments. Recently, increasing the flexibility of affordance labels has been studied as a one-shot or few-shot learning problem [33], [34]. However, they only consider the 2D images as input and treat the problem as the classical pixel-wise affordance segmentation task. In this work, we overcome the limitation of the fixed affordance label set by addressing the affordance detection in 3D point clouds under the *open-vocabulary* setting. Our key idea is to learn collaboratively the mapping between the *language labels* and the *visual features* of the point cloud. In contrast to traditional methods that are limited to a predefined set of affordance labels, our approach allows the robot to utilize an unrestricted number of natural language texts as input and, therefore, can be used in a broader range of applications. The main concept of our approach is illustrated in Figure 1. Unlike [34], our method does not require annotation examples for unseen affordances and can also work directly with 3D data instead of 2D images.

In this paper, we present Open-Vocabulary Affordance Detection (OpenAD). Our main goal is to provide a framework that does not restrict the application to a fixed affordance label set. Our method takes advantage of the recent large-scale language models, i.e., CLIP [35], and enhances the generalizability of the affordance detection task in 3D point clouds. Particularly, we propose a simple, yet effective method for learning the affordance label and the visual feature together in a shared space. Our method enables zero-shot learning with the ability to process new open-language texts as query affordances.

Our contributions are summarized as follows:

- We present OpenAD, a simple but effective method to tackle the task of open-vocabulary affordance detection.
- We conduct intensive experiments to validate our method and demonstrate the usability of OpenAD in real-world robotic applications.

II. Related Work

Pixel-Wise Affordance Detection. A large number of works consider affordance detection as a pixel-wise labeling task, see, e.g., [19], [20], [22], [36]–[39]. Nguyen *et al.* [22] detected object affordances in real-world scenes using an object detector and dense conditional random fields (CRF). The authors in [20] proposed a two-branch framework that simultaneously recognizes multiple objects and their affordances from RGB images. Chen *et al.* [38] presented a multi-task dense prediction architecture and a tailored training framework to address the problem of joint semantic, affordance, and attribute parsing. More recently, Luo *et al.* [39] proposed a cross-view knowledge transfer framework to extract invariant affordances from exocentric observations and transfer them to egocentric views. Some other works designed different settings for their task of affordance detection on images [15], [40]. In particular, Hassan *et al.* [15] predicted high-level affordances by investigating mutual contexts of humans, objects, and the ambient environment, while Chen *et al.* [40] learned meaningful affordance indicators to predict actions for autonomous driving.

Affordance Detection in 3D Point Clouds. Affordance detection is also studied on 3D point cloud data [26], [28]–[31], [41]. Particularly, Kim *et al.* [31] presented a method to extract geometric features from point cloud segments and classify affordances by logistic regression. Subsequently, Kim and Sukhame [41] presented a technique that voxelizes point cloud objects and creates affordance maps via interactive manipulation. Kokic *et al.* [28] proposed a system for modeling relationships between task, object, and a grasp to address the problem of task-specific robot grasping. Also focusing on detecting grasping affordances, the authors in [29] localized grasping affordances for industrial bin-picking applications. Lately, Mo *et al.* [30] learned affordance heatmaps from object-object interaction. Regardless of performing on images or 3D point clouds, none of the above methods addressed the task of open-vocabulary affordance detection.

Language-Driven Segmentation. Language-driven segmentation has recently attracted research interest in computer vision and machine learning. Promising results of language-driven segmentation are achieved by large-scale language models such as CLIP [35] or BERT [42]. Inspired by these works, several researchers attempted to apply the same idea to other domains, including semantic segmentation. For instance, Li *et al.* [43] aligned pixel-wise features and text features to tackle the task of zero-shot semantic segmentation. With a similar focus, Xu *et al.* [44] proposed a two-stage framework that first extracts mask proposals and then performs classification on the masked image crops generated. Rozenberszki *et al.* [45] proposed a language-driven contrastive pre-training method to address the task of indoor semantic segmentation. Using the pre-trained text encoder from CLIP for open-vocabulary 3D scene understanding, Peng *et al.* [46] proposed a technique that fused 2D and 3D features and aligned the combination with text embeddings. Despite recent development in the field of language-driven segmentation, most recent works have focused only on 2D images, while the task of open-vocabulary affordance detection in 3D point clouds remains an open problem.

III. Open-Vocabulary Affordance Detection

A. Problem Formulation

We consider the task of open-vocabulary affordance detection in 3D point clouds by training a combined vision-language model. Specially, we take into account an input point cloud $\mathcal{C} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ of n unordered points, $\mathbf{p}_i \in \mathbb{R}^3$, $i = 1, ..., n$. Each point is represented by its coordinate in Euclidean space. The affordance labels are presented in a natural language form, i.e., $\mathcal{L} = \{\mathbf{l}_1, \mathbf{l}_2, ..., \mathbf{l}_m\}$. Note that, in our open-vocabulary label setting, the number of labels m can ideally be unlimited. The testing label set can differ from the training label set and can contain unseen affordance labels.

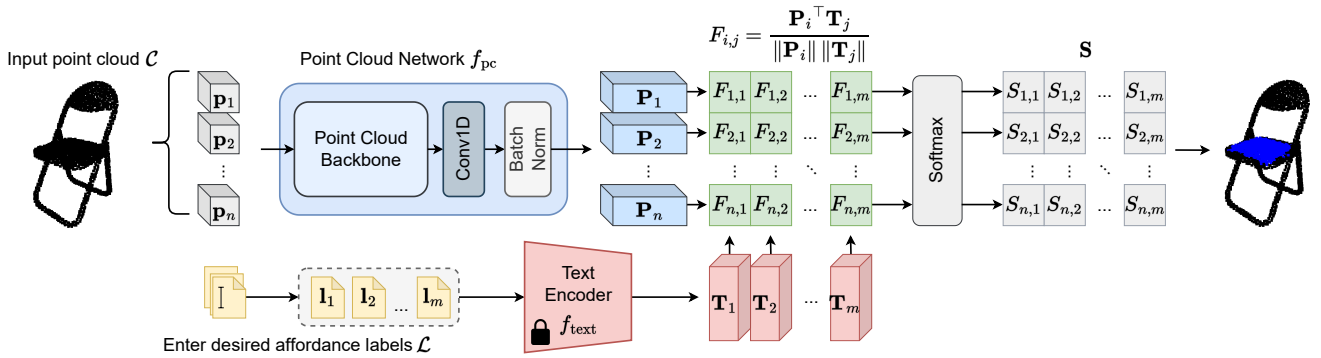

Fig. 2. The overview of the proposed Open-Vocabulary Affordance Detection (OpenAD) network. First, the input point cloud is fed into a point cloud network to extract per-point embeddings. Second, the affordance labels are passed into a text encoder to extract the text embeddings. Subsequently, the correlation between the point-wise features and the corresponding text embeddings is computed using the cosine similarity function. Finally, a softmax layer is employed to predict language-driven affordances.

Our goal is to jointly learn the affordance labels and the input point cloud. We first extract the point cloud feature using a deep point cloud network. The affordance label is encoded by a text encoder. We then introduce a simple, however, effective correlation metric to jointly learn the point cloud visual feature and the text embedding features. In this manner, our method leverages the similarities of text embeddings of unseen affordances that are semantically related to the ones seen in the training process. The overall framework of our method is depicted in Figure 2.

B. Open-Vocabulary Affordance Detection

Text encoder. The text encoder $f_{\text{text}}(\cdot)$ embeds the set of potential affordance labels into an embedding space $\mathbb{R}^D$. Similar to other works [43], [44], the text encoder can be an arbitrary network. In this work, we employ the state-of-the-art pre-trained ViT-B/32 text encoder from CLIP [35]. The text encoder produces m word embeddings $\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_m \in \mathbb{R}^D$ as the presentation of the input affordance labels. Note that we only use the CLIP text encoder to extract the text features and freeze $f_{\text{text}}(\cdot)$ during both training and testing.

Point cloud network. The second component of our method is the point cloud network. The n input points are plugged into the point cloud network $f_{\text{pc}}(\cdot)$ producing an embedding vector for every input point. Similar to the text encoder, the architecture of the point cloud network can be various. In this work, we use the state-of-the-art point cloud model, i.e., PointNet++ [47] as the underlying architecture. Furthermore, we append to the end of the backbone with a convolutional layer with D output units shared across all points, followed by a batch norm layer. More specifically, the point cloud network $f_{\text{pc}}(\cdot)$ produces a set of n vectors $\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_n \in \mathbb{R}^D$. In contrast to the text encoder, the weights of the point cloud network $f_{\text{pc}}(\cdot)$ are updated during the training.

Learning text-point correlation. To enable open-vocabulary affordance detection, the semantic relationships between the point cloud affordance and its potential labels have to be computed. Particularly, we correlate point-wise embeddings of the input point cloud $\mathbf{P}_i$ and text embeddings of affordance labels $\mathbf{T}_j$ using the cosine similarity function. The correlation value $F_{i,j}$, which is an element of the i-th row and j-th column of the correlation matrix $\mathbf{F} \in \mathbb{R}^{n \times m}$, is computed as

$$F_{i,j} = \frac{\mathbf{P}_i^\top \mathbf{T}_j}{\|\mathbf{P}_i\| \, \|\mathbf{T}_j\|} \quad . \tag{1}$$

The point-wise softmax output of a single point i is then computed in the form

$$S_{i,j} = \frac{\exp\left(F_{i,j}/\tau\right)}{\sum_{k=1}^{m} \exp(F_{i,k}/\tau)} \quad , \tag{2}$$

where τ is a learnable parameter [48]. This computation is applied for every point in the point cloud.

During the training, we encourage the point cloud network $f_{\text{pc}}(\cdot)$ to provide point embeddings that are close to the text embeddings. These text embeddings are produced by the text encoder $f_{\text{text}}(\cdot)$ of the corresponding ground-truth classes. Specifically, given the embedding $\mathbf{P}_i \in \mathbb{R}^D$ of point i, we aim to maximize the value of the entry $F_{i,j}$ that is the similarity of $\mathbf{P}_i$ and the text embedding $\mathbf{T}_j$ corresponding to the ground-truth label $j = y_i$. This can be accomplished by optimizing the weighted negative log-likelihood loss of the point-wise softmax output over the entire point cloud in the form

$$L = -\sum_{i=1}^{n} w_{y_i} \log S_{i,y_i} \quad , \tag{3}$$

where w_{y_i} is the weighting parameter to the imbalance problem of the label classes during the training. Inspired by [47], we define this weight as

$$w_j = \left(\frac{\max\{c_1, c_2, ..., c_m\}}{c_j} \right)^{1/3} \quad , \tag{4}$$

where c_j is the number of points of the class j of the training set.

C. Training and Inference

During the training, we fix the text encoder and train the rest of the network end-to-end. Similar to [26], we fix the number of points in a point cloud to $n = 2048$. We set D to

512. We train our network using the Adam optimizer with the learning rate $\alpha = 10^{-3}$ and the weight decay $\gamma = 10^{-4}$. The proposed framework is trained over 200 epochs on a 24GB-RAM NVIDIA Geforce RTX 3090 Ti with a batch size of 16. We initialize the value of $\ln(1/0.07)$ for τ. During the inference, we feed the point cloud and any text as the input label to detect the wanted affordance from the point cloud. The inference process takes approximately $100\,\mathrm{ms}$ on average with our proposed network.

IV. Experiments

In this section, we perform several experiments to validate the effectiveness of our OpenAD. We start with a zero-shot detection setting to verify the ability of OpenAD to generalize to previously unseen affordances. Secondly, we present OpenAD's notable qualitative results together with visualizations. Finally, we conduct additional ablation studies to further investigate other aspects of OpenAD.

A. Zero-Shot Open-Vocabulary Affordance Detection

Dataset. We use the 3D AffordanceNet dataset [26] in our experiments. 3D AffordanceNet dataset is currently the largest dataset for affordance understanding using 3D point cloud data with $22,949$ instances from 23 object categories and 18 affordance labels. As in other zero-shot setups [49]–[51], we need more label classes to verify the robustness of the methods, therefore, we re-label the 3D AffordanceNet with the extra 18 affordance classes and also consider the background as a class, hence making the total number of affordance labels 37. Following [26], we benchmark on the two tasks: the full-shape and partial-view tasks. The partial-view setup is more useful in robotics as the robot usually can only observe a partial-view of the object's point cloud.

Baselines and Evaluation Metrics. We compare our method with the following recent methods for zero-shot learning in 3D point clouds: ZSLPC [49], TZSLPC [50], and 3DGenZ [51]. Note that, since these baselines used GloVe [52] or Word2Vec [53] for word embedding, which are less powerful models than CLIP, we replace their original text encoders with CLIP for a fair comparison. For ZSLPC [49] and TZSLPC [50], we change their classification heads to the segmentation task. As in [47], [54], [55], we use three metrics to evaluate the results: mIoU (mean IoU over all classes), Acc (overall accuracy over all points), and mAcc (mean accuracy over all classes).

Resutls. Table I shows that our OpenAD achieves the best results on both tasks and all three metrics. In particular, on the full-shape task, OpenAD significantly surpasses the runner-up model (ZSLPC) by 4.40% on mIoU. OpenAD also outperforms others on Acc (0.84% over 3DGenZ) and on mAcc (0.81% over ZSLPC). Similarly, for the partial-view task, our method has the highest mIoU (3.98% higher than the second-best ZSLPC), and also the highest Acc and mAcc.

TABLE I
Zero-shot Open-vocabulary detection results

Task	Method	mIoU	Acc	mAcc
Full-shape	TZSLPC [50]	3.86	42.97	10.37
	3DGenZ [51]	6.46	45.47	18.33
	ZSLPC [49]	9.97	40.13	18.70
	OpenAD (ours)	**14.37**	**46.31**	**19.51**
Partial-view	TZSLPC [50]	4.14	42.76	8.49
	3DGenZ [51]	6.03	45.24	15.86
	ZSLPC [49]	9.52	40.91	17.16
	OpenAD (ours)	**12.50**	**45.25**	**17.37**

B. Qualitative Results

We present several examples to demonstrate the generality and flexibility of OpenAD. Primarily, we use objects from the 3D AffordanceNet [26] for our visualizations. We also select objects from the ShapeNetCore dataset [56] to analyze the capability of OpenAD to generalize to unseen object categories and new affordance labels.

Generalization to New Affordance Labels. We illustrate several examples showing the ability of OpenAD to generalize to unseen affordance classes in Figure 3. In the upper row of Figure 3, we present the detection results of OpenAD for nine seen affordances on appropriate objects. As trained on these affordances, OpenAD produces good detection results. Next, for each object, we feed new affordance labels to the models while keeping the same corresponding object, and present the detection results in the two rows below. The visualization shows that OpenAD successfully detects the associated regions for the queried new affordance labels, even though the labels are not included in the training set.

Generalization to Unseen Object Categories. In this work, OpenAD is trained on 3D AffordanceNet dataset [26], which covers 23 object categories. To verify the generalization ability of our method on new unseen objects, we select new novel objects from the ShapeNetCore dataset [56] and test them in both cases: seen and open-vocabulary affordance labels. We use the farthest point sampling algorithm to uniformly sample $2,048$ points from the surface of each object. All points are then centered and scaled before being fed into OpenAD. Figure 4 summarises the results. This figure shows that OpenAD is able to detect the affordance classes on new object categories. This confirms the generalization of our OpenAD for downstream robotic tasks.

Multi-Affordance Detection. In OpenAD, the number of affordances in the label set, m, can vary. This flexible design allows OpenAD to detect multiple affordances at once. Figure 5 presents the detection results of two objects given label sets with different numbers of affordances. Moreover, we observe a notable ability of OpenAD that it does not fix the label for a particular point but finds the most suitable label in a specific label set. Concretely, given an object, OpenAD first detects affordances in a particular label set. By maintaining the earlier affordances, once a new affordance label is added to the label set, there are certain points will be labeled by new affordances over the previous affordance classes. For instance, in the left column of Figure 5, the points in the upper body of the bottle are labeled as `contain` in the first run, then they are re-labeled as `wrap-grasp` in the second run and finally as `grasp` in the last run. This OpenAD's ability can also be observed in the case of the knife object

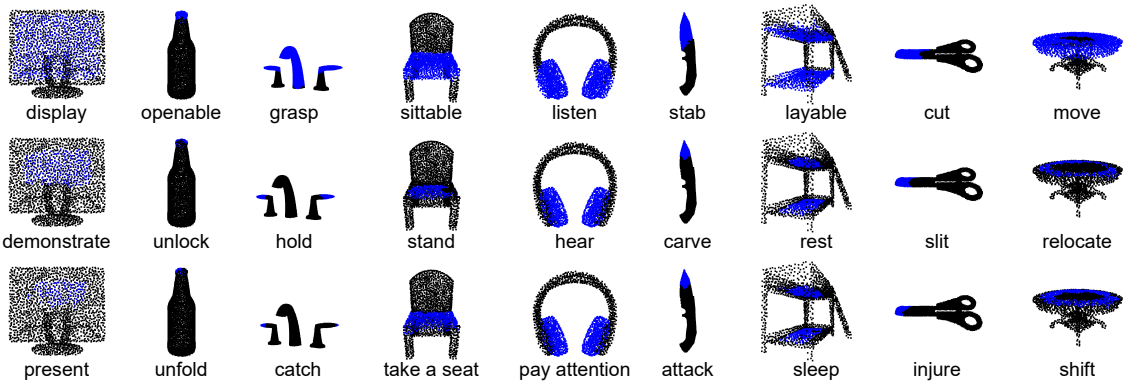

Fig. 3. The visualization of our OpenAD's capability to detect seen and unseen affordances. *The upper row:* Detection results of OpenAD on seen affordances. *The middle and lower rows:* Detection results of OpenAD on unseen affordances that do not exist in the training set.

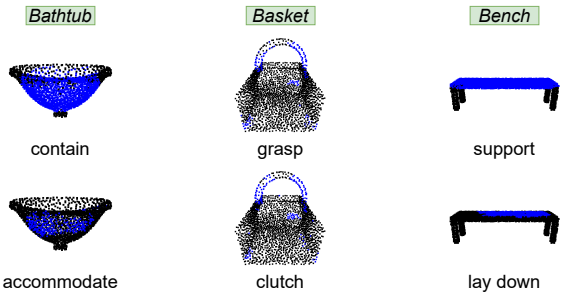

Fig. 4. Results on unseen object categories. OpenAD gives reasonable outputs when detecting affordances on object categories that are not in the training set. *Upper row:* Seen affordances. *Lower row:* Unseen affordances.

in the right column of Figure 5. It further demonstrates our OpenAD's flexibility.

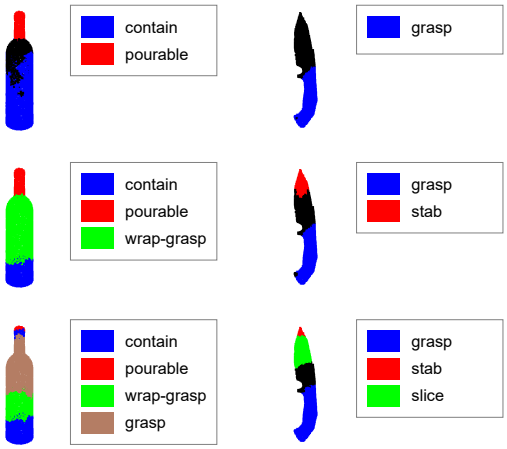

Fig. 5. Multi-affordance detection using our method.

C. Ablation Study

Will language-driven architecture affect the detection results? In this work, we mainly focus on jointly learning a vision-language model to improve the generalization in downstream robotic tasks, and do not aim for improving the accuracy of the traditional affordance detection tasks. This leads to the question that whether our language-driven model affects the accuracy of the traditional affordance detection task. To verify this, we train our OpenAD on the original 3D AffordanceNet with its original label set and training split, and compare our result with other state-of-the-art methods, including PointNet++ [47], Dynamic Graph CNN (DGCNN) [54] and Point Transformer [55]. For PointNet++ and DGCNN, we follow similar designs in [26] and change the final classifier to a linear layer detecting the affordance classes. For Point Transformer, we apply the same architecture in [55]. Table II presents the results of all methods on 3D AffordanceNet [26]. From this table, we find that OpenAD performs competitively when compared to other methods. Therefore, we can conclude that while our method is designed for a different purpose, it still can be used as a strong benchmark for closed-set affordance detection.

TABLE II
CLOSED-SET DETECTION RESULTS

Task	Method	mIoU	Acc	mAcc
Full-shape	Point Transformer [55]	41.26	68.67	67.03
	PointNet++ [47]	41.26	68.51	**68.14**
	DGCNN [54]	**42.09**	68.47	61.47
	OpenAD (ours)	42.00	**69.03**	67.31
Partial-view	Point Transformer [55]	40.51	69.22	65.34
	PointNet++ [47]	41.10	69.40	**66.74**
	DGCNN [54]	**41.93**	69.38	63.12
	OpenAD (ours)	41.87	**69.91**	66.34

Backbones and Text Encoders. In table III, we conduct an ablation study on two different point cloud backbones, i.e., PointNet++ [47] and DGCNN [54], and two different pre-trained text encoders, i.e., CLIP ViT-B/32 [35] and BERT [42]. Note that with BERT, the parameter D is set to 768. We observe that different combinations of backbones and text encoders perform equivalently on the closed-set tasks. Meanwhile, on the open-vocabulary tasks, PointNet++ performs better than DGCNN, and CLIP performs better than BERT. The gap in the performance of frameworks using CLIP ViT-B/32 text encoder compared to those using BERT is significant, demonstrating the superiority of CLIP in semantic language-vision understanding.

D. Robotic Demonstration

The experiment setup, shown in Figure 6, comprises five main components, i.e. the robot KUKA LBR iiwa R820, the PC1 running the real-time automation software Beckhoff TwinCAT, the Intel RealSense D435i camera, the Robotiq

TABLE III

ABLATION STUDY ON POINT CLOUD BACKBONE AND TEXT ENCODER

Task & setting	Point cloud backbone	Text encoder	mIoU	Acc	mAcc
Full-shape & Open-vocabulary	PointNet++	CLIP	**14.37**	46.31	**19.51**
	PointNet++	BERT	10.55	**46.81**	16.02
	DGCNN	CLIP	10.88	45.21	15.40
	DGCNN	BERT	9.43	46.37	14.61
Partial-view & Open-vocabulary	PointNet++	CLIP	**12.50**	45.25	**17.37**
	PointNet++	BERT	9.33	**45.71**	14.45
	DGCNN	CLIP	11.19	44.21	16.43
	DGCNN	BERT	7.00	44.93	10.97
Full-shape & Closed-set	PointNet++	CLIP	42.00	**69.03**	**67.31**
	PointNet++	BERT	41.04	68.79	67.22
	DGCNN	CLIP	**42.11**	68.59	61.31
	DGCNN	BERT	42.06	68.31	61.26
Partial-view & Closed-set	PointNet++	CLIP	**41.87**	**69.91**	**66.34**
	PointNet++	BERT	41.50	69.77	65.55
	DGCNN	CLIP	41.20	69.16	64.16
	DGCNN	BERT	41.27	69.62	58.09

2F-85 gripper, and the PC2 running Robot Operating System (ROS) Noetic 20.04. PC1 communicates with the robot via a network interface card (NIC) using the EtherCAT protocol, marked by the blue region in Figure 6. Note that the robot control is implemented in a C++ module in PC1. The sampling time is set to 125 µs for the robot sensors and actuators. PC2 controls the gripper and the camera via the USB protocol in the ROS environment. Additionally, the two PCs communicate with each other via an Ethernet connection. After receiving point cloud data of the environment from the RealSense D435i camera, we utilize the state-of-the-art object localization method [57] to identify the object, then perform point sampling to get 2048 points. We then feed this point cloud with a natural language affordance command. Note that, using our OpenAD, we can have a general input command and are not restricted to a predefined affordance label set. Our OpenAD returns the affordance region, which can be used for the grasp pose detection module [32], the analytical inverse kinematics module [58], and the trajectory optimization module [59]. Several demonstrations, such as holding and raising a bag, wrapping a bottle, and pushing an earphone, can be found in our supplementary material.

E. Discussion

Despite achieving promising results, OpenAD has its limitation. Our method is still far from being able to detect completely unseen affordances. The upper row of Figure 7 shows cases when OpenAD fails to detect unseen affordances on seen objects. Moreover, we present false-positive predictions of OpenAD in the lower row of Figure 7. In these cases, OpenAD detects affordances that the objects do not provide, i.e., `display` for the bag, `support` for the microwave, and `openable` for the hat. From our intensive experiments, we see several improvement points for future work: *i)* learning the visual-language correlation plays an important role in this task and can be further improved by using more complicated techniques such that cross-attention mechanism [60] , *ii)* applying a stronger point cloud backbone would likely improve

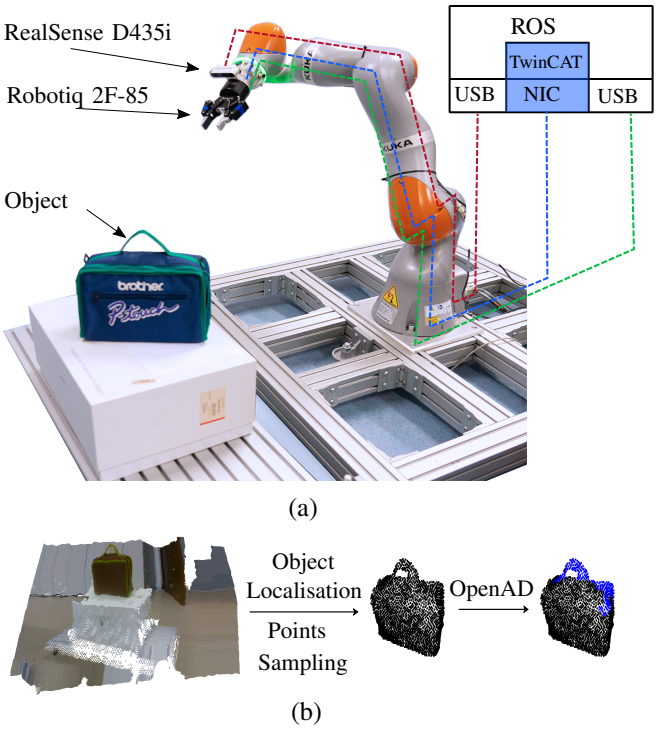

Fig. 6. Example of a robot demonstration. (a) Experimental setup. (b) Result from OpenAD.

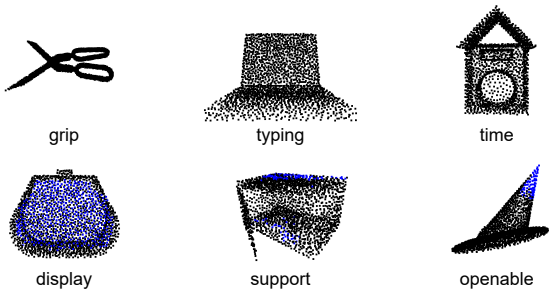

Fig. 7. Failure cases of OpenAD. *Upper row:* Cases when OpenAD fails to detect unseen affordances. *Lower row:* OpenAD detects affordances that are not furnished by the objects.

the result, and *iii)* having a large-scale dataset with several affordance classes would be beneficial for benchmarking and real-world robotic applications. Finally, we will also release our source code to encourage further study.

V. CONCLUSIONS

We proposed OpenAD, a simple yet effective method for open-vocabulary affordance detection in 3D point clouds. Different from traditional approaches, OpenAD, with its capability of semantic understanding, can effectively detect unseen affordances without requiring annotated examples. Empirical results show that OpenAD outperforms other methods by a large margin. We further verified the capability of OpenAD to detect unseen affordances on both known and unseen objects. We additionally demonstrated the usability of OpenAD in real-world robotic applications. Although there is ongoing work to be accomplished, OpenAD's results provide encouraging evidence that intelligent robots understand many possibilities and perform better in complex environments.

REFERENCES

[1] J. J. Gibson, *The senses considered as perceptual systems.* Bloomsbury Academic: Michigan, USA, 1966.

[2] S. Thermos, G. T. Papadopoulos, P. Daras, and G. Potamianos, "Deep affordance-grounded sensorimotor object recognition," in *CVPR*, 2017.

[3] Z. Hou, B. Yu, Y. Qiao, X. Peng, and D. Tao, "Affordance transfer learning for human-object interaction detection," in *CVPR*, 2021.

[4] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *CVPR*, 2016.

[5] D. Roy and B. Fernando, "Action anticipation using pairwise human-object interactions and transformers," *TIP*, 2021.

[6] T.-H. Vu, C. Olsson, I. Laptev, A. Oliva, and J. Sivic, "Predicting actions from static scenes," in *ECCV*, 2014.

[7] S. Qi, S. Huang, P. Wei, and S.-C. Zhu, "Predicting human activities using stochastic grammar," in *ICCV*, 2017.

[8] J. Chen, D. Gao, K. Q. Lin, and M. Z. Shou, "Affordance grounding from demonstration video to target image," in *CVPR*, 2023.

[9] G. Li, V. Jampani, *et al.*, "Locate: Localize and transfer object parts for weakly supervised affordance grounding," in *CVPR*, 2023.

[10] J. Jiang, G. Cao, T.-T. Do, and S. Luo, "A4T: Hierarchical affordance detection for transparent objects depth reconstruction and manipulation," *RA-L*, 2022.

[11] H. Min, R. Luo, J. Zhu, and S. Bi, "Affordance research in developmental robotics: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, 2016.

[12] T. Hermans, J. M. Rehg, and A. Bobick, "Affordance prediction via learned object attributes," in *ICRAW*, 2011.

[13] H. O. Song, M. Fritz, C. Gu, and T. Darrell, "Visual grasp affordances from appearance-based cues," in *ICCVW*, 2011.

[14] B. Moldovan and L. De Raedt, "Occluded object search by relational affordances," in *ICRA*, 2014.

[15] M. Hassan and A. Dharmaratne, "Attribute based affordance detection from human-object interaction images," in *Proceedings of the Image and Video Technology Workshops*, 2016.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, 2017.

[17] R. Mottaghi, C. Schenck, D. Fox, and A. Farhadi, "See the glass half full: Reasoning about liquid containers, their volume and content," in *ICCV*, 2017.

[18] C.-Y. Chuang, J. Li, A. Torralba, and S. Fidler, "Learning to act properly: Predicting and explaining affordances from images," in *CVPR*, 2018.

[19] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Detecting object affordances with convolutional neural networks," in *IROS*, 2016.

[20] T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *ICRA*, 2018.

[21] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Leverage interactive affinity for affordance learning," in *CVPR*, 2023.

[22] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Object-based affordances detection with convolutional neural networks and dense conditional random fields," in *IROS*, 2017.

[23] X. Li, S. Liu, K. Kim, X. Wang, M.-H. Yang, and J. Kautz, "Putting humans in a scene: Learning affordance in 3d indoor environments," in *CVPR*, 2019.

[24] A. Pacheco-Ortega and W. Mayol-Cuervas, "One-shot learning for human affordance detection," in *ECCV*, 2022.

[25] M. Hassanin, S. Khan, and M. Tahtali, "Visual affordance and function understanding: A survey," *CSUR*, 2021.

[26] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3D Affordancenet: A benchmark for visual object affordance understanding," in *CVPR*, 2021.

[27] W. Liu, J. Sun, W. Li, T. Hu, and P. Wang, "Deep learning on point clouds and its application: A survey," *Sensors*, 2019.

[28] M. Kokic, J. A. Stork, J. A. Haustein, and D. Kragic, "Affordance detection for task-specific grasping using deep learning," in *Humanoids*, 2017.

[29] A. Iriondo, E. Lazkano, and A. Ansuategi, "Affordance-based grasping point detection using graph convolutional networks for industrial bin-picking applications," *Sensors*, 2021.

[30] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2O-Afford: Annotation-free large-scale object-object affordance learning," in *CoRL*, 2022.

[31] D. I. Kim and G. S. Sukhatme, "Semantic labeling of 3d point clouds with object affordance for robot manipulation," in *ICRA*, 2014.

[32] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, 2017.

[33] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "One-shot affordance detection," *arXiv:2106.14747*, 2021.

[34] W. Zhai, H. Luo, J. Zhang, Y. Cao, and D. Tao, "One-shot object affordance detection in the wild," *IJCV*, 2022.

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[36] A. Roy and S. Todorovic, "A multi-scale cnn for affordance segmentation in rgb images," in *ECCV*, 2016.

[37] S. Thermos, P. Daras, and G. Potamianos, "A deep learning approach to object affordance segmentation," in *ICASSP*, 2020.

[38] X. Chen, T. Liu, H. Zhao, G. Zhou, and Y.-Q. Zhang, "Cerberus transformer: Joint semantic, affordance and attribute parsing," in *CVPR*, 2022.

[39] H. Luo, W. Zhai, J. Zhang, Y. Cao, and D. Tao, "Learning affordance grounding from exocentric images," in *CVPR*, 2022.

[40] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *ICCV*, 2015.

[41] D. I. Kim and G. S. Sukhatme, "Interactive affordance map building for a robotic task," in *IROS*, 2015.

[42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv:1810.04805*, 2018.

[43] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, "Language-driven semantic segmentation," in *ICLR*, 2021.

[44] M. Xu, Z. Zhang, F. Wei, Y. Lin, Y. Cao, H. Hu, and X. Bai, "A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model," in *ECCV*, 2022.

[45] D. Rozenberszki, O. Litany, and A. Dai, "Language-grounded indoor 3d semantic segmentation in the wild," in *ECCV*, 2022.

[46] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "Openscene: 3d scene understanding with open vocabularies," in *CVPR*, 2023.

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

[48] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, "Unsupervised feature learning via non-parametric instance discrimination," in *CVPR*, 2018.

[49] A. Cheraghian, S. Rahman, and L. Petersson, "Zero-shot learning of 3d point cloud objects," in *ICMVA*, 2019.

[50] A. Cheraghian, S. Rahman, D. Campbell, and L. Petersson, "Transductive zero-shot learning for 3d point cloud classification," in *WACV*, 2020.

[51] B. Michele, A. Boulch, G. Puy, M. Bucher, and R. Marlet, "Generative zero-shot learning for semantic segmentation of 3d point clouds," in *3DV*, 2021.

[52] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014.

[53] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," *NeurIPS*, 2013.

[54] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *TOG*, 2019.

[55] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *ICCV*, 2021.

[56] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv:1512.03012*, 2015.

[57] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, 2018.

[58] M. N. Vu, F. Beck, M. Schwegel, C. Hartl-Nesic, A. Nguyen, and A. Kugi, "Machine learning-based framework for optimally solving the analytical inverse kinematics for redundant manipulators," *Mechatronics*, 2023.

[59] F. Beck, M. N. Vu, C. Hartl-Nesic, and A. Kugi, "Singularity avoidance with application to online trajectory optimization for serial manipulators," *arXiv:2211.02516*, 2022.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *NeurIPS*, 2017.