Variation-Robust Few-Shot 3D Affordance Segmentation for Robotic Manipulation

Dingchang Hu, Tianyu Sun, Pengwei Xie, *Member, IEEE*, Siang Chen, *Graduate Student Member, IEEE*, Huazhong Yang, *Fellow, IEEE*, and Guijin Wang, *Senior Member, IEEE*

***Abstract*—Traditional affordance segmentation on 3D point cloud objects requires massive amounts of annotated training data and can only make predictions within predefined classes and affordance tasks. To overcome these limitations, we propose a variation-robust few-shot 3D affordance segmentation network (VRNet) for robotic manipulation, which requires only several affordance annotations for novel object classes and manipulation tasks. In particular, we design an orientation-tolerant feature extractor to address pose variation between support and query point cloud objects, and present a multi-scale label propagation algorithm for variation in completeness. Extensive experiments on affordance datasets show that VRNet provides the best segmentation performance compared with previous works. Moreover, experiments in real robotic scenarios demonstrate the generalization ability of our method.**

***Index Terms*—Perception for grasping and manipulation, deep learning in grasping and manipulation, recognition.**

I. Introduction

ROBOTIC manipulation technology [1], [2] holds significant potential for practical and intelligent real-world applications. 3D affordance region segmentation, as a fundamental aspect, identifies regions on object point clouds suitable for robot interaction in manipulation tasks. Current affordance estimation methods [3], [4], [5] typically adopt a fully-supervised approach, training a model with affordance annotations for a specific manipulation task and predicting corresponding labels for similar class objects. However, the continuous emergence of novel object classes and manipulation definitions reveals two major drawbacks of the current approach: 1) Requirement of sufficient annotated training examples for every object class and task; 2) Limited extension to other novel class objects and tasks.

Differently, based on meta-learning [6], the few-shot learning mechanism [7], [8] operates in a learning-to-learn mode, where a model is supervised to learn knowledge from only one or several annotated examples and then make predictions on target objects. A comparison between fully-supervised and few-shot learning mechanisms is depicted in Fig. 1. The few-shot model takes limited annotated examples as additional network inputs, termed support objects. As the classes of support objects and annotated affordance regions change, the model has the ability to recognize novel objects of different classes and tasks, termed query objects.

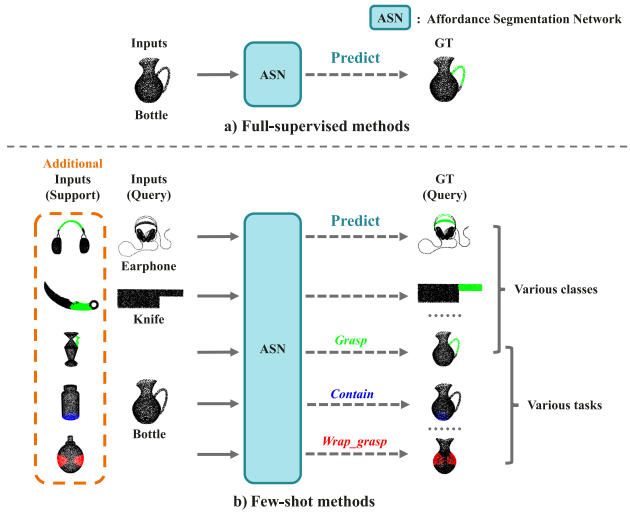

Fig. 1. Comparison between full-supervised and few-shot affordance segmentation methods. The object's affordance regions are marked in different colors, while background regions are marked in black.

Nevertheless, the variations between support and query point cloud objects restrict the performance of the few-shot affordance segmentation model to a certain extent. On the one hand, support and query features extracted from object point clouds are related to their geometric poses, and the pose variation can lead to chaotic feature matching in the subsequent process. On the other hand, the completeness of objects composed by 3D points cannot always be guaranteed, even when multi-view fusion is performed on raw data captured from depth cameras in real-world scenarios. The variation in completeness is likely to result in incorrect label propagation.

In this letter, we integrate meta-learning based few-shot learning into 3D affordance segmentation. With a limited number of examples consisting of object point clouds and their corresponding affordance annotations as support, the few-shot 3D affordance segmentation model can predict affordance regions on similar objects of the same class for the same task. Moreover, the orientation-tolerant feature extractor and multi-scale label propagation effectively alleviate variation issues in the few-shot framework.

Received 6 August 2024; accepted 15 December 2024. Date of publication 1 January 2025; date of current version 9 January 2025. This letter was recommended for publication by Associate Editor Lin Shao and Editor Abhinav Valada upon evaluation of the reviewers' comments. *(Dingchang Hu and Tianyu Sun contributed equally to this letter.) (Corresponding author: Guijin Wang.)*

The authors are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: yanghz@tsinghua.edu.cn; wangguijin@tsinghua.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3524904, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3524904

In summary, our primary contributions are as follows:

- We propose a novel meta-learning based Variation-Robust few-shot Network for robotic manipulation, VRNet, which predicts affordance regions from target point cloud objects with a few annotated examples. As object classes and tasks change, the model is able to achieve 3D affordance estimation adaptively.
- We design an orientation-tolerant feature extractor to address pose variation between support and query point cloud objects, and present a multi-scale label propagation algorithm for variation in completeness.
- Experimental results on datasets and real-world scenarios demonstrate the effectiveness and flexibility of our method.

II. Related Works

A. Affordance Estimation

Robotic manipulation [9], [10], [11] streamlines operations and enhances efficiency in industrial sectors and daily life. Affordance estimation [12], [13] plays a crucial role in the field of robotic manipulation for interactive region localization, and thus has received extensive research attention recently. Deng et al. [3] construct a dataset named 3D AffordanceNet with visual affordance annotations on point cloud objects and design an affordance estimation model for all object shapes and defined manipulation tasks in the dataset. For category-level robotic manipulation, Manuelli et al. [14] locate affordance regions by performing keypoint detection on target objects. Xu et al. [4] propose a multi-task network for joint learning of affordance position and operation direction. Chen et al. [15] discover affordance regions by analyzing object part representations.

Nevertheless, these methods demand a large amount of labeled data for model training when encountering novel object classes or tasks, a situation that always occurs in real-world robotic manipulation.

B. Few-Shot Learning

Few-shot learning [7], [8] enables models to learn from limited data and generalize to new tasks, even the challenging low-level segmentation tasks that require fine-grained estimation. Shaban et al. [16] first introduce meta-learning into plain few-shot segmentation tasks. They represent each foreground and background class with one prototype by calculating the mean of its support features, which serves as the basis for subsequent feature similarity measurement. Later on, Zhang et al. [17] propose a self-guided learning approach to avoid critical information lost during feature extraction. For 3D point cloud scenes, Zhao et al. [18] construct a few-shot 3D framework based on a k-NN graph structure to estimate point-wise semantic labels. Lang et al. [19] predict non-target regions with an additional network branch, distinguishing the target class from training classes. Hu et al. [20] improve the quality of support prototypes through progressive generation using a feedback mechanism.

Unlike previous few-shot methods, we innovatively incorporate few-shot learning into the 3D point cloud affordance estimation task. Furthermore, orientation-tolerant feature extraction and multi-scale label propagation are presented to improve the performance of few-shot 3D affordance segmentation.

III. Problem Statement

Few-shot 3D affordance estimation aims to identify feasible regions for robotic manipulation with only a few labeled 3D data. The training of the model typically adheres to the meta-learning paradigm [6], also known as episode training in conventional few-shot learning. Generally, given two point cloud object sets D_{train} and D_{test} that are disjoint in terms of object categories or manipulation tasks, the few-shot model is expected to learn on D_{train} with sufficient annotations and exhibit good generalization on D_{test} with a few annotated examples. Specifically, D_{train} consists of many episode examples, each of which contains a pair of support examples $S_{train} = (x_i^{S_{train}}, m_i^{S_{train}})_{i=1}^{K}$ and query example $Q_{train} = (x^{Q_{train}}, m^{Q_{train}})$ belonging to the same object category and manipulation task. x and m represent a point cloud object and its affordance label, respectively. K-shot denotes that only K labeled support examples are provided. During training, model f is optimized to predict the affordance label of Q_{train} with the prior knowledge of labels in S_{train}:

$$\widehat{m}^{Q_{train}} = f\left(x^{Q_{train}}, S_{train}\right). \tag{1}$$

Since the S_{train} and Q_{train} of different episodes belong to different object classes and manipulation tasks, the model f is not tied to any specific class or task. Its output will only change in response to variations in label annotations $m^{S_{train}}$ of the current S_{train}. Therefore, the model has the capability to generalize to other novel classes and manipulation tasks in D_{test} with their corresponding S_{test}.

IV. Method

A. Overview

To achieve 3D affordance estimation for different object classes and tasks with limited annotations, we propose VRNet, a novel variation-robust point cloud based few-shot network in Section IV-B. In detail, Section IV-C introduces an orientation-tolerant feature extractor, while Section IV-D present a multi-scale label propagation mechanism for further performance enhancement.

B. Framework

The overall framework of VRNet for robotic manipulation is illustrated in Fig. 2. Given K support point cloud objects S and a query object Q, we first extract point-wise support features F_S and query features F_Q using a weight-shared orientation-tolerant feature extractor, respectively. Then, according to the affordance label annotations of all objects in S, the foreground and background point features are collected from F_S. Foreground and background prototypes are separately obtained by computing the average foreground point features and background point features from all K objects. The larger K is, the more accurate and representative the prototypes from multiple objects become, potentially leading to better segmentation results. Subsequently,

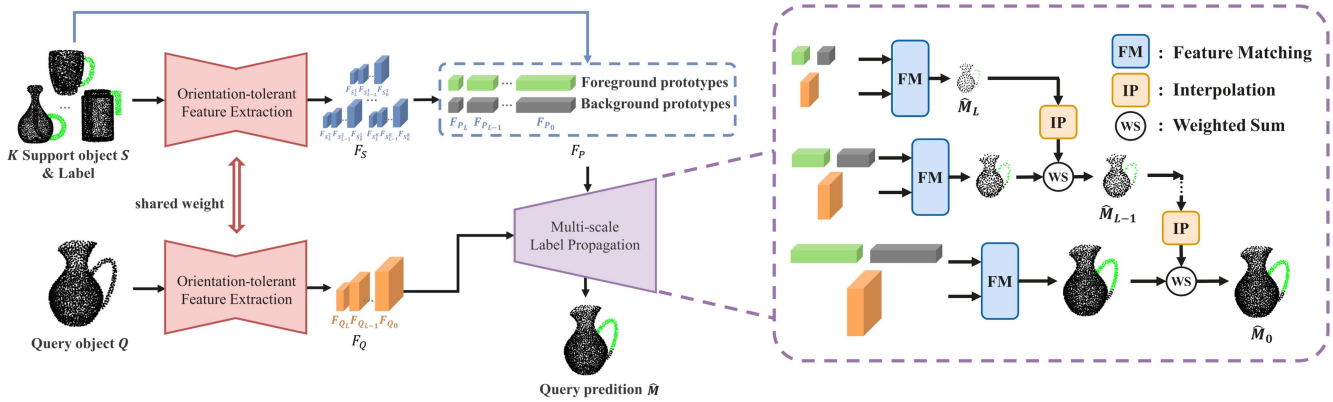

Fig. 2. The pipeline of our proposed VRNet for robotic manipulation. The framework takes two inputs: K support object point clouds with affordance labels and a query object. To extract orientation-agnostic characteristics, orientation-tolerant feature extraction is performed on the objects using a shared extractor. Then, guided by the affordance labels, foreground and background prototypes are computed from multi-level support features. Finally, query prediction is generated by a multi-scale label propagation module, which performs feature matching between prototypes and query features and gradually advances labels from coarse-grained to fine-grained. .

the support prototypes and F_Q are fed into a multi-scale label propagation module, which conducts multi-level feature matching between support and query objects and propagates affordance labels to query objects for the final prediction.

C. Orientation-Tolerant Feature Extractor

Poses of different objects are always variant in practice. Despite normalization, the absolute coordinates of points within point cloud objects invariably change after rotation and flipping, serving as the initial features for network input. However, this variation is scarcely captured by orientation-sensitive networks. In the few-shot framework, the variation between support and query point clouds can easily lead to incorrect feature matching.

Unlike rotational equivariant networks [21], [22], [23], which maintain a consistent relationship between input rotations and output feature transformations, orientation-agnostic characteristics of related distances and angles are introduced from the source, which are more conducive to the subsequent feature matching in the few-shot settings. As essential features of point clouds, the relative distances and angles among points remain consistent before and after rotation and flipping. In light of their modeling principles, we consider integrating these two relative features into our feature extractor PointNet++ [32].

Similar to 2D convolutional networks such as SegNet [25] and PSPNet [26], the encoder of PointNet++ extracts features by progressively downsampling points, grouping neighbor points, and using convolutional layers and max-pooling layers to aggregate features of neighbor points. During the grouping process, for every point in the downsampled point cloud, termed a center point, a constant number of neighboring points is selected for the center point based on the distances between its absolute coordinates and those of other points. Subsequently, we extract relative features with these selected points, as depicted in Fig. 3. Specifically, a mean point is calculated by averaging the coordinates of all neighbor points. Every neighbor point p_1, the mean point p_2, and center point p_3 form a triangle structure, where the edge lengths and angles remain consistent regardless

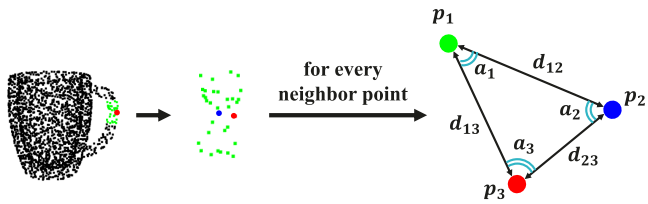

Fig. 3. Orientation-agnostic characteristics including relative distances and angles in the feature extractor. The neighbor points of a red center point are marked in green, while the mean point of neighbor points is marked in blue.

of changes in the point cloud's orientation. Therefore, we are able to construct orientation-agnostic features:

$$F_{agn} = (d_{12}, d_{13}, d_{23}, \cos\angle a_1, \cos\angle a_2, \cos\angle a_3)\,, \qquad (2)$$

where d_{ij} represents the relative distance between pi and pj, and a_i represents the angle formed by the two edges connected to vertex p_i. Then, F_{agn} and the features of neighbor points indexed from features extracted by shallower layers are concatenated for feature aggregation.

D. Multi-Scale Label Propagation

The label propagation module takes support prototypes and query features as inputs to produce the predicted affordance labels for the query point cloud objects. However, current methods [19], [27], [28] propagate labels by directly matching the point features of query objects with prototype features of support objects in a fine-grained manner. This inadequate utilization of global information poses considerable challenges, which lead to inaccurate label propagation, especially when the target query objects are partial point clouds. We observe that it is essential to perform global alignment between objects to address the inconsistency in completeness. Thus, we design a multi-scale label propagation mechanism, utilizing coarse-grained information to guide fine-grained pixel-wise predictions.

In the encoder of the feature extractor, the points of each object are progressively downsampled using the farthest point

sampling algorithm, and point features are encoded at every level accordingly. This process represents the entire object with fewer and fewer points, transitioning from the original N_0 to N_1, N_2, and finally N_L. Then, the decoder gradually decodes point features from N_L to N_0 for point-wise embedding. We perform multi-scale label propagation using these multi-scale features of support and query objects, starting from the last scale L.

For the l-th scale label propagation, the N_l features of the support object at the l-th scale are used to generate support prototypes F_{P_l}. Then we predict point labels of the query object at this scale:

$$\hat{M}_l = f_{fm}(F_{P_l}, F_{Q_l}) + w * f_{ip}\left(\hat{M}_{l+1}\right). \tag{3}$$

$f_{fm}(*)$ performs feature matching by calculating pair-wise similarities between F_{P_l} and query features F_{Q_l}, and outputs a plain query label prediction at the current scale. $f_{ip}(*)$ is an interpolation function that extends the prediction of N_{l+1} points at the $l+1$ scale to N_l points at the current scale.

$$\hat{M}_l^i = \frac{1}{\mathcal{T}} \sum_{p_j \epsilon Neigh_{\mathcal{T}}(p_i)} \hat{M}_{l+1}^j, \tag{4}$$

where $\mathcal{T}$ denotes the number of selected nearest neighbor points. By weighting the predictions at the current scale and $l+1$ scale with a weighting factor w, the final result $\hat{M}_l$ is obtained and will be used for label propagation at the $l-1$ scale, until labels are predicted for all the original N_0 points in the query object. Compared with only performing label propagation at the initial scale, the multi-scale mechanism provides a more comprehensive prediction result.

E. Loss Function

We use the cross-entropy loss function as the training objective of VRNet. The affordance prediction loss L_{ce} is calculated as

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\Big(\mathbb{I}\left[M^i = 1\right]\log \hat{M}^i + \mathbb{I}[M^i = 0]\log\left(1-\hat{M}^i\right)\Big) \tag{5}$$

where N is the number of points in query point clouds, M and $\hat{M}$ represent the ground-truth labels and predicted affordance confidence, respectively. We train the model in an end-to-end manner by minimizing L_{ce}.

V. Experiments

A. Datasets

As aforementioned in Section IV, we treat 3D affordance estimation as a point-wise binary classification task. Thus, ShapeNetPart [29] and 3DAffordanceNet [3] are suitable to be adopted for our model training and evaluation. Additionally, we construct a new grasp affordance dataset termed AcronymAffordance based on Acronym [30]. Its diverse categories enable a better representation of the advantage of the few-shot algorithm in handling novel objects. The detailed implementation is depicted in Fig. 4. For every object in Acronym, we uniformly

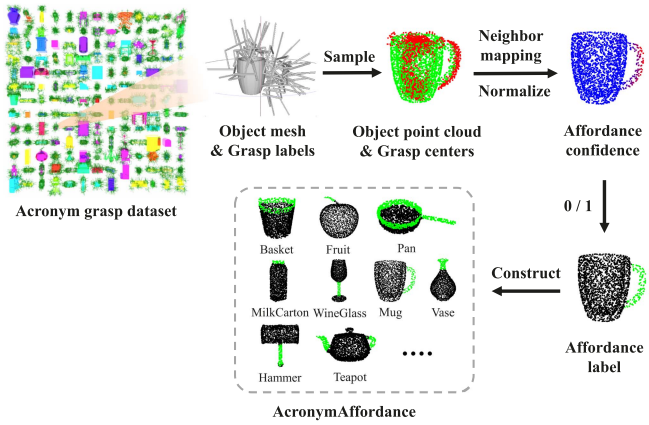

Fig. 4. A dataset named AcronymAffordance is constructed based on Acronym. We convert object meshes and grasp candidate labels in Acronym to object point clouds and grasp affordance labels.

sample 2048 points from the object mesh and calculate grasp centers based on the provided grasp candidate labels. Then a Gaussian weight broadcast mapping is performed from each grasp center to its 32 neighbor points according to their relative distance. Here, every object point may receive weights from zero to several grasp centers. Subsequently, to calculate the affordance confidence, we aggregate the weights of every point and normalize them to a range from 0 to 1 across all object points. And the final affordance label is obtained by thresholding the affordance confidence of every point, which is empirically set to 0.1. Generally, we construct the AcronymAffordance dataset through converting grasp candidate labels to object affordance labels. To summarize, the details of the above three datasets are listed below:

1) *ShapeNetPart* consists of 31693 point clouds categorized into 16 common object classes from ShapeNet. Each point cloud object is annotated with 2-5 part labels. We consider one part of each object as the affordance region.
2) *3DAffordanceNet* is derived from the PartNet dataset, containing 22949 objects belonging to 23 shape classes. Each point of the objects is annotated with 18 affordance labels for different robotic tasks, including *grasp*, *press*, *push*, *pull*, and others.
3) *AcronymAffordance* provides 5006 object point clouds from 88 shape classes and their grasp affordance labels, after filtering out object classes with very few objects and aggregating highly similar classes in Acronym.

Partial dataset generation: We are also concerned with estimating affordance on partial point clouds, as in practical applications, objects are often partially captured by a single depth camera. Thus, we perform partial dataset generation correspondingly. Except for 3DAffordanceNet, which has its partial version, we adopt the hidden point removal algorithm [31] to generate partial point clouds for ShapeNetPart and AcronymAffordance, respectively.

Few-shot dataset construction: Support-query object pairs are required for network input in few-shot settings. We construct few-shot datasets for the above three datasets. Taking training and evaluating a few-shot affordance model for various object classes on the 3DAffordanceNet dataset as an example, we

TABLE I
COMPARISON WITH THE FULLY-SUPERVISED METHOD AND OTHER STATE-OF-THE-ART FEW-SHOT METHODS IN THE FULL-TO-FULL SETTINGS. USING MIOU (%) AS THE EVALUATION METRIC. '†': THE BENCHMARK METHOD, DISTINCT FROM THE 3D AFFORDANCENET DATASET

Methods \ Datasets		ShapeNetPart		3DAffordanceNet@class		3DAffordanceNet@task		AcronymAffordance	
Fully-supervised	3DAffordanceNet† [3]	66.28		54.96		54.02		52.16	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Few-shot	FT	42.91	44.46	35.88	40.10	32.02	34.82	36.75	40.56
	SCL [17]	44.34	45.80	40.32	41.39	39.38	42.25	46.83	49.51
	AttMPTI [18]	46.59	49.45	40.86	40.95	41.73	42.44	48.38	50.07
	BAM [19]	47.39	49.66	41.48	43.58	41.19	43.89	48.99	52.75
	QGPNet [20]	49.33	53.78	43.34	44.03	42.20	43.38	49.91	53.22
	VRNet	**55.12**	**57.93**	**46.47**	**47.83**	**44.85**	**47.64**	**52.88**	**57.13**

The bold values represent the best results among various few-shot methods under different experimental settings.

divide all object classes in the dataset into two equal parts according to alphabetical order, one for training and the other for evaluation. For every sample in an episode training batch, a class is first randomly selected from the training classes. Then, $K+1$ object point clouds belonging to this class are randomly chosen to form a K-support and 1-query pair for the sample. During evaluation, 200 samples are randomly selected from each evaluation class to construct the evaluation set. In particular, since the 3DAffordanceNet dataset contains both various object classes and multiple manipulation affordance labels, we construct few-shot datasets via class split and manipulation task split, termed 3DAffordanceNet@class and 3DAffordanceNet@task, respectively.

Few-shot settings: Note that in our few-shot experiments, support and query point cloud objects are randomly rotated to emulate the varying orientations of objects in real-world environments. We establish three types of few-shot configurations based on the shape type of support and query objects: a) full-to-full, where both support and query objects are full shapes; b) full-to-partial, where support objects are full shapes while query objects are partial shapes; c) partial-to-partial, where both support and query objects are partial shapes. The ratio of affordance points in partial support samples is more than 5%.

B. Evaluation Metrics

Since the few-shot 3D affordance estimation task predicts the affordance of each point in a point cloud object, mean Intersection-over-Union (mIoU) is used as the evaluation metric for all the experiments:

$$mIoU = \frac{1}{C}\sum_{i=1}^{C}\frac{n_{ii}}{\sum_{j=1}^{C} n_{ij} + \sum_{j=1}^{C} n_{ji} - n_{ii}} \tag{6}$$

where $C=2$ denotes the number of classes, comprising the foreground class representing the affordance and the background class representing the absence of affordance. n_{ij} represents the number of points that belong to class i and are predicted as class j by the estimation model. For the entire evaluation set, we compute the average mIoU across all evaluation classes.

C. Implementation Details

Compared with its simplified version PointNet [32], PointNet++ [24] has the advantage of embedding local neighborhood information for every 3D point. Thus, we adopt PointNet++ as the backbone to build a feature extractor for few-shot 3D affordance estimation. The point number is sampled from 2048 to 512, 128, and 1 for local-to-global multi-level feature encoding, and then gradually decoded to the origin number to produce robust point-wise features. In the initial training stage, we pretrain the backbone on base classes in a fully-supervised manner by following the feature extractor with 3 convolution layers, which produce the affordance confidence for all base classes. In the subsequent stage, VRNet is fine-tuned based on the meta-learning paradigm with an Adam optimizer and a learning rate of 0.001. The weighting factor w is set to 0.5. All experiments are conducted on NVIDIA GeForce RTX 3090.

D. Performance Evaluation

We establish experiments to compare our method with start-of-the-arts, including the fully-supervised method 3DAffordanceNet [3], as well as few-shot methods like SCL [17], AttMPTI [18], BAM [19], and QGPNet [20].

Table I illustrates quantitative results of different methods under the full-to-full setting. The fully-supervised method uses all labeled objects for training. "1-shot" and "5-shot" represent that 1 labeled and 5 labeled support examples are provided, respectively. As shown in the table, fine-tuning (FT) affordance models for target classes and tasks with only several labeled examples, compared to fully-supervised models using all objects, still leaves a significant performance gap across all three datasets. Meanwhile, most few-shot methods achieve better performance due to their learning-to-learn mechanism, which renders affordance models agnostic to training classes and tasks. Moreover, VRNet consistently achieves superior performance across all evaluated datasets.

Besides, we visualize the affordance prediction results of different methods, as depicted in Fig. 5. The first column shows different query objects marked with groundtruth affordance regions. For instance, the *grasp* affordance region of a headphone is on the headband (row 1), the *contain* affordance region of a bag is at the bottom (row 4), and the *openable* affordance region of a bottle is associated with its cap (row 5). As demonstrated, AttMPTI and QGPNet misclassify certain parts of objects as affordance regions, whereas our method yields more accurate segmentation results.

We also conduct experiments when target query objects are partial shapes in the few-shot affordance segmentation. The full-to-partial and partial-to-partial results are summarized in Tables II and III. Although the prediction performance is affected by the partial shape of query objects, VRNet outperforms other

TABLE II
COMPARISON WITH THE FULLY-SUPERVISED METHOD AND OTHER STATE-OF-THE-ART FEW-SHOT METHODS IN THE FULL-TO-PARTIAL SETTINGS. USING MIOU (%) AS THE EVALUATION METRIC. '†': THE BENCHMARK METHOD, DISTINCT FROM THE 3D AFFORDANCENET DATASET

Methods \ Datasets		ShapeNetPart		3DAffordanceNet@class		3DAffordanceNet@task		AcronymAffordance	
Fully-supervised	3DAffordanceNet† [3]	61.80		50.82		48.87		50.84	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Few-shot	FT	37.69	42.55	33.67	35.95	30.45	33.73	38.54	40.48
	SCL [17]	43.47	43.91	36.92	41.47	36.27	38.33	39.74	44.84
	AttMPTI [18]	45.19	49.75	37.21	42.75	38.69	39.48	40.54	45.73
	BAM [19]	46.04	48.60	36.83	40.38	38.02	39.28	39.58	45.28
	QGPNet [20]	46.37	51.27	39.47	43.85	37.74	40.83	43.64	48.92
	VRNet	**53.41**	**54.59**	**43.04**	**45.66**	**40.38**	**41.84**	**48.29**	**53.74**

The bold values represent the best results among various few-shot methods under different experimental settings.

TABLE III
COMPARISON WITH THE FULLY-SUPERVISED METHOD AND OTHER STATE-OF-THE-ART FEW-SHOT METHODS IN THE PARTIAL-TO-PARTIAL SETTINGS. USING MIOU (%) AS THE EVALUATION METRIC. '†': THE BENCHMARK METHOD, DISTINCT FROM THE 3D AFFORDANCENET DATASET

Methods \ Datasets		ShapeNetPart		3DAffordanceNet@class		3DAffordanceNet@task		AcronymAffordance	
Fully-supervised	3DAffordanceNet† [3]	62.03		50.91		48.88		50.96	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Few-shot	FT	33.83	39.48	28.63	32.26	27.63	30.27	32.67	35.55
	SCL [17]	37.16	40.83	31.84	36.72	30.74	34.23	38.28	41.61
	AttMPTI [18]	40.02	44.29	34.38	37.48	30.18	35.98	40.17	42.02
	BAM [19]	42.92	47.38	33.91	37.31	33.19	36.31	40.83	42.25
	QGPNet [20]	45.35	49.25	35.17	40.85	32.92	35.22	40.47	43.87
	VRNet	**49.34**	**51.94**	**38.31**	**42.02**	**36.43**	**38.14**	**43.16**	**46.30**

The bold values represent the best results among various few-shot methods under different experimental settings.

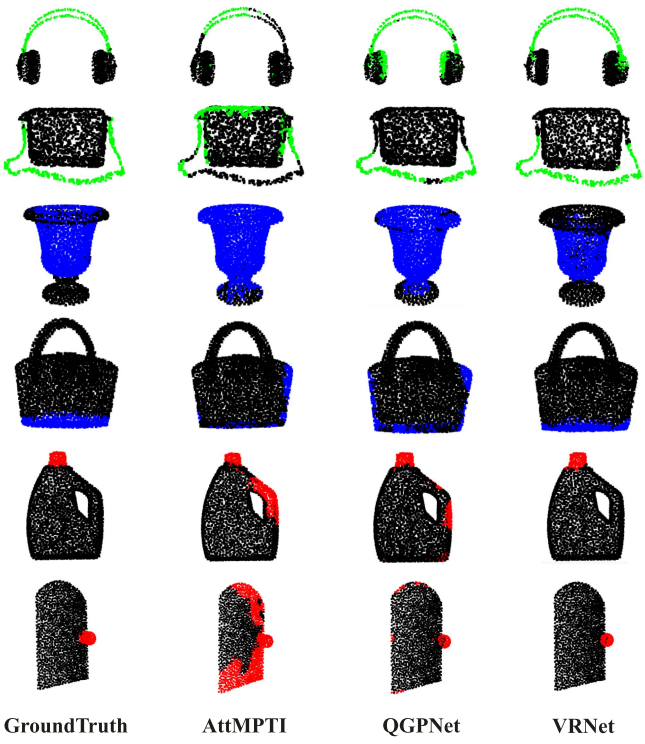

Fig. 5. Visualization analysis of query prediction results using different methods under the 1-shot and full-to-full setting. *Grasp*, *contain*, and *openable* manipulation affordances are included as examples.

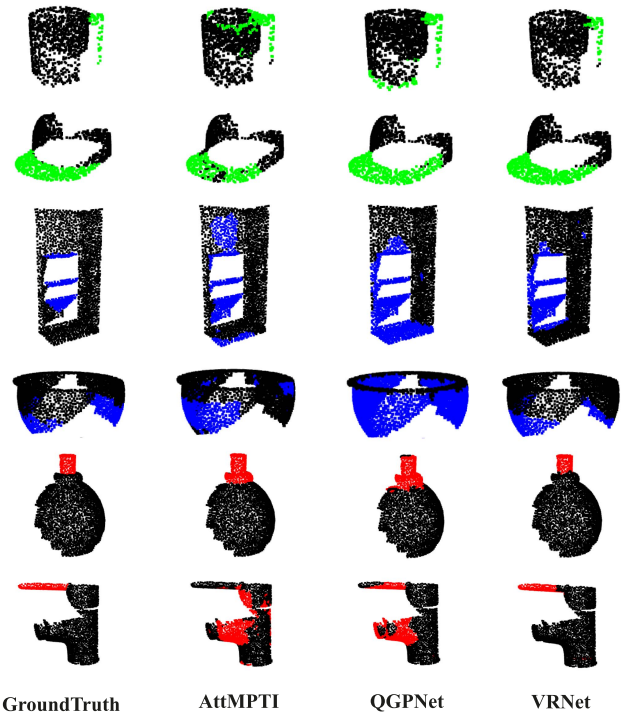

Fig. 6. Visualization analysis of query prediction results using different methods under the 1-shot and full-to-partial setting. *Grasp*, *contain*, and *openable* manipulation affordances are included as examples.

state-of-the-art few-shot methods. We visualize prediction results under the full-to-partial setting in Fig. 6. It can be found that our method also suppresses other methods across different object classes and manipulation tasks, which verify the effectiveness of the proposed approach.

E. Ablation Studies

We first conduct an ablation study to objectively analyze the role of each module in VRNet. The experiment results are presented in Table IV. The lowest mIoU is seen in the case where both orientation-tolerant characteristics and multi-scale mechanism are removed. Taking the full-to-partial as an example,

TABLE IV
ABLATION ANALYSIS OF DIFFERENT MODULES UNDER THE 1-SHOT SETTING ON SHAPENETPART. 'OTFE': ORIENTATION-TOLERANT FEATURE EXTRACTOR. 'MSLP': MULTI-SCALE LABEL PROPAGATION

OTFE	MSLP	Full-to-full	Full-to-partial	Partial-to-partial
		44.13	41.85	39.52
√		52.46	50.95	45.11
	√	48.07	48.54	46.72
√	√	**55.12**	**53.41**	**49.34**

The bold values represent the best results among various combinations of modules under different few-shot configurations.

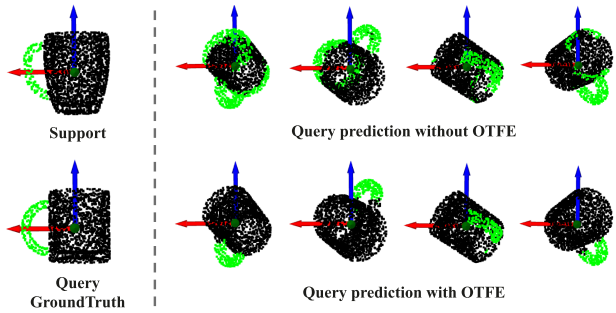

Fig. 7. Query prediction results of objects at different angles using the few-shot model without/with the orientation-tolerant feature extractor (OTFE).

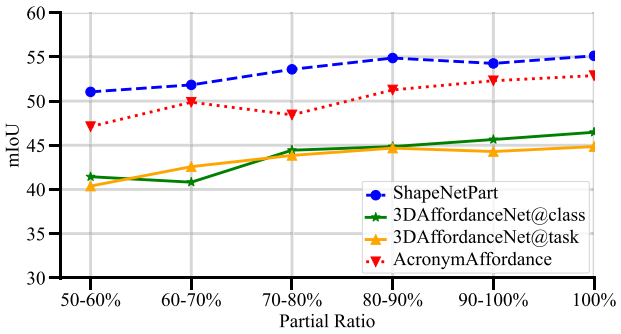

Fig. 8. The influence of the partial ratio of objects on affordance region prediction in different datasets.

applying only the orientation-tolerant feature extractor increases the segmentation mIoU by 9.1%. Fig. 7 shows that query predictions remain consistent for objects at different angles after using the module. Multi-scale label propagation helps improve the mIoU performance to 53.41%.

In addition, we analyze the segmentation performance of affordance models on objects with various partial ratios in the full-to-partial and 1-shot setting. We omit those objects with partial ratios less than 50% because their corresponding affordance areas are likely to be severely lost. Fig. 8 concludes that higher partial ratios generally lead to performance gains, which is consistent with our expectations. The number of neighbor points, denoted as $\mathcal{T}$, is a crucial factor affecting the effectiveness of multi-scale label propagation module in VRNet. As shown in Fig. 9, regardless of the type of few-shot configuration, a small number of neighbor points always bias prediction results. At the same time, label propagation based on too many neighbor points blurs the guiding role of label predictions in the former scales.

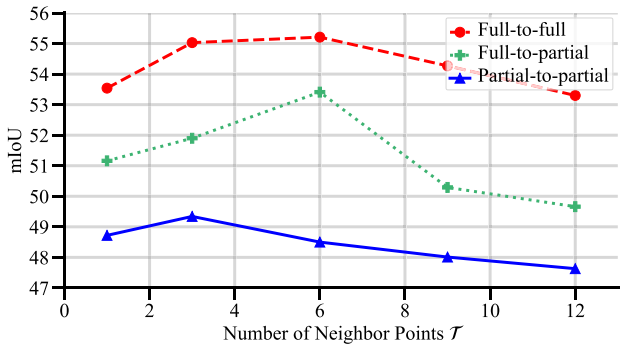

Fig. 9. The influence of the number of neighbor points $\mathcal{T}$ on affordance region prediction in two types of configurations.

TABLE V
THE DETAILS OF REAL ROBOTIC EXPERIMENTAL DEMONSTRATIONS

Demonstrations	Objects	Affordances
Hang an earphone on a stand	Earphone, Stand	*Grasp*, *Support*
Pour water into a bowl	Bottle, Bowl	*Grasp*, *Pour*, *Contain*
Store a cup in a drawer	Cup, Drawer	*Grasp*, *Pull*, *Push*
Get a mug from a cabinet	Mug, Cabinet	*Grasp*, *Openable*
Place a cup into a box	Cup, Box	*Grasp*, *Contain*

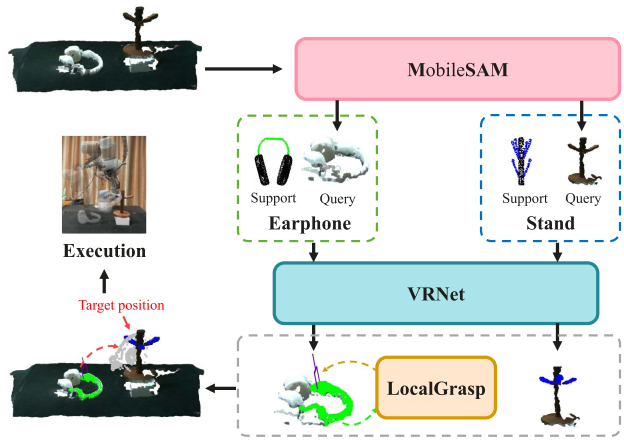

Fig. 10. The pipeline of few-shot real robotic manipulation, where VRNet is the key component that provides affordance region prediction for real partial object point clouds.

F. Real World Experiments

VRNet can predict affordance regions for novel class objects in manipulation tasks, which determine the manipulation positions and assist robotic execution. In order to showcase the functional role of affordance regions better in real robotic manipulation, we deploy several representative robotic manipulation scenarios, as demonstrated in Table V. These typical demonstrations involve both gripper-object and object-object interactions, which prove our framework's generalization ability in real-world situations.

An example of real robotic manipulation involves hanging an earphone on a stand, as illustrated in Fig. 10. First, a 3D point cloud scene is captured using a Realsense-D435i camera from a single viewpoint. To identify target objects, we employ MobileSAM [33], which locates object regions by clicking several points. Then, for each target real object, affordance

segmentation is performed by VRNet. The query sample is the real partial object. The support sample is a simulated object point cloud of the same class randomly selected from the above datasets. Since LocalGrasp [34] performs grasp detection on local regions, for object manipulation requiring gripper-object interactions, we generate the most suitable grasp candidate on the affordance part. At last, motion planning is used to generate a complete trajectory, and the robot executes the manipulation with a Robotiq 2-finger parallel-jaw gripper.

Note that query objects are not seen during the few-shot model training, that is to say, they are novel class objects. In real-world scenarios, our VRNet performs well and facilitates successful manipulation execution. Please refer to the supplementary materials for more details.

G. Limitations

The limitations of our method is summarized as follows: 1) The incomplete or biased affordance information from support samples of partial shapes can negatively affect affordance segmentation performance of VRNet on query objects; 2) MobileSAM's failure to segment an entire target object from real scenes impacts the success of robotic execution.

VI. Conclusion

In this letter, we have proposed a meta-learning based few-shot affordance segmentation network VRNet for 3D point cloud objects in robotic manipulation tasks. To address the variations in target query objects that are likely to occur and negatively affect the segmentation of the segmentation model, we have designed an orientation-tolerant feature extractor to extract similar features for the same object in different poses, and we have devised a multi-scale label propagation mechanism to handle incomplete object representation. Evaluations on simulation datasets and real-world scenarios demonstrate the practicality of the method.

References

[1] J. Cui and J. Trinkle, "Toward next-generation learned robot manipulation," *Sci. Robot.*, vol. 6, no. 54, 2021, Art. no. eabd9461.

[2] Y. Cui, Z. Xu, L. Zhong, P. Xu, Y. Shen, and Q. Tang, "A task-adaptive deep reinforcement learning framework for dual-arm robot manipulation," *IEEE Trans. Automat. Sci. Eng.*, early access, Jan. 18, 2024, doi: 10.1109/TASE.2024.3352584.

[3] S. Deng, X. Xu, C. Wu, K. Chen, and K. Jia, "3D affordancenet: A benchmark for visual object affordance understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 1778–1787.

[4] R. Xu, F.-J. Chu, C. Tang, W. Liu, and P. A. Vela, "An affordance keypoint detection network for robot manipulation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2870–2877, Apr. 2021.

[5] W. Chen, H. Liang, Z. Chen, F. Sun, and J. Zhang, "Learning 6-dof task-oriented grasp detection via implicit estimation and visual affordance," in *Proc. 2022 IEEE Int. Conf. Intell. Robots Syst.*, 2022, pp. 762–769.

[6] T. Hospedales, A. Antoniou, P. Micaelli, and A. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5149–5169, Sep. 2022.

[7] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.

[8] V. G. Satorras and J. B. Estrach, "Few-shot learning with graph neural networks," in *Proc. 6th Int. Conf. Learn. Representations*, 2018, vol. 3, no. 4, pp. 1–13.

[9] H. Kim, Y. Ohmura, and Y. Kuniyoshi, "Using human gaze to improve robustness against irrelevant objects in robot manipulation tasks," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4415–4422, Jul. 2020.

[10] S. Abondance, C. B. Teeple, and R. J. Wood, "A dexterous soft robotic hand for delicate in-hand manipulation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5502–5509, Oct. 2020.

[11] Y. Qin, A. Escande, F. Kanehiro, and E. Yoshida, "Dual-arm mobile manipulation planning of a long deformable object in industrial installation," *IEEE Robot. Automat. Lett.*, vol. 8, no. 5, pp. 3039–3046, May 2023.

[12] B. Moldovan, P. Moreno, M. V. Otterlo, J. Santos-Victor, and L. De Raedt, "Learning relational affordance models for robots in multi-object manipulation tasks," in *Proc. 2012 IEEE Int. Conf. Robot. Automat.*, IEEE, 2012, pp. 4373–4378.

[13] F.-J. Chu, R. Xu, and P. A. Vela, "Learning affordance segmentation for real-world robotic manipulation via synthetic images," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1140–1147, Apr. 2019.

[14] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, " KPAM: Keypoint affordances for category-level robotic manipulation," in *Proc. The Int. Symp. Robot. Res.*, 2019, pp. 132–157.

[15] C. Xu, Y. Chen, H. Wang, S.-C. Zhu, Y. Zhu, and S. Huang, "Partafford: Part-level affordance discovery from 3D objects," 2022, *arXiv:2202.13519*.

[16] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. 28th Brit. Mach. Vis. Conf.*, 2017, pp. 167.1–167.13

[17] B. Zhang, J. Xiao, and T. Qin, "Self-guided and cross-guided learning for few-shot segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8312–8321.

[18] N. Zhao, T.-S. Chua, and G. H. Lee, "Few-shot 3 d point cloud semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8873–8882.

[19] C. Lang, G. Cheng, B. Tu, and J. Han, "Learning what not to segment: A new perspective on few-shot segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 8057–8067.

[20] D. Hu, S. Chen, H. Yang, and G. Wang, "Query-guided support prototypes for few-shot 3D indoor segmentation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 6, pp. 4202–4213, Jun. 2024.

[21] C. Deng, O. Litany, Y. Duan, A. Poulenard, A. Tagliasacchi, and L. J. Guibas, "Vector neurons: A general framework for so(3)-equivariant networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 12200–12209.

[22] F. Fuchs, D. Worrall, V. Fischer, and M. Welling, "SE(3)-Transformers: 3D roto-translation equivariant attention networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1970–1981.

[23] Y. Chen, C. Tie, R. Wu, and H. Dong, "Eqvafford: SE(3) equivariance for point-level affordance learning," 2024, *arXiv:2408.01953*.

[24] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet : Deep hierarchical feature learning on point sets in a metric space," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5105–5114.

[25] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[26] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.

[27] Z. Wu, X. Shi, G. Lin, and J. Cai, "Learning meta-class memory for few-shot semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 517–526.

[28] Q. Fan, W. Pei, Y.-W. Tai, and C.-K. Tang, "Self-support few-shot semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 701–719.

[29] A. X. Chang et al., "Shapenet: An information-rich 3 d model repository," 2015, *arXiv:1512.03012*.

[30] C. Eppner, A. Mousavian, and D. Fox, "Acronym: A large-scale grasp dataset based on simulation," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6222–6227.

[31] S. Katz, A. Tal, and R. Basri, "Direct visibility of point sets," in *Proc. ACM SIGGRAPH*, 2007, pp. 24–es.

[32] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3 d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[33] C. Zhang et al., "Faster segment anything: Towards lightweight sam for mobile applications," 2023, *arXiv:2306.14289*, 2023.

[34] W. Tang, S. Chen, P. Xie, D. Hu, W. Yang, and G. Wang, "Rethinking 6-DoF grasp detection: A flexible framework for high-quality grasping," 2024, *arXiv:2403.15054*.