SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes

Alexandros Delitzas[1] Ayça Takmaz[1] Robert Sumner[1] Federico Tombari[2]
Marc Pollefeys[1,3] Francis Engelmann[1,2]
[1] ETH Zurich [2] Google [3] Microsoft

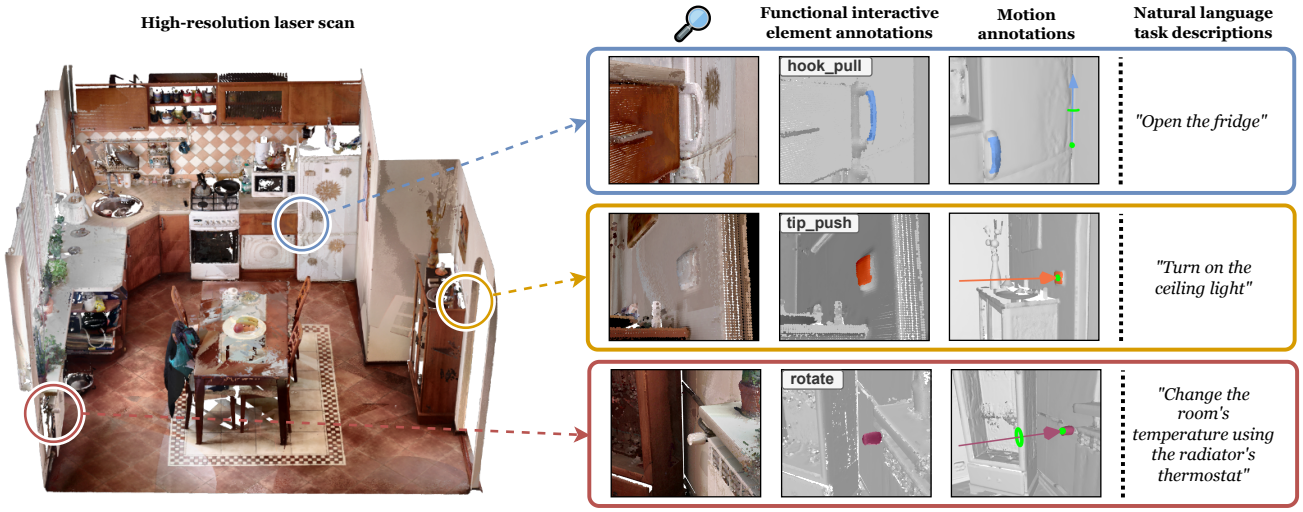

Figure 1. **The SceneFun3D dataset.** We introduce the first large-scale dataset with highly accurate interaction annotations in 3D real-world indoor environments. SceneFun3D contains more than 14.8k annotations of functional interactive elements in 710 high-resolution 3D scenes accompanied by nine affordance categories. Additionally, it provides motion annotations that describe how to interact with the functional elements and natural language descriptions of tasks that involve manipulating them in the scene context.

Abstract

Existing 3D scene understanding methods are heavily focused on 3D semantic and instance segmentation. However, identifying objects and their parts only constitutes an intermediate step towards a more fine-grained goal, which is effectively interacting with the functional interactive elements (e.g., *handles, knobs, buttons) in the scene to accomplish diverse tasks. To this end, we introduce SceneFun3D, a large-scale dataset with more than 14.8k highly accurate interaction annotations for 710 high-resolution real-world 3D indoor scenes. We accompany the annotations with motion parameter information, describing how to interact with these elements, and a diverse set of natural language descriptions of tasks that involve manipulating them in the scene context. To showcase the value of our dataset, we introduce three novel tasks, namely functionality segmentation, task-driven affordance grounding and 3D motion estimation, and adapt existing state-of-the-art methods to tackle them. Our experiments show that solving these tasks in real 3D scenes remains challenging despite recent progress in closed-set and open-set 3D scene understanding methods.*

1. Introduction

Datasets of 3D indoor environments have been extensively used for computer vision, robotics, embodied AI and mixed reality. To perceive 3D environments, 3D object instance segmentation has served as a fundamental task to provide the appropriate knowledge to agents about the objects in the scene and subsequently enable the interaction with them. Going a step further, some works have studied the task of part-object segmentation focusing on the lower-level object parts, *e.g.*, drawers of a cabinet. However, these two tasks serve only as a proxy since in the real-world setting, agents need to successfully detect and interact with the *functional interactive elements* (e.g., knobs, handles, buttons)

of the objects in the scene. Detecting these elements is an under-explored area, mainly due to the fact that most existing datasets, which are based on commodity RGB-D reconstructions, often fail to accurately capture the 3D geometry of small details in the scene.

To successfully interact with the functional elements in the scene, agents should be capable of understanding visual affordances. The concept of *affordance* was first defined by Gibson [20], as those actions or behaviors afforded due to the physical structure and design of an object (*e.g.*, a button affords pressing, a drawer knob affords pulling). To encourage research in this direction, prior works [10, 63] create 3D datasets which include dense label annotations on the object parts of 3D CAD models in the PartNet dataset. While these datasets are very helpful for identifying affordances on the object-level, there is no dataset providing geometrically fine-grained annotations of visual affordances in real-world 3D scenes, to the best of our knowledge.

Although the Gibsonian notion [20] of affordance might be sufficient in object-level, it lacks to inform about the purpose or the specific function of an interactive element in the context of a scene. 3D environments are characterized by complex inter- and intra-object functional relationships. For instance, if an agent is instructed to turn on the ceiling light, knowing that the buttons of the scene can be pressed (Gibsonian affordance) does not offer enough information about what will happen when the button is pressed. To address this limitation, Pustejovsky [47] introduced the notion of *telic affordance* which is defined as the action or behavior conventionalized due to an object's typical use or purpose [24]. For example, while the Gibsonian affordance of a light switch button is that it can be pressed, its telic affordance is turning on the ceiling light. Interestingly, we see that recent open-vocabulary models [33, 46, 54] display promising results towards understanding telic affodances of functionalities in 3D scenes by leveraging the knowledge of foundation models, such as CLIP [50] and OpenSeg [19]. However, there is no benchmark to assess and compare their visual affordance understanding capability.

In this work, we build the first large-scale dataset, namely SceneFun3D (Fig. 1), containing more than 14.8k high-fidelity annotations of functional interactive elements in scenes along with nine Gibsonian-inspired affordances. These are complemented by accurate motion parameters, outlining how to manipulate these elements and diverse natural language descriptions of tasks that involve interacting with them. With the introduction of this dataset we hope to encourage future research on the following questions.

Where are the functionalities located in 3D indoor environments and what actions they afford? We construct a dataset of 710 scenes captured with a Faro laser scanner by leveraging the ARKitScenes data assets [4]. This provides us with high-resolution 3D geometry, compared to previous datasets that comprise commodity-level RGB-D reconstructions, which is essential to capture small interactive elements in the scene (Fig. 2). We develop a lightweight web-based interface that enables the fine-grained annotation of high-resolution point clouds. We utilize it to collect annotations of the functional interactive elements of the scenes accompanied by a Gibsonian-inspired affordance category (*e.g.*, rotate, hook_pull, tip_push).

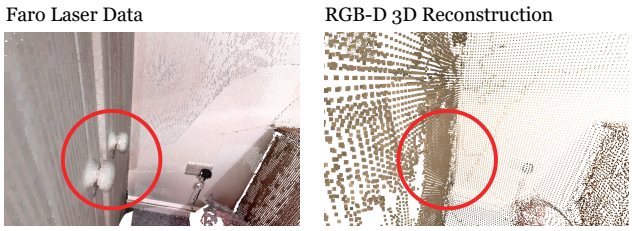

Figure 2. **Details in laser scans compared to commodity-level RGB-D reconstructions.** Laser scans capture a higher level of detail, which is required for the geometrically fine-grained annotation of small interactive elements. In datasets with commodity-level RGB-D reconstructions (e.g, ScanNet, MultiScan, Matterport) these details are not visible.

What purpose do the functionalities serve in the scene context? We argue that telic affordances are crucial to understand the purpose of interactive elements in the scene and propose a natural approach to study them. We show functionality annotations to human annotators and ask them to provide free-form language descriptions of tasks that involve interacting with the displayed functionality (*e.g.*, the task description "turn on the ceiling light" involves interacting with the light switch). To the best of our knowledge, we are the first to link Gibsonian and telic affordances for enhanced 3D scene understanding.

Which motions are involved to interact with the functional elements? To further investigate how an agent can interact with the functional elements in the scene, we collect 3D motion annotations. For example, pressing a button involves a motion perpendicular to the button's surface to be performed, while opening a cabinet's door involves a rotational motion around its hinge by pulling its handle. In contrast to previous works [36, 40] that focus on the motion of articulated parts for a limited set of object categories, we study motions from the actor's perspective with an interaction-centric approach which better resembles the real-world setting.

We introduce three challenging tasks and we leverage our SceneFun3D dataset for systematic benchmarking. We propose techniques to adapt state-of-the-art methods on closed-set and open-set 3D scene understanding to tackle the proposed tasks and perform extensive experiments.

2. Related Work

Semantic understanding of indoor 3D scenes. Existing 3D indoor datasets [4, 6, 9, 51, 66], largely focus on identifying scene semantics and object instances. Earlier datasets such as ScanNet [9] and Matterport3D [6] rely on commodity-level RGB-D reconstruction, whereas more recent datasets such as ARKitScenes[4] and ScanNet++ [66] provide laser scans, resulting in much precise reconstructions capturing even small geometrical details, as illustrated in Fig. 2. Benefiting from these large-scale annotated datasets, several 3D semantic segmentation methods [1, 3, 8, 13, 14, 27–29, 35, 37, 38, 48, 49, 56, 57, 60, 64] and 3D instance segmentation methods [12, 15, 22, 25, 32, 34, 52, 55, 58, 59, 65] have been developed. While these 3D segmentation models trained on 3D indoor datasets work well as a proxy for identifying objects from certain categories with which an agent can interact, they fall short on providing information about the functional elements necessary for interactions. In this work, we focus on identifying functional elements and how to interact with them. We build upon the 3D laser scans from the large-scale ARKitScenes [4] dataset, which captures much higher detail, providing a well-suited medium for us to explore functional elements.

Affordance understanding. Understanding scene affordances has been a long-standing goal in vision and robotics. Prediction of affordances has been first explored through the lens of rule-based approaches [61]. Then, several learning-based methods have addressed the prediction of functionalities from images and videos [11, 17, 39, 45], and in 3D [10, 43, 44, 62, 63]. Another line of work targets language grounding in 3D scenes [7, 67], and 3D scene understanding guided by open-vocabulary queries [33, 46, 54]. Existing methods are largely limited to point-level or object-level predictions. Functional element annotations in our dataset enable the extension of object-level open-vocabulary approaches such as OpenMask3D [54] to identify fine-grained functional-elements based on complex affordance descriptions. Our dataset focuses specifically on interactive functional elements, and provides a benchmark consisting of a rich set of natural language task descriptions.

3D motion estimation. A line of work [26, 31, 36, 40, 53] explores the estimation of 3D motion and mobility of interactable elements. MultiScan [40] focuses on scenes with articulated objects, and estimates object part mobility. OPD [31] and OPDMulti[53] address openable part detection and motion parameter estimation. Hsu *et al*. [26] explore the inference of articulation properties of scene objects. These datasets focus on articulated objects and address a limited set of interaction categories. In our work, we instead study 3D motion estimation from an interaction-centric perspective, addressing a larger variety of interaction cases.

3. Task definitions

We address three novel 3D scene understanding tasks:

Task 1: Functionality segmentation. Given an input point cloud $\mathcal{P} = \{(p_i, f_i)\}$, where $p_i \in \mathbb{R}^3$ are the point coordinates and f_i are the additional point features, such as RGB color and normals, the task is to predict the instance masks $\{m_i\}_{i=1}^K$ of the functional interactive elements of the scene as well as the associated affordance label $\{\ell_i\}_{i=1}^K$ for each instance, where K is the number of instances in the scene. We define C Gibsonian-inspired affordance categories to describe interactions afforded by common functionalities in indoor scenes (*e.g.*, "rotate"). We highlight that this task is conceptually different from traditional 3D instance segmentation. The model needs to understand visual affordances in an object-class-agnostic manner and infer the action that a functional element affords from the 3D geometry. We consider functional interactive elements as the object components in the scene that humans and agents interact with to perform specific actions (*e.g.*, turning a handle to open a door, rotating a dial to control the temperature).

Task 2: Task-driven affordance grounding. Given an input point cloud $\mathcal{P}$ and a free-form task description $\mathcal{D}$ (*e.g.*, "open the door", "turn on the ceiling light"), the goal is to predict the instance mask m of the functional interactive element referred to by the task description as well as the associated affordance label ℓ. To tackle this task, models need to display understanding of the telic affordance, i.e., the purpose, of the functionalities in the context of the scene.

Task 3: Motion estimation. Given an input point cloud $\mathcal{P}$, this task complements Task 1 and aims to identify the motion parameters $\{\phi_i\}_{i=1}^K$, which describe the action that the agent should perform, to interact with the predicted functionality. Following the same notation as [31, 40, 53], we represent the motion parameters as $\phi_i = \{t_i, a_i, o_i\}$, where $t_i = \{\textit{rotation}, \textit{translation}\}$ is the motion type, $a_i \in \mathbb{R}^3$ is the motion axis direction and $o_i \in \mathbb{R}^3$ is the motion origin. To address this task, the model should be able to understand how the functionality works and what motion is required to interact with the corresponding functional element.

4. Building the SceneFun3D dataset

In this section, we describe our approach towards building the SceneFun3D dataset.

4.1. Laser scans

To study the functional interactable elements in 3D scenes, high-resolution point clouds are necessary so that the details and the 3D geometry are of high quality (Fig. 2). To this end, we construct a dataset consisting of high-quality 3D indoor scenes by leveraging the data assets provided by ARKitScenes [4] as follows.

ARKitScenes provides multiple laser scans per scene (four on average) by placing a Faro Focus S70 laser scanner in different positions in the scene. We use the provided laser scanner's poses for each scene and combine the laser scans under the same coordinate system to increase the scene coverage. Afterwards, we downsample the combined laser scan with a voxel size of $5mm$, which is sufficient to preserve the details of the interactive functional elements of the scene (*e.g.*, small buttons, knobs, handles, etc.), while enabling processing by machine learning models.

Next, we visually verify the output XYZRGB point cloud. We exclude scenes where the laser scanner's poses are incorrect, small scenes without interaction spots as well as scenes for which high-resolution RGB frames are not available. After following this selection process, we construct a dataset of 710 scenes. We highlight that our pipeline is scalable, presenting the potential to expand the number of scenes by leveraging high-resolution datasets released concurrently with our work, such as ScanNet++ [66].

4.2. RGB images and camera poses

We accompany the scans from our dataset with posed RGB images and video. ARKitScenes provides on average three video sequences for each scene recorded with a 2020 iPad Pro. These video sequences come with RGB images from the Wide and Ultra Wide cameras, depth maps from the on-device LiDAR scanner, ARKit camera trajectory as well as an ARKit mesh reconstruction of the scene based on the low-resolution frames. However, the aforementioned iPad data and the Faro laser scans are expressed in a different coordinate system and the transformations are not provided by ARKitScenes. To enable scene understanding with multiple sensor data, we register the laser scans in the coordinate system of the RGB images as described in Sec. 4.3.

Furthermore, the camera poses in the ARKit camera trajectory are not synced with the iPad's RGB frames. To help the registration process as well as utilization in downstream tasks, we extract and provide accurate camera poses for each frame by performing rigid body motion interpolation in $\mathbb{SO}(3) \times \mathbb{R}^3$ [21].

4.3. Registration and alignment

We perform a series of steps to register the laser scans in the coordinate system of the camera poses. First, we reconstruct a high-resolution point cloud using the high-resolution RGB-D frames and the interpolated camera poses. This high-resolution point cloud is used as a proxy for registering the laser scan in the coordinate system of the camera poses. To help the registration process, we remove extraneous points from the laser scan due to transparent surfaces, such as windows, by using the DBSCAN clustering algorithm [16]. Consequently, we align the laser scan to the proxy point cloud using Predator [30] and then refine the alignment by performing Multi-Scale Iterative Closest Point (ICP). As a final step, we visually inspect the alignment by projecting the color of the RGB frames to the laser scan. In rare cases when the registration result is not successful, we use manual correspondences for initialization.

label	description
rotate	functionalities that are adjusted by a rotary switch knob, *e.g.* thermostat
key_press	surfaces that consist of keys that can be pressed, *e.g.* remote control, keyboard
tip_push	functionalities that can be triggered by the tip of the finger, *e.g.* light switch
hook_pull	surfaces that can be pulled by hooking up fingers, *e.g.* fridge handle
pinch_pull	surfaces that can be pulled through a pinch movement, *e.g.* drawer knob
hook_turn	surfaces that can be turned by hooking up fingers, *e.g.* door handle
foot_push	surfaces that can be pushed by foot, *e.g.* foot pedal of a trash can
plug_in	surfaces that comprise electrical power sources
unplug	removing a plug from a socket

Table 1. Affordance label descriptions.

4.4. Semantic annotation and data collection

For the data collection process, we have created a lightweight web-based tool to facilitate the fine-grained annotation on large and dense point clouds. Our tool presents three main advantages compared to existing open-source tools which were used for annotating existing datasets [4, 6, 9, 40, 66]. First, previous works annotate decimated meshes after performing over-segmentation [18] which reduces the annotation accuracy. Instead, we directly annotate the high-resolution point cloud and allow an annotation accuracy of up to a single 3D point. Second, our tool enables the annotation of high-resolution laser scans with minimum hardware requirements (no GPU required). To do this, we utilize an accelerated ray-casting algorithm based on Bounding Volume Hierarchies (BVH) [41]. More specifically, we group the 3D points into bounding volumes in a recursive fashion which speeds up the spatial queries significantly during the annotator's clicks. Lastly, during annotation, annotators can see videos of the scene. This not only helps annotators to identify the scene functionalities and affordances more accurately and faster but also provides further information which might not be clearly visible in the 3D point cloud.

Functionalities. We use our annotation UI to collect annotations of the functional interactive elements in the scenes which include an instance mask as well as an affordance label. We compile a list of nine Gibsonian-inspired affor-

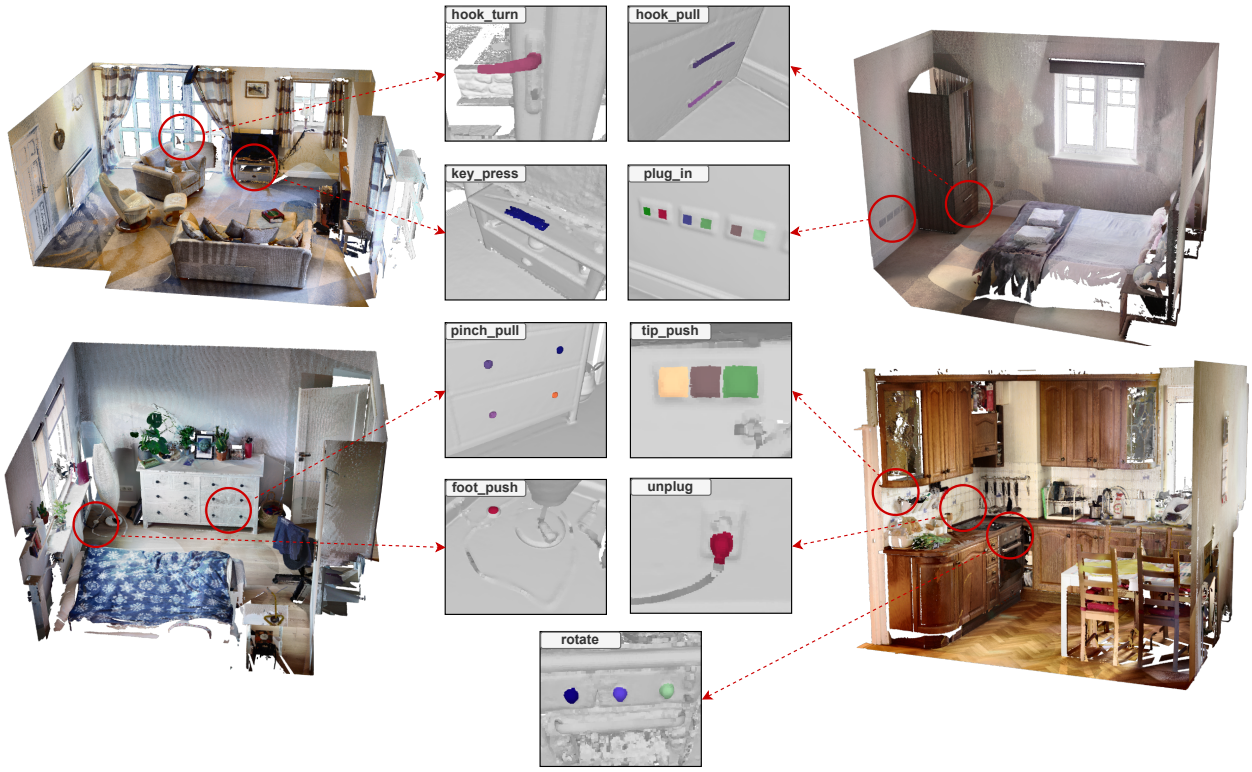

Figure 3. Examples of functional interactive element annotations.

dance labels, drawing inspiration from prior research [10, 23, 39], to represent the interaction with common functional interactable elements of in indoor environments. A short description of each label can be found in Tab. 1. Annotators are tasked with detecting functionalities in the scene, selecting the affordance label that describes the interaction with the functionality and then annotate the instance mask. To facilitate the annotation process, they are allowed to freely navigate in the 3D scene using our UI's controls as well as watch the video sequences of the scene.

Additionally, we annotate functionalities whose geometry or the parent object's geometry is not well-represented in the laser scans. This may occur in cases where the functional part (*e.g.* a knob, handle, etc.) or the parent object (*e.g.* a fridge) is built of a reflective material. We categorize these samples under the label "exclude" and we exclude these cases from evaluation in our experiments.

Natural language task descriptions. To study the telic affordance or purpose of the collected functionalities in the scene context, we collect natural language descriptions of tasks that involve interacting with the corresponding functionalities. First, functionalities are displayed to annotators in the 3D scene. We ask them to provide natural language descriptions for tasks that uniquely involve the displayed functionality annotation. For example, if the displayed functionality is a light switch under the affordance category "tip_push", then the associated task description is "Turn on the ceiling light". We omit collecting descriptions for functionalities whose purpose is not clear in the context of the scene (*e.g.* buttons on an unknown electronic device).

Inspired by [67], we augment our collected language descriptions by rephrasing them to increase diversity. We utilize the ChatGPT model *gpt-3.5-turbo-instruct* for sentence rephrasing. During the verification phase, we ensure the rephrased task descriptions are well-written and correctly correspond to the functionalities in the scene context.

3D Motions. We collect the motions needed to interact with the annotated functionalities as follows. Initially, functionality annotations are displayed to annotators in the 3D scene. By observing the high-quality 3D point cloud as well as the associated scene videos, human annotators can easily infer the motion required to interact with the element. For each functional interactive element, the annotators select the motion type (translational or rotational), the motion axis origin by selecting a point in the scene as well as the motion axis direction by setting the direction of the 3D vector using our UI's helper tools.

5. The SceneFun3D Dataset

In this section, we describe the SceneFun3D dataset and we present statistics concerning the scenes, functionality annotations, collected language task descriptions and motion an-

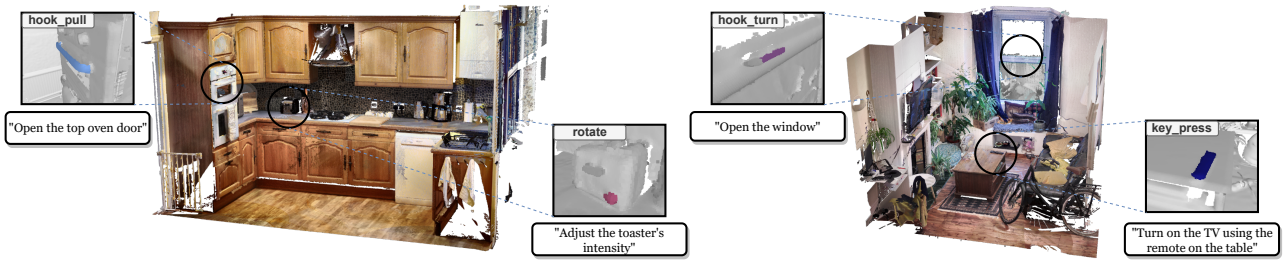

Figure 4. Examples of the collected natural language task descriptions.

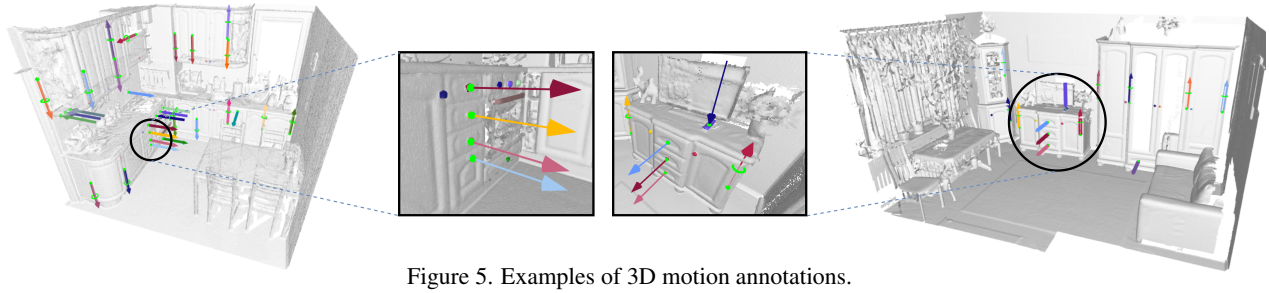

Figure 5. Examples of 3D motion annotations.

notations. Figures 3, 4 and 5 show examples of functionality annotations, collected task descriptions and motion annotations in our dataset respectively.

Train/Validation/Test splits. Following the standard practice, we split our data into training, validation and test splits. Since the test set of ARKitScenes is not publicly available, we use the scenes in its validation set as our test set. To construct the validation set, we randomly draw scenes from the training set of ARKitScenes and use the rest as the training split. Overall, our dataset consists of 545, 80 and 85 scenes for training, validation and testing respectively.

Dataset statistics. Our dataset offers the total of 14,867 annotations of functional interactive elements along with their affordance class for 710 scenes. Furthermore, we provide motion annotations for 14,279 interactive elements out of which 8325 require translational motion and 6542 rotational motion. Last, we offer natural language task descriptions for 10,913 interactive elements. After the automated rephrasing augmentation process, we receive 6,220 additional descriptions, which results in the total of 17,133. We refer the reader to the supplementary material for additional statistics.

6. Baselines and Experiments

We leverage the SceneFun3D dataset to introduce benchmarks for the novel tasks of functionality segmentation, task-driven affordance grounding and 3D motion estimation. For each task, we first describe the baselines and evaluation metrics and then we show quantitative and qualitative results on our test set. For further implementation details, we refer the reader to the supplementary material.

6.1. Functionality segmentation

For this task, we adapt two state-of-the-art methods for 3D object instance segmentation, Mask3D [52] and SoftGroup [58]. We also report results on the open-vocabulary LERF model [33].

Mask3D-F. Since instance masks of the functional interactive elements are smaller in size than the masks of object instances, state-of-the-art methods on object instance segmentation do not work well out-of-the-shelf. As a first step, we substitute the distribution-based BCE loss in the loss function with a region-based Dice loss which can handle better the background/foreground class imbalance. We train Mask3D with the overall loss $\mathcal{L}_{\text{seg}} = \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{ce}}\mathcal{L}_{\text{ce}}$, where $\mathcal{L}_{\text{dice}}$ is the dice loss to supervise the masks and $\mathcal{L}_{\text{ce}}$ is a multi-label cross entropy loss.

SoftGroup-F. Following [58], we first train the U-Net backbone on the semantic masks. Similar to Mask3D-F, we substitute the cross-entropy loss with a weighted multi-class dice loss $\mathcal{L}_{\text{m-dice}}$. We train the backbone using the combined loss $\mathcal{L}_{\text{backbone}} = \lambda_{\text{m-dice}}\mathcal{L}_{\text{m-dice}} + \lambda_{\text{offset}}\mathcal{L}_{\text{offset}}$, where $\mathcal{L}_{\text{offset}}$ is the offset loss [58] used to supervise the offset vectors. Next, we freeze the backbone and train the top-down refinement module on the instance masks. For this stage, we use the combined loss $\mathcal{L} = \mathcal{L}_{\text{backbone}} + \mathcal{L}_{\text{top-down}}$. We utilize the loss $\mathcal{L}_{\text{top-down}} = \lambda_{\text{ce}}\mathcal{L}_{\text{ce}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} + \lambda_{\text{score}}\mathcal{L}_{\text{score}}$, where $\mathcal{L}_{\text{ce}}$ is a multi-label cross-entropy loss, $\mathcal{L}_{\text{dice}}$ is the dice loss and $\mathcal{L}_{\text{score}}$ is the mask score loss [58].

LERF. LERF [33] is a method for grounding language embeddings from CLIP [50] into NeRF [42], enabling open-ended language queries in 3D. We evaluate the zero-shot capabilities of LERF on our dataset.

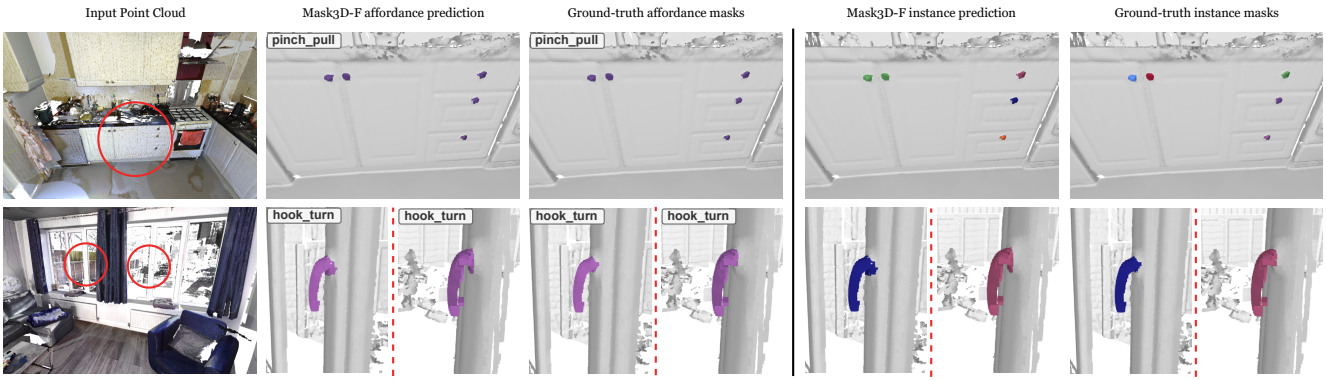

Figure 6. Qualitative results on the Mask3D-F predictions for the task of functionality segmentation.

Method	AP	AP_{50}	AP_{25}
SoftGroup-F	3.6	8.4	17.2
LERF [33]	4.8	12.3	18.1
Mask3D-F	**7.9**	**18.3**	**26.6**

Table 2. Quantitative results on functionality segmentation.

Method	AP_{50}	AP_{25}
OpenMask3D [54]	0.0	0.0
LERF [33]	4.9	11.3
OpenMask3D-F	**8.0**	**17.5**

Table 3. Quantitative results on task-driven affordance grounding.

Coarse-to-fine learning based on a curriculum. For training Mask3D-F and SoftGroup-F, we propose a curriculum learning technique to boost the performance. Since these methods were originally designed to work with larger instance masks of objects, they may struggle with detecting the smaller masks of interactive elements in the scene. Inspired by the concept of curriculum learning [5], we start training with coarse instance masks at first, which are easier for the network to detect. Then, during training, we gradually start feeding the network with more fine-grained masks closer to the ground-truth. To generate the coarser masks, we expand the ground-truth instance masks by considering all the points within a certain radius from the mask's points. Specifically, denoting the point cloud as $\mathcal{P}$, the ground-truth instance mask as $\mathcal{Q}$, and the mask expansion radius as r_n, the expanded instance mask is calculated as follows

$$\mathcal{Q}^{r_n}_{\text{expand}} = \{p \mid \|p-q\|_2 < r_n,\ p \in \mathcal{P}, q \in \mathcal{Q}\} \quad (1)$$

The mask expansion radius is gradually reduced during training using step decay

$$r_n = r_0 d^{\lfloor \frac{n}{\alpha} \rfloor} \quad (2)$$

where r_n is the mask expansion radius in epoch n, r_0 is the initial expansion radius at the beginning of training, d is the decay rate and α is the decay interval. After r_n becomes smaller than a threshold r_{thr}, we disable mask expansion and we refine the network using the ground-truth masks.

Metrics. As our main metrics, we report the mean Average Precision at the IoU thresholds of 0.25 and 0.50, AP_{25} and AP_{50} respectively. We also report AP, the average over different IoU thresholds from 0.5 to 0.95 with a step of 0.05.

Results. We report the performance on Tab. 2. We observe that Mask3D-F achieves better performance than SoftGroup-F on functionality segmentation, which is in accordance with previous findings on object instance segmentation benchmarks such as ScanNet [9] and S3DIS [2]. LERF achieves better scores than SoftGroup-F but fails to match the performance of Mask3D-F. Furthermore, all methods are effective at segmenting distinctive elements such as handles that are easily observable but struggle with very small structures, such as knobs on an electrical device. In Fig. 6, we show qualitative results on the Mask3D-F semantic and instance predictions.

We also perform an ablation study on the effect of the initial mask expansion radius (Tab. 4, left). We observe that setting the mask expansion radius to $r_0 = 0.1$ leads to optimal performance and increasing it further does not yield any performance gains. Our results demonstrate that if we disable coarse-to-fine training ($r_0 = 0$) the model fails to detect interactive elements in the scene.

r_0	AP_{50}	AP_{25}
0.2	18.3	26.2
0.1	**18.3**	**26.6**
0.05	9.8	18.6
None	0.0	0.0

k_{exp}	AP_{50}	AP_{25}
0.1	4.5	11.2
0.5	8.3	16.2
1.0	**8.0**	**17.5**
2.0	8.0	16.5

Table 4. **Ablation studies.** Left: Initial mask expansion radius (r_0) used for coarse-to-fine learning on Mask3D-F for the task of functionality segmentation. Right: Expansion ratio (k_{exp}) parameter of OpenMask3D-F for the task of task-driven affordance grounding.

6.2. Task-driven affordance grounding

For this task, we adapt OpenMask3D [54] to perform language-guided segmentation of functional elements,

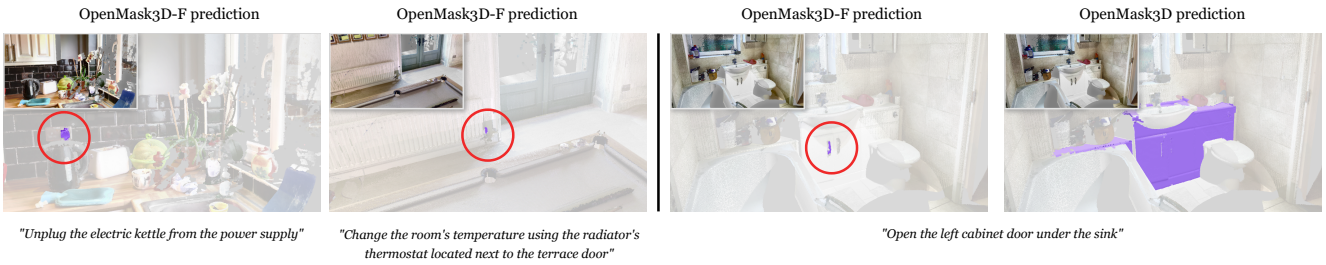

Figure 7. Qualitative results on the OpenMask3D-F predictions for the task of task-driven affordance grounding. OpenMask3D-F uses functional element-level masks, which enables more fine-grained segmentation compared to object-level approaches such as OpenMask3D [54].

Method	AP_{25}	+M	+MA	+MAO
Mask3D-FM (rgb)	**26.6**	23.8	9.8	7.9
Mask3D-FM (rgb + n)	26.5	**24.0**	**10.2**	**8.1**

Table 5. Quantitative results on motion estimation

driven by complex descriptions. We also use the LERF [33] model as a baseline.

OpenMask3D-F. OpenMask3D [54] is an instance based approach, which relies on object mask proposals obtained from Mask3D [52]. This object-level representation does not allow the segmentation of functional elements such as buttons, handles and switches. We extend OpenMask3D to use mask proposals from our adapted Mask3D-F which proposes masks for functional elements, and we refer to this approach as OpenMask3D-F. For this task, we compute a CLIP-based [50] embedding for each proposed functional element-mask. Then we encode the task-description queries for each scene using the CLIP text encoder. We measure the similarity between the mask-embeddings and query-embeddings, and retrieve the mask for a given description text, if the similarity score is above a certain threshold. As OpenMask3D relies on multi-scale image crops, it is sensitive to the crop-expansion ratio, k_{exp} (details in [54]). We also experiment with varying k_{exp} values to investigate its affect on task-driven affordance grounding. This ablation study is presented in Tab. 4 (right).

Metrics. For this task, we use instance segmentation metrics, and report the AP_{50} and AP_{25}.

Results. Scores are presented in Tab. 3. We observe that OpenMask3D-F outperforms LERF by a significant margin. The results on OpenMask3D-F also highlight the importance of having fine-grained functional element masks in order to successfully identify how a certain task can take place. Fig. 7 shows qualitative results of OpenMask3D-F.

6.3. Motion estimation

Mask3D-FM. We extend the Mask3D-F baseline to additionally predict the per-instance motion parameters along with the segmentation mask and affordance label. To this end, we enhance the mask module [52] of the architecture to jointly predict the motion type, motion axis and motion origin by adding a per-instance prediction head for each motion parameter. For training, we utilize the overall loss of $\mathcal{L} = \mathcal{L}_{\text{seg}} + \mathcal{L}_{\text{motion}}$, where $\mathcal{L}_{\text{motion}}$ is a combined loss to supervise the motion parameters, inspired by [31, 53]. Specifically, it is defined as $\mathcal{L}_{\text{motion}} = \lambda_{\text{type}}\mathcal{L}_{\text{type}} + \lambda_{\text{axis}}\mathcal{L}_{\text{axis}} + \lambda_{\text{origin}}\mathcal{L}_{\text{origin}}$, where $\mathcal{L}_{\text{motion}}$ is a cross-entropy loss for the motion type, $\mathcal{L}_{\text{axis}}$ is a smooth L1 loss for the motion axis and $\mathcal{L}_{\text{origin}}$ is a smooth L1 loss for the motion origin.

Metrics. To evaluate the motion prediction performance, we follow [40, 53] and extend the AP_{25} metric for motion parameter accuracy. More specifically, we further constrain mask prediction by whether the model accurately predicted the motion type (**+M**), the motion type and the motion axis direction (**+MA**) and the motion type, motion axis direction, and motion origin (**+MAO**). We consider the motion axis matched if the angle between the ground-truth axis direction and the predicted axis does not exceed 15° and the motion origin matched if the minimum distance between the axis is lower than 0.25.

Results. Quantitative results can be seen on Tab. 5. We report two variants of our baseline, one that uses only the rgb color information of the point cloud and one that additionally uses the estimated normal information. We observe that the normal information helps the model to predict the motion parameters more accurately.

7. Conclusion

In this work, we present SceneFun3D, the first large-scale dataset that leverages laser scans to provide geometrically fine-grained masks along with affordance labels of functional interactive elements in 3D real-world indoor scenes, followed by motion parameter information and a diverse set of natural language descriptions of tasks that require interaction with them. To investigate multi-task and holistic 3D scene understanding, we introduce the three novel tasks of functionality segmentation, task-driven affordance grounding and 3D motion estimation. We adapt state-of-the-art methods on closed-set and open-set 3D scene understanding and report promising results. We believe that our dataset will stimulate advancements in embodied AI, robotics and realistic human-scene interaction modelling.

References

[1] Abhishek Anand, Hema Swetha Koppula, Thorsten Joachims, and Ashutosh Saxena. Contextually Guided Semantic Labeling and Search for 3D Point Clouds. In *International Journal on Robotics Research (IJRR)*, 2011. 3

[2] Iro Armeni, Ozan Sener, Amir R. Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3D Semantic Parsing of Large-Scale Indoor Spaces. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7

[3] Matan Atzmon, Haggai Maron, and Yaron Lipman. Point Convolutional Neural Networks by Extension Operators. In *ACM Transactions On Graphics (TOG)*, 2018. 3

[4] Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes - A Diverse Real-World Dataset For 3D Indoor Scene Understanding Using Mobile RGB-D Data. In *Neural Information Processing Systems (NeurIPS)*, 2021. 2, 3, 4

[5] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, 2009. 7

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D Data in Indoor Environments. In *3DV*, 2017. 3, 4

[7] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. ScanRefer: 3D Object Localization in RGB-D Scans using Natural Language. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[8] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 4, 7

[10] 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 Conference on Computer Vision and Pattern Recognition*, 2021. 2, 3, 5

[11] Thanh-Toan Do, Anh Viet Nguyen, Ian D. Reid, Darwin Gordon Caldwell, and Nikos G. Tsagarakis. AffordanceNet: An End-to-End Deep Learning Approach for Object Affordance Detection. In *International Conference on Robotics and Automation (ICRA)*, 2018. 3

[12] Cathrin Elich, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. 3D-BEVIS: Birds-Eye-View Instance Segmentation. In *German Conference on Pattern Recognition (GCPR)*, 2019. 3

[13] Francis Engelmann, Theodora Kontogianni, Alexander Hermans, and Bastian Leibe. Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. In *International Conference on Computer Vision (ICCV) Workshops*, 2017. 3

[14] Francis Engelmann, Theodora Kontogianni, Jonas Schult, and Bastian Leibe. Know What Your Neighbors Do: 3D Semantic Segmentation of Point Clouds. In *European Conference on Computer Vision (ECCV) Workshops*, 2018. 3

[15] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3D-MPA: Multi Proposal Aggregation for 3D Semantic Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*, page 226–231. AAAI Press, 1996. 4

[17] Kuan Fang, Te-Lin Wu, Daniel Yang, Silvio Savarese, and Joseph J. Lim. Demo2Vec: Reasoning Object Affordances From Online Videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[18] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal on Computer Vision (IJCV)*, 59:167–181, 2004. 4

[19] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling Open-Vocabulary Image Segmentation with Image-Level Labels. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[20] James J. Gibson. *The Ecological Approach to Visual Perception: Classic Edition*. Houghton Mifflin, 1979. 2

[21] Adrian Haarbach, Tolga Birdal, and Slobodan Ilic. Survey of Higher Order Rigid Body Motion Interpolation Methods for Keyframe Animation and Continuous-Time Trajectory Estimation. In *2018 International Conference on 3D Vision (3DV)*, pages 381–389, 2018. 4

[22] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. OccuSeg: Occupancy-aware 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[23] Mohammed Hassanin, Salman Khan, and Murat Tahtali. Visual Affordance and Function Understanding: A Survey. *ACM Comput. Surv.*, 54(3), 2021. 5

[24] Alexander Henlein, Anju Gopinath, Nikhil Krishnaswamy, Alexander Mehler, and James Pustejovsky. Grounding human-object interaction to affordance behavior in multimodal datasets. *Frontiers in Artificial Intelligence*, 6, 2023. 2

[25] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[26] Cheng-Chun Hsu, Zhenyu Jiang, and Yuke Zhu. Ditto in the House: Building Articulation Models of Indoor Scenes through Interactive Perception. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 3

[27] Zeyu Hu, Xuyang Bai, Jiaxiang Shang, Runze Zhang, Jiayu Dong, Xin Wang, Guangyuan Sun, Hongbo Fu, and Chiew Lan Tai. VMNet: Voxel-Mesh Network for Geodesic-Aware 3D Semantic Segmentation. In *International Conference on Computer Vision (ICCV)*, 2021. 3

[28] Binh-Son Hua, Minh-Khoi Tran, and Sai-Kit Yeung. Pointwise Convolutional Neural Network. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] Jing Huang and Suya You. Point Cloud Labeling Using 3D Convolutional Neural Network. In *International Conference on Pattern Recognition (ICPR)*, 2016. 3

[30] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, and Konrad Schindler Andreas Wieser. PREDATOR: Registration of 3D Point Clouds with Low Overlap. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[31] Hanxiao Jiang, Yongsen Mao, Manolis Savva, and Angel X Chang. OPD: Single-view 3D openable part detection. In *European Conference on Computer Vision (ECCV)*, 2022. 3, 8

[32] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. PointGroup: Dual-Set Point Grouping for 3D Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[33] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. LERF: Language Embedded Radiance Fields. In *International Conference on Computer Vision (ICCV)*, 2023. 2, 3, 6, 7, 8

[34] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R. Oswald. 3D Instance Segmentation via Multi-task Metric Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[35] Loic Landrieu and Martin Simonovsky. Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[36] Xiaolong Li, He Wang, Li Yi, Leonidas Guibas, A. Lynn Abbott, and Shuran Song. Category-Level Articulated Object Pose Estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[37] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed Points. In *Neural Information Processing Systems (NeurIPS)*, 2018. 3

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully Convolutional Networks for Semantic Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[39] Timo Lüddecke and F. Wörgötter. Learning to Segment Affordances. In *International Conference on Computer Vision (ICCV) Workshops*, 2017. 3, 5

[40] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X Chang, and Manolis Savva. MultiScan: Scalable RGBD scanning for 3D environments with articulated objects. In *Neural Information Processing Systems (NeurIPS)*, 2022. 2, 3, 4, 8

[41] Daniel Meister, Shinji Ogaki, Carsten Benthin, Michael Doyle, Michael Guthe, and Jiri Bittner. A Survey on Bounding Volume Hierarchies for Ray Tracing. *Computer Graphics Forum*, 40:683–712, 2021. 4

[42] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 6

[43] Kaichun Mo, Yuzhe Qin, Fanbo Xiang, Hao Su, and Leonidas Guibas. O2O-Afford: Annotation-free large-scale object-object affordance learning. In *Conference on Robot Learning (CoRL)*, 2021. 3

[44] Tushar Nagarajan and Kristen Grauman. Learning Affordance Landscapes for Interaction Exploration in 3D Environments. In *Neural Information Processing Systems (NeurIPS)*, 2020. 3

[45] Tushar Nagarajan, Christoph Feichtenhofer, and Kristen Grauman. Grounded Human-Object Interaction Hotspots from Video. In *ICCV*, 2019. 3

[46] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D Scene Understanding with Open Vocabularies. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 3

[47] James Pustejovsky. *The generative lexicon*. MIT press, 1998. 2

[48] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[49] Charles R. Qi, Li Yi, Hao Su, and Leonidas J. Guibas. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In *Neural Information Processing Systems (NeurIPS)*, 2017. 3

[50] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *International Conference on Machine Learning (ICML)*, 2021. 2, 6, 8

[51] David Rozenberszki, Or Litany, and Angela Dai. Language-Grounded Indoor 3D Semantic Segmentation in the Wild. In *European Conference on Computer Vision (ECCV)*, 2022. 3

[52] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In *International Conference on Robotics and Automation (ICRA)*, 2023. 3, 6, 8

[53] Xiaohao Sun, Hanxiao Jiang, Manolis Savva, and Angel Xuan Chang. OPDMulti: Openable Part Detection for Multiple Objects. *3DV*, 2024. 3, 8

[54] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3, 7, 8

[55] Ayça Takmaz, Jonas Schult, Irem Kaftan, Mertcan Akçay, Bastian Leibe, Robert Sumner, Francis Engelmann, and Siyu Tang. 3D Segmentation of Humans in Point Clouds with Synthetic Data. In *International Conference on Computer Vision (ICCV)*, 2023. 3

[56] Lyne P. Tchapmi, Christopher B. Choy, Iro Armeni, JunYoung Gwak, and Silvio Savarese. SEGCloud: Semantic Segmentation of 3D Point Clouds. In *International Conference on 3D Vision (3DV)*, 2017. 3

[57] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and Deformable Convolution for Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2019. 3

[58] Thang Vu, Kookhoi Kim, Tung M. Luu, Xuan Thanh Nguyen, and Chang D. Yoo. SoftGroup for 3D Instance Segmentation on 3D Point Clouds. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3, 6

[59] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. SGPN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[60] Silvan Weder, Hermann Blum, Francis Engelmann, and Marc Pollefeys. LabelMaker: Automatic Semantic Label Generation from RGB-D Trajectories. In *International Conference on 3D Vision (3DV)*, 2024. 3

[61] Patrick H. Winston. Learning physical descriptions from functional definitions, examples, and precedents. In *Proceedings of AAAI*, 1983. 3

[62] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-Mart: Learning Visual Action Trajectory Proposals for Manipulating 3D ARTiculated Objects, 2021. 3

[63] Chao Xu, Yixin Chen, He Wang, Song-Chun Zhu, Yixin Zhu, and Siyuan Huang. PartAfford: Part-level Affordance Discovery from 3D Objects. In *ECCVW*, 2022. 2, 3

[64] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters. In *European Conference on Computer Vision (ECCV)*, 2018. 3

[65] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning Object Bounding Boxes for 3D Instance Segmentation on Point Clouds. In *Neural Information Processing Systems (NeurIPS)*, 2019. 3

[66] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. ScanNet++: A High-Fidelity Dataset of 3D Indoor Scenes. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 4

[67] Yiming Zhang, ZeMing Gong, and Angel X Chang. Multi3DRefer: Grounding Text Description to Multiple 3D Objects. In *International Conference on Computer Vision (ICCV)*, 2023. 3, 5