Segment Any 3D-Part in a Scene from a Sentence

Hongyu Wu , **Pengwan Yang**
University of Amsterdam

Abstract

This paper aims to achieve the segmentation of any 3D part in a scene based on natural language descriptions, extending beyond traditional object-level 3D scene understanding and addressing both data and methodological challenges. Due to the expensive 3D acquisition and annotation burden, existing datasets and methods are typically limited to object-level comprehension. To overcome these limitations, we introduce the 3D-PU dataset, the first large-scale 3D scene dataset with dense 3D part annotations, created through a cost-effective method for constructing synthetic 3D scenes with fine-grained part-level annotations, paving the way for advanced 3D-part scene understanding. On the methodological side, we propose OpenPart3D, a system to effectively tackle the challenges of part-level segmentation with three integrated modules that converts a 3D scene into 3D-part segmentation masks. Extensive experiments demonstrate the effectiveness in open-vocabulary 3D scene understanding tasks at part level, with strong generalization across real-world 3D scene datasets.

1 Introduction

Significant progress has been made in open-vocabulary 3D scene understanding, propelled by vision-language models trained on large-scale datasets [Chen *et al.*, 2023; Ding *et al.*, 2023; Ding *et al.*, 2024; Yang *et al.*, 2024; Zeng *et al.*, 2023; Nguyen *et al.*, 2024]. For instance, OpenMask3D [Takmaz *et al.*, 2023] builds on Mask3D [Schult *et al.*, 2023] to generate 3D instance proposals, leveraging the CLIP model [Radford *et al.*, 2021] to align objects with a text query. Similarly, OpenIns3D [Huang *et al.*, 2024] incorporates text comprehension and object detection by adapting a 2D open-world model to 3D scenes. Currently, most open-vocabulary 3D scene understanding efforts are object-focused due to the availability of relevant data. Recently, Search3D [Takmaz *et al.*, 2025] extends beyond object segmentation by constructing 2D-part representations that correspond to 3D objects, relying on well-aligned RGB-D images. In this paper, we aim to advance open-vocabulary 3D scene understanding to the 3D-part level, enabling fine-grained segmentation of individual 3D parts within scenes.

The task of 3D-part understanding introduces new technical challenges and addresses the limitations of existing datasets, which are typically designed for object-level understanding [Dai *et al.*, 2017; Straub *et al.*, 2019; Yeshwanth *et al.*, 2023]. Recent datasets, such as MultiScan [Mao *et al.*, 2022] and SceneFun3D [Delitzas *et al.*, 2024], have started to provide part-level annotations, with MultiScan offering 5K annotations across five categories in 117 scenes, and SceneFun3D delivering 14K annotations with nine affordance labels across 710 scenes. While these are valuable contributions, they are constrained by the limitations of manual labeling, resulting in a relatively coarse granularity and an insufficient number of part annotations for training or fine-tuning robust part-segmentation models. To overcome these limitations, we propose a novel, cost-effective approach for constructing 3D scene data with detailed, fine-grained part-level annotations, bypassing the need for extensive manual scanning and labeling.

We make two key contributions in this paper. First, we introduce the 3D-PU (3D Part Understanding) dataset. We create 3D scenes by combining annotated 3D shapes guided by structured layouts, as illustrated in Figure 1 (left). 3D-PU is a larg-scale 3D dataset with 843,654 dense part annotations across 206 categories for 10,000 scenes, establishing a robust foundation for advancing part-level 3D scene understanding. Second, building upon the part-level annotated scene dataset, we introduce the task of open-vocabulary 3D-part segmentation, enabling the segmentation of any 3D-part within a scene by a text query, as shown in Figure 1 (right). To address this task, we propose OpenPart3D, a system designed to efficiently and accurately segment 3D-parts in complex scenes. OpenPart3D comes with three integrated modules that converts a 3D scene into 3D-part segmentation masks. First, the *Room-Tour Snap Module* generates multiple view images from the 3D scene by positioning cameras at predefined locations with optimized poses, ensuring that small objects and parts are clearly visible in the captured views. Second, these images are then input into a 2D open-vocabulary model to generate 2D-part masks relevant to the text query. The *View-Weighted Voting Module* assigns varying weights to the different views and computes confidence scores for geometrically consistent regions in the 3D point cloud. Finally, the

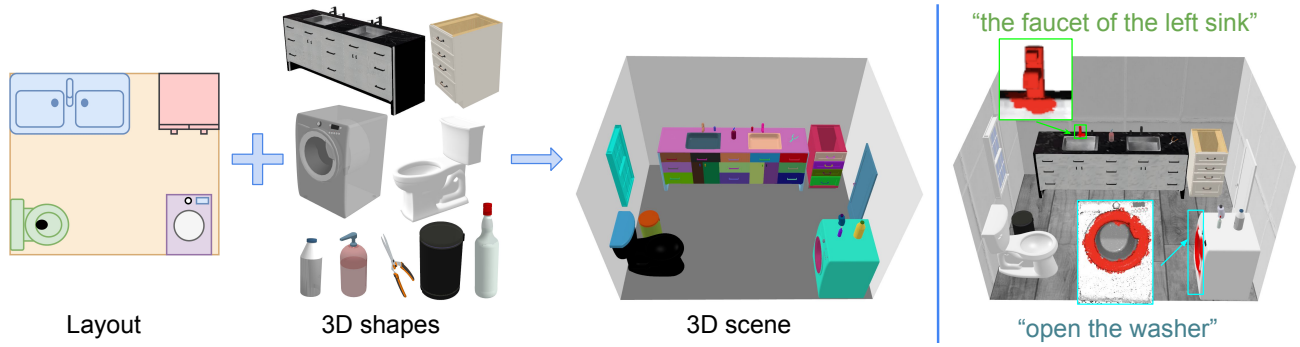

Figure 1: 3D-part understanding dataset and system. We propose the first large-scale 3D scene dataset with dense part annotations, by an automated approach that constructs 3D scenes with detailed part annotations (left). Using this dataset, we introduce the 3D-part understanding task and system that enables flexible part segmentation and identification in a 3D scene based on any query sentence (right).

third *Progressive Merging Module* adaptively merges high-confidence superpoints to produce the final 3D part segmentation masks. Overall, we address both the data limitations and technical challenges inherent in the hard task of open-vocabulary part segmentation in 3D scenes, and establish its effectiveness, even accross real-world 3D scene datasets unseen at training time.

2 Related Work

Open-Vocabulary 3D Scene Understanding. Open-vocabulary understanding of 3D scenes has recently garnered considerable attention [Chen *et al.*, 2023; Ding *et al.*, 2023; Ding *et al.*, 2024; Yang *et al.*, 2024; Zeng *et al.*, 2023; Nguyen *et al.*, 2024; Xue *et al.*, 2025]. OpenScene [Peng *et al.*, 2023] pioneered open-vocabulary 3D scene understanding by leveraging per-pixel image features extracted from posed images of a scene to obtain a task-agnostic, point-wise scene representation. OpenIns3D [Huang *et al.*, 2024] introduced the "Snap and Lookup" strategy, which has proven effective as a 3D object recognition engine, delivering competitive open-vocabulary instance segmentation without relying on well-aligned images. Currently, most existing open-vocabulary 3D scene understanding approaches focus on object-level segmentation. More recently, Search3D [Takmaz *et al.*, 2025] has sought to extend beyond object segmentation by constructing 2D image object-part representations that correspond to 3D objects, thereby enabling hierarchical object-part segmentation. However, a key limitation of Search3D lies in its reliance on well-aligned RGB-D images, which are often unavailable or impractical in real-world applications. Building on the spirit of Search3D, we aim to advance open-vocabulary 3D scene understanding to the 3D part-level, extending the capabilities of object-level understanding and enabling query-based part segmentation directly within 3D scenes. Importantly, we simplify input requirements to 3D data only, enhancing method flexibility and compatibility across diverse settings.

3D Scene Datasets. Existing 3D scene datasets are constrained by their reliance on multiple observations of the same scene and lack part-level information. Datasets like ScanNet [Dai *et al.*, 2017], Replica [Straub *et al.*, 2019], SceneNN [Hua *et al.*, 2016] and ScanNet++ [Yeshwanth *et al.*, 2023] focus on single-room reconstructions, providing dense object annotations on mesh vertices, while ARKitScenes[Baruch *et al.*, 2021]—the largest room-scale 3D dataset—includes 1,661 unique scenes but only offers 3D object bounding box annotations. Matterport3D [Chang *et al.*, 2017], a building-scale dataset, offers manual object-level annotations. Recently, two datasets have provided part-level annotations for 3D scenes. MultiScan [Mao *et al.*, 2022] includes 5,129 part annotations across 5 categories for 117 scenes, and SceneFun3D [Delitzas *et al.*, 2024] delivers 14,867 part-level annotations with 9 affordance labels across 710 scenes. These are valuable contributions, however, as they are limited by manual labeling, the part annotation granularity remains coarse and the relatively small number of part annotations is insufficient for training or fine-tuning robust part-segmentation models. To address these limitations, we propose a novel approach for constructing 3D scene data with extensive fine-grained part-level annotations at a low cost. We introduce the first large-scale 3D scene dataset with detailed 3D-part annotations, named the 3D-PU dataset. The 3D-PU dataset includes 843,654 dense part annotations across 206 categories for 10,000 scenes, creating a robust foundation for advancing 3D part-level scene understanding.

Vision-Language Models for 3D. The recent success of large-scale model training has led to the development of a range of vision-language models (VLMs). Models such as CLIP [Radford *et al.*, 2021] provide a joint embedding space for both image and text, typically taking a single image as input and generating a global image embedding. While effective for tasks like image classification, these representations are limited in their ability to handle tasks requiring spatial localization. SAM [Kirillov *et al.*, 2023] and SAM2 [Ravi *et al.*, 2024] provide the ability to generate class-agnostic masks for any 2D instances, while Mask3D [Schult *et al.*, 2023] extends this capability to 3D scenes, predicting class-agnostic masks for all 3D objects. To address localization-based open-vocabulary detection, a series of methods [Xiao *et al.*, 2024; Liu *et al.*, 2023] have been proposed. Florence2 [Xiao *et al.*, 2024] and GroundingDino [Liu *et al.*, 2023] feature pixel-aligned embeddings, associating each pixel with an embedding vector in the vision-language space, enabling flexible part querying on 2D planes. Leveraging the spatial capa-

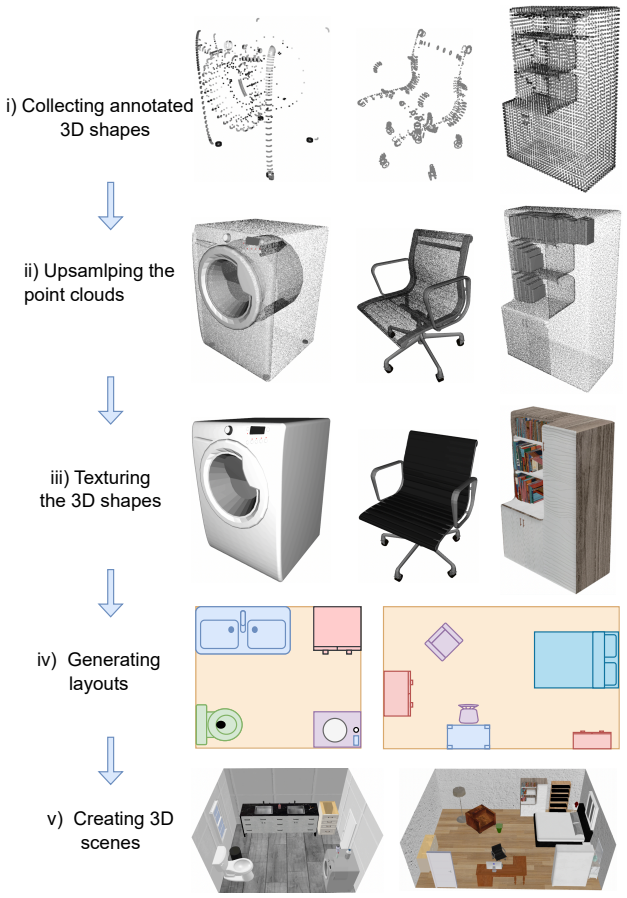

Figure 2: 3D-PU dataset construction consists of five core steps: i) collecting a large volume of 3D shapes with part annotations from PartNet [Mo *et al.*, 2019], SAPIEN [Xiang *et al.*, 2020], and ShapeNetPart [Chang *et al.*, 2015], ii) uniformly upsampling the point clouds from meshes [Yuksel, 2015], iii) generating textures and colors for each 3D shape using both an offline text-guided texturing model [Richardson *et al.*, 2023] and an online texturing service, iv) generating diverse layouts with the assistance of ChatGPT-4, and v) constructing 3D scenes by populating part-annotated objects into the defined scene layouts.

bilities of VLMs, we propose extending these models to the novel task of open-vocabulary 3D part segmentation in scenes, thus minimizing the need for extensive retraining.

3 3D-PU Dataset

A straightforward approach to building a part-annotated 3D scene dataset is to manually label all parts in 3D scenes; however, this is prohibitively costly and impractical. Another common method involves projecting 2D pseudo-part masks from 2D foundation models like SAM2 [Ravi *et al.*, 2024] onto 3D scenes, but this technique falls short of providing precise and comprehensive small-part annotations. In contrast to these methods, we propose constructing 3D scenes with part annotations by leveraging existing 3D shapes that already contain part-level annotations. This approach allows for more accurate and scalable dataset creation. Figure 2 outlines the key steps in the process.

Collecting Annotated 3D Shapes. We gather 3D shapes with part annotations from PartNet [Mo *et al.*, 2019], SAPIEN [Xiang *et al.*, 2020], and ShapeNetPart, a subset of ShapeNet [Chang *et al.*, 2015]. Each shape includes a point cloud, mesh, and fine-grained part annotations. In total, we collected 51,300 shapes across 85 object categories, along with 587,600 part annotations across 206 part classes.

Upsampling the Point Clouds. As shown in Figure 2 a), the original collected shapes typically have denser points on salient parts and sparser points on inner or flat areas. To create uniformly distributed and denser point clouds, we upsample the points of each shape from its mesh [Yuksel, 2015]. Shapes are categorized into large, medium, and small based on their object categories. Large shapes are upsampled to 102,400 points, medium shapes to 51,200 points, and small shapes to 25,600 points. This upsampling results in denser and more complete point clouds.

Texturing the 3D Shapes. To enhance the realism of the 3D scenes, we generate textures and color specifications for each 3D shape. Initially, we use GPT-4 to generate ten natural language prompts, each containing stylized descriptors and color specifications. These ten prompts, along with the corresponding 3D shapes, are processed by both an offline text-guided texturing model [Richardson *et al.*, 2023] and an online texturing service provided by meshy.ai, yielding a total of 20 textured variations. Subsequently, we compute the similarity score between each textured shape and its corresponding text prompt in the language-vision embedding spaces of CLIP2 [Zeng *et al.*, 2023] and Uni3D [Zhou *et al.*, 2023]. Finally, we select the five textured shapes with the highest similarity scores from the pool of 20.

Generating Layouts. We use layouts to guide the structured generation of 3D scenes. With the assistance of ChatGPT-4, we designed 100 distinct layouts representing a wide range of scenes, such as bedrooms, bathrooms, kitchens, living rooms, offices, classrooms, and more. These automatically generated layouts were manually reviewed to correct issues such as object collisions, incorrect orientations, and other inconsistencies. As shown in Figure 2 d), each layout specifies the required large objects and their placement within the scene.

Creating 3D Scenes. Guided by the predefined layouts, we position the specified large 3D objects in their designated locations. Next, we add randomly selected small objects at various positions within the scene to enhance realism and diversity. This process results in natural, realistic 3D scenes with comprehensive part annotations.

Generating Text Queries. In the 3D scenes, each part annotation is assigned an "object_part" label, such as "door_handle" or "table_leg." These "object_part" labels can serve directly as text queries for open-vocabulary part segmentation. Beyond direct text queries, we generate descriptive prompts from "object_part" labels and shape texture cues, with the assistance of ChatGPT-4, to serve as implicit textual queries. For example, "door_handle" yields "open the door", and "table_leg" corresponds to "parts that support the table".

Figure 3 showcases two 3D scenes from the 3D-PU dataset, each containing a point cloud, mesh, textured mesh, RGB-D images, and detailed part annotations for all objects. The 10,000 3D scenes are divided into 9,000 for training, with 500

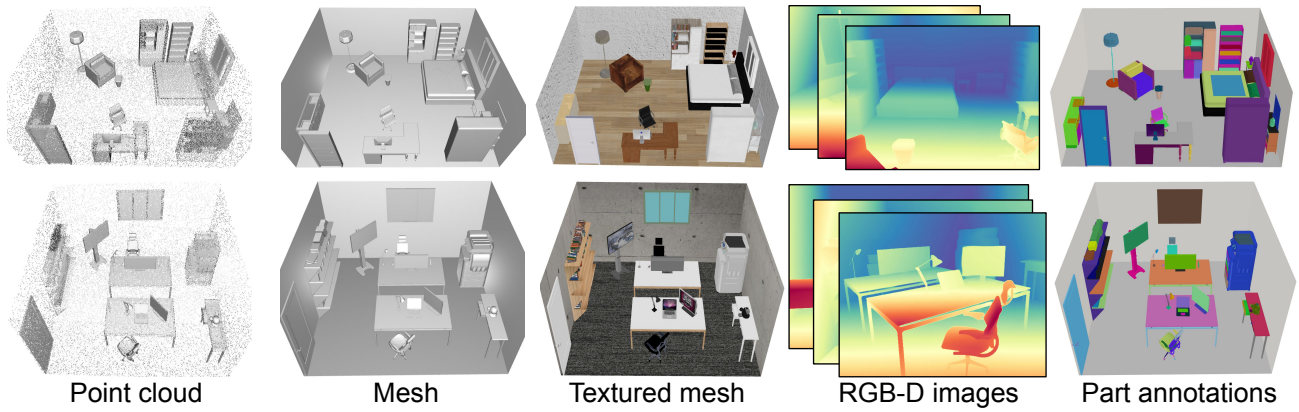

Figure 3: 3D-PU dataset examples. Each 3D scene includes a point cloud, mesh, textured mesh, RGB-D images, and detailed 3D-part annotations for all objects. In total the dataset comes with 10,000 scenes, covering 221,267 objects across 85 object-categories, together with 843,654 dense 3D-part annotationss across 206 part-categories.

each allocated for validation and testing. Having established the dataset we are now ready to highlight our 3D-part level segmentation task and method.

4 OpenPart3D System

System Overview. Our OpenPart3D model is illustrated in Figure 4. First, the *Room-Tour Snap Module* captures multiple view images of the 3D scene mesh by strategically positioning cameras at predefined locations with optimized poses, ensuring clear visibility of small objects and parts. These images are then processed by a 2D open-vocabulary model to generate 2D part masks corresponding to the given text query. Next, the *View-Weighted Voting Module* assigns varying weights to the different views and computes confidence scores for geometrically consistent regions in the 3D point cloud. Finally, the *Progressive Merging Module* adaptively merges high-confidence superpoints to produce the final 3D part segmentation mask. Detailed descriptions of each step are provided below.

Room-Tour Snap. Parts are typically small, making effective 2D part segmentation in subsequent steps challenging. To enhance the quality of part segmentation, it is crucial that each 3D object is not only present but also clearly visible in at least some of the projected view images. In other words, each object should be positioned near the virtual camera in at least some of the images. To address this, we propose the *Room-Tour Snap Module*. As Figure 5 illustrates, the scene is divided into a 3x3 grid, with cameras strategically positioned above each grid cell. Each cell contains a camera placed above its center, oriented toward the centroid of the points within the cell. Additionally, corner cells include one camera aimed at the farthest corner of the scene, side cells have two such cameras, and the center cell is equipped with four cameras pointing toward the farthest corners of the scene. In total, we capture 25 view images across all grid cells, ensuring comprehensive coverage of the scene and clear visibility of each object.

2D Part Segmentation. Upon the projected view images, we apply a pretrained 2D open-vocabulary part segmentor, Florence2 [Xiao *et al.*, 2024], using the text query $\mathcal{T}$ as input. The network generates a set of 2D part masks corresponding to the text query. However, the performance of the vision-language model may be constrained by the specific images on which it was trained. To enhance the model's performance, we fine-tune the vision decoder of the VLM by leveraging ground-truth 2D part annotations projected from the 3D geometries, while freezing the other components of the VLM. Given the input x, which combines the projected images and the text query, and the ground-truth 2D masks y, the objective is formulated as follows:

$$\mathcal{L} = -\sum_{i=1}^{|y|} \log P_\theta(y_i | y_{<i}, x), \tag{1}$$

where θ denotes the vision decoder parameters and $|y|$ represents the total number of target tokens.

Superpoint Generation. In the pre-processing step, the scene points are grouped into geometrically homogeneous regions, referred to as superpoints [Landrieu and Simonovsky, 2018; Landrieu and Obozinski, 2017]. This results in a set of S superpoints $\{\hat{P}_i\}_{i=1}^{S} \in \{0,1\}^{N \times S}$, where $\hat{P}_i$ represents a binary mask of the points. Superpoints serve as foundational elements for 3D parts, improving processing efficiency in subsequent stages.

View-Weighted Voting. The *View-Weighted Voting Module* computes confidence scores for each superpoint by evaluating its consistency with 2D part masks across multiple views. The possibility of each superpoint belonging to the *foreground* class is voted by all 2D part segmentations across multiple projected view images that overlap with its 2D projection. Specifically, for a superpoint $\hat{P}_i$, its score s_i is calculated as the weighted ratio of its visible points covered by all 2D part masks across all views:

$$s_i = \frac{\sum_v \sum_{p \in \hat{P}_i} [\mathrm{VIS}_v(p)]\,[\mathrm{INS}(p)]\,W_v}{\sum_v \sum_{p \in \hat{P}_i} [\mathrm{VIS}_v(p)]\,W_v}, \tag{2}$$

where $[\cdot]$ represents the Iverson bracket (which evaluates to 1 if the predicate inside is true, and 0 if false); $\mathrm{VIS}_v(p)$ indicates whether the 3D point p is visible in view v; and $\mathrm{INS}(p)$

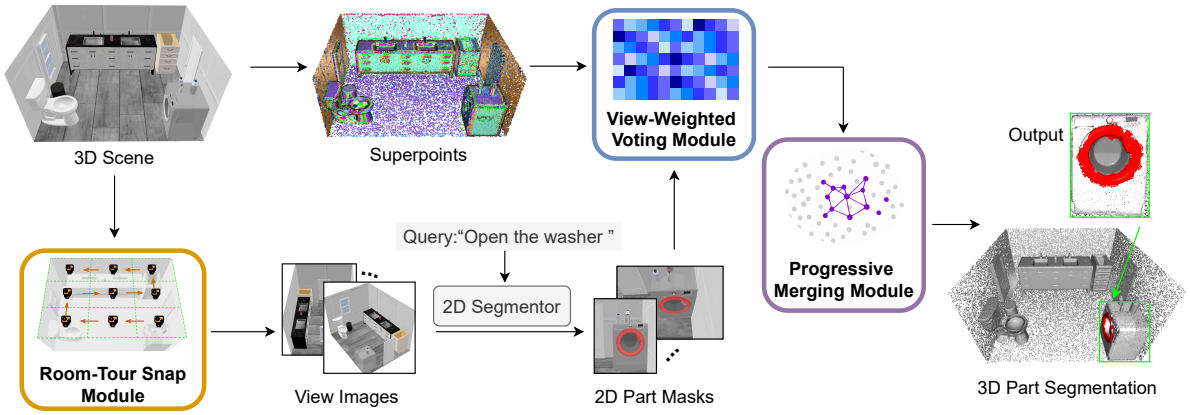

Figure 4: OpenPart3D system overview. First, the *Room-Tour Snap Module* captures multiple view images of the 3D scene mesh by strategically positioning cameras at predefined locations with optimized poses. These images are then processed by a 2D open-vocabulary model to generate 2D part masks corresponding to the given text query. Subsequently, the *View-Weighted Voting Module* assigns varying weights to the different views and computes confidence scores for geometrically consistent regions in the 3D point cloud. Finally, the *Progressive Merging Module* adaptively merges high-confidence superpoints to produce the final 3D part segmentation mask.

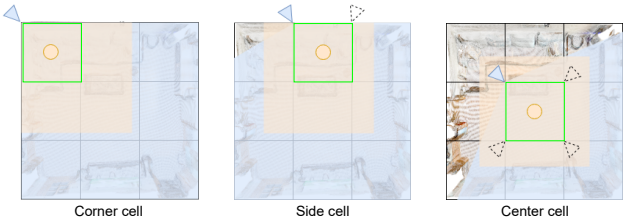

Figure 5: Camera position and pose. The orange circle ◯ represents a camera positioned above the center of a grid cell, oriented toward the cell's centroid, while the blue triangle ▽ indicates a camera oriented toward the farthest corner of the scene. Within the 3×3 grid of the scene, each cell contains a camera above its center, directed at its centroid. Additionally, corner cells include one camera pointing toward the farthest corner of the scene, side cells have two such cameras, and the center cell is equipped with four cameras oriented toward the farthest corners of the scene.

indicates whether the projection of point p in view v is inside the part masks.

Not all views contribute equally to the voting process, as certain views may provide more informative perspectives of the target instances. The weight $W_v = \{1, 2, 3\}$ is assigned to view v based on the spatial proximity between its camera position and the corresponding part masks. A higher weight is assigned to views where the camera positions are closer to the part masks: If the 2D masks and the camera are located within the same grid cell, $W_v = 3$. If they are in neighboring grid cells, $W_v = 2$. If they are in non-adjacent grid cells, $W_v = 1$.

Progressive Merging. To determine the final 3D segmentation, we leverage the confidence score s computed for each superpoint to merge high-confidence *foreground* regions. However, directly applying a fixed threshold to these scores often leads to suboptimal results: too high a threshold causes over-segmentation, while too low a threshold results in under-segmentation.

To address this, we propose a progressive merging strategy. We begin by filtering out superpoints with confidence scores below 0.5. The remaining superpoints are then sorted in descending order of their scores and merged incrementally—from the most to the least confident—until the projected area of the merged set reaches the total area covered by all 2D masks across views. This adaptive approach avoids the limitations of fixed-threshold heuristics and improves both the robustness and accuracy of the final 3D part segmentation. The full procedure is outlined in Algorithm 1.

Algorithm 1 Progressive Merging Strategy

Require: Set of superpoints $\{\hat{P}_i\}$ with confidence scores $\{s_i\}$, total area of all 2D masks across views A_{target}
Ensure: Final merged points $\mathcal{M}$
Filter out superpoints with scores $s_i < 0.5$
Sort the remaining $\{\hat{P}_i\}$ in descending order of $\{s_i\}$
Initialize merged set $\mathcal{M} \leftarrow \emptyset$
for each superpoint $\hat{P} \in \{\hat{P}_i\}$ **do**
  $\mathcal{M} \leftarrow \mathcal{M} \cup \hat{P}$
  if ProjectArea$(\mathcal{M}) \geq A_{\text{target}}$ **then**
    break
  end if
end for
return $\mathcal{M}$

5 Results

Benchmarks. We leverage the 3D-PU dataset as the benchmark for the task of open-vocabulary 3D part segmentation in scenes. We train the approach on the 3D-PU training set and evaluate on the test set. For further evaluation on real-world dataset, we also adapt the MultiScan dataset [Mao *et al.*, 2022], which includes 5,129 manually annotated part instances across 5 part categories (static, door, drawer, window, lid) within 17 object categories in 117 scenes. We create "object_part" labels based on each part's category and associated

	3D-PU		MultiScan	
Fine-tuning	AP_{50}	AP_{25}	AP_{50}	AP_{25}
✗	10.3	21.2	11.1	23.5
✓	17.9	35.6	13.8	30.3
	(+7.6)	(+14.4)	(+2.7)	(+6.8)

Table 1: Effect of fine-tuning. Fine-tuning the vision decoder of our model on the synthetic 3D scene data from the 3D-PU dataset significantly enhances its 3D-part understanding capabilities, yielding improved performance on both 3D-PU and MultiScan datasets.

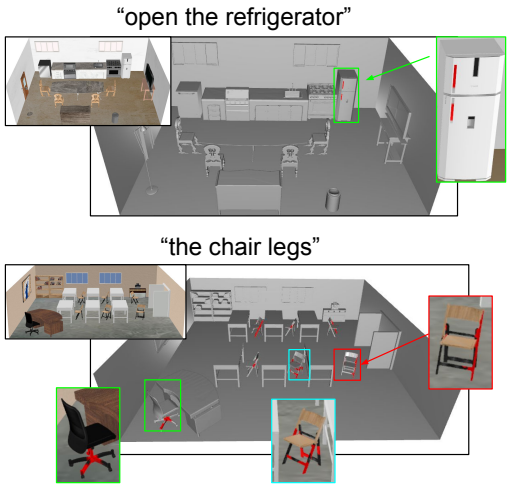

Figure 6: Qualitative results on the 3D-PU test set. In the first scene, the small handle (in green box) of the refrigerator is accurately segmented in response to the query "open the refrigerator." In the second scene, most chair legs are identified, with only a few legs overlooked (in red box), demonstrating an overall strong performance.

object category, resulting in 47 unique labels, such as "cabinet_door" and "cabinet_drawer." We use these "object_part" labels as text queries.

Metrics. As primary metrics, we report the mean Average Precision (mAP) at Intersection over Union (IoU) thresholds of 0.25 and 0.5, referred to as AP_{25} and AP_{50}. Additionally, we provide the AP, calculated as the average mAP across IoU thresholds from 0.5 to 0.95 in 0.05 increments.

Effect of Fine-Tuning. Table 1 compares the 3D part segmentation results with and without fine-tuning using the synthetic data from the 3D-PU dataset. Without fine-tuning, our approach already achieves reasonable performance on 3D-PU, indicating that the pre-trained vision-language model can effectively interpret the synthetic scenes in the 3D-PU dataset. Fine-tuning the vision decoder of our model on the 3D-PU training set enhances the part segmentation performance considerably on both 3D-PU and MultiScan datasets, demonstrating that the 3D scene data in the 3D-PU dataset can effectively improve the model's capability in 3D part understanding. In Figure 6, we present two qualitative examples of 3D part segmentation from the 3D-PU dataset.

Effect of Varied View Generation. The method of view generation plays a crucial role, as it directly impacts the ability of the view images to capture small objects and parts. Table 2 compares various view generation techniques on the 3D-PU dataset. The *Global Snap* method moves the camera around the scene, always pointing directly toward the scene's center. The *Multi-view Snap* [Kundu *et al.*, 2020] samples synthetic viewpoints to maximize 2D semantic context, ensure geometric diversity, and mitigate occlusions. The *Multi-level Snap* [Huang *et al.*, 2024] captures view images at three different scales. The results demonstrate that the view images generated by our *Room-Tour Snap* method lead to the best performance in part segmentation.

View generation	#views	AP_{50}	AP_{25}
Global snap	24	13.5	29.3
Multi-view snap [Kundu *et al.*, 2020]	25	13.7	30.6
Multi-level snap [Huang *et al.*, 2024]	24	15.3	32.9
Room-tour snap (ours)	25	**17.9**	**35.6**

Table 2: Effect of varied view generation on 3D-PU dataset. View images generated by our *Room-Tour Snap Module* are effective in capturing small objects and parts, resulting in the best performance for part segmentation.

View weight	AP_{50}	AP_{25}
✗	14.7	31.6
✓	**17.9**	**35.6**

Table 3: Effect of view weight on 3D-PU dataset. Placing more emphasis on views where the camera position is closer to the part masks can benefit part segmentation performance.

Effect of View Weight. The *View-weighted Voting Module* prioritizes views where the camera position is closer to the 2D part masks. Table 3 illustrates that assigning different weights to views based on the distance between the camera position and the 2D part masks can enhance part segmentation performance.

Effect of Varied Merging Strategies. The *Progressive Merging Module* merges the high-confidence regions in a adaptive way other than relying on a fixed threshold. Table 4 illustrates that the adaptive approach leads to more accurate and reliable 3D part segmentation compared to fixed-threshold merging.

Open-Vocabulary 3D-part Segmentation in Scenes. Given that most existing methods are not directly applicable to open-vocabulary part segmentation in 3D scenes, we implement several strong baselines by adapting author-released codebases. OpenScene [Peng *et al.*, 2023] generates dense point-level features for 3D scenes, and we aggregate features of 3D parts based on cosine similarity to CLIP [Radford *et al.*, 2021] embeddings to respond to text queries. We construct the two-stage baseline by concatenating OpenIns3D [Huang *et al.*, 2024] and PartDistill [Umam *et al.*, 2024]. So that

Merging strategy	AP_{50}	AP_{25}
Fixed threshold	14.2	30.8
Adaptive merging	**17.9**	**35.6**

Table 4: Effect of varied merging strategies on 3D-PU dataset. The proposed adaptive merging procedure enhances the precision of the final 3D part segmentation.

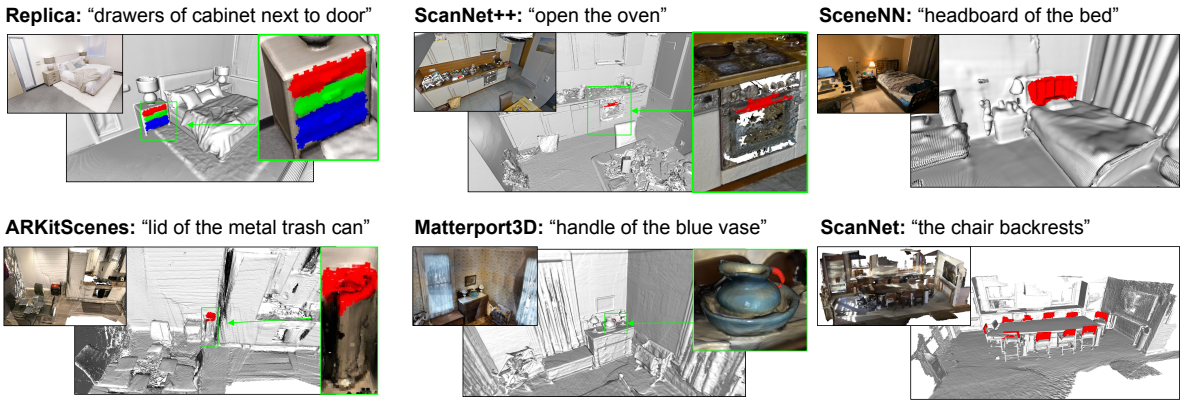

Figure 7: Generalization to real-world 3D datasets. OpenPart3D can be applied to a wide range of real-world datasets, including Replica, ScanNet++, SceneNN, ARKitScenes, Matterport3D, and ScanNet. We believe it is the first time 3D-part segmentation works on these datasets. Note that due to the absence of part-level annotations on these datasets, quantitative evaluation is not feasible.

		3D-PU			MultiScan		
	RGB-D	AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}
OpenScene	✗	4.1	8.7	15.3	3.1	5.4	13.3
OpenIns3D+PartDistill	✗	4.7	9.8	19.5	3.9	7.3	15.2
Search3D	✓	6.1	12.7	27.5	5.9	11.5	26.4
OpenPart3D (ours)	✗	**8.7**	**17.9**	**35.6**	**7.6**	**13.8**	**30.3**

Table 5: Open-Vocabulary 3D-part Segmentation on 3D-PU and MultiScan datasets. We constructed the baselines from the author-provided code. Our method achieves the best performance on both synthetic and real-world datasets, demonstrating its strong capability in 3D part segmentation.

		SceneFun3D	
	RGB-D	AP_{50}	AP_{25}
LERF [Kerr *et al.*, 2023]	✓	4.9	11.3
OpenMask3D-F [Delitzas *et al.*, 2024]	✓	8.0	17.5
Fun3DU [Corsetti *et al.*, 2025]	✓	12.4	23.6
OpenPart3D (ours)	✗	**13.3**	**25.7**

Table 6: Languge-guided functionality segmentation on the SceneFun3D dataset. OpenPart3D outperforms baselines despite not utilizing well-aligned RGB-D images as input.

OpenIns3D can localize the objects of interest in the scene according to the text query then PartDistill can further segment the parts within the objects of interest. Search3D [Takmaz *et al.*, 2025] constructing 2D object-part representations that correspond to 3D objects, relying on well-aligned RGB-D images as input. We train our method and the baseline methods on the 3D-PU training set and evaluate them on the 3D-PU test set and the MultiScan dataset. For fair comparison, we ensure consistent input and fine-tuning across all methods. As shown in Table 5, our method achieves the best performance on both synthetic and real-world datasets, demonstrating its superior 3D part segmentation capability.

Language-guided Functionality Segmentation in 3D Scenes. Language-guided functionality segmentation [Delitzas *et al.*, 2024] focuses on identifying functional, interactive elements in scenes based on a task description, such as "open the door" or "turn on the ceiling light." These interactive elements are often small parts of objects, making our approach particularly well-suited for this task. We compare our method with LERF [Kerr *et al.*, 2023], OpenMask3D-F [Delitzas *et al.*, 2024] and Fun3DU [Corsetti *et al.*, 2025] on the SceneFun3D [Delitzas *et al.*, 2024] functionality segmentation benchmark. Notably, while all baselines rely on well-aligned 2D-3D pairs as input, our approach operates directly on 3D geometries alone. As shown in Table 6, our method outperforms all baselines on the SceneFun3D dataset even without requiring aligned RGB-D images as input.

Generalization to Real-World Datasets. As shown in Table 1 & 5 & 6, our OpenPart3D, finetuned on the 3D-PU dataset, demonstrates strong performance on the real-world MultiScan and SceneFun3D datasets. Since MultiScan and SceneFun3D are currently the only real-world 3D scene datasets with part annotations, quantitative evaluation on additional real-world datasets is not feasible. Nevertheless, we apply OpenPart3D to a wide range of real-world 3D datasets. Figure 7 presents qualitative visualizations of open-vocabulary part segmentation on datasets including Replica, ScanNet++, SceneNN, ARKitScenes, Matterport3D, and ScanNet. We believe this marks the first application of 3D-part segmentation on these datasets, highlighting the strong generalization of our approach.

6 Conclusion

In this paper, we introduce the task of segmenting any 3D-part in scenes based on a natural language description. To address data limitations, we propose a cost-effective approach for constructing 3D scene data with extensive, fine-grained part-level annotations. This method culminates in the creation of the first large-scale 3D scene dataset with detailed part annotations, termed the 3D-PU dataset. To tackle the challenges of open-vocabulary 3D-part segmentation, we present OpenPart3D. Our approach, evaluated on the 3D-PU dataset and the restructured MultiScan dataset, demonstrates superior performance over baseline methods in open-vocabulary 3D part segmentation, and showcases strong generalization across multiple real-world 3D scene datasets.

References

[Baruch *et al.*, 2021] 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 *NeurIPS*, 2021.

[Chang *et al.*, 2015] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv*, 2015.

[Chang *et al.*, 2017] 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. *3DV*, 2017.

[Chen *et al.*, 2023] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3d scene understanding by clip. In *CVPR*, 2023.

[Corsetti *et al.*, 2025] Jaime Corsetti, Francesco Giuliari, Alice Fasoli, Davide Boscaini, and Fabio Poiesi. Functionality understanding and segmentation in 3d scenes. In *CVPR*, 2025.

[Dai *et al.*, 2017] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017.

[Delitzas *et al.*, 2024] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-grained functionality and affordance understanding in 3d scenes. In *CVPR*, 2024.

[Ding *et al.*, 2023] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *CVPR*, 2023.

[Ding *et al.*, 2024] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Lowis3d: Language-driven open-world instance-level 3d scene understanding. *TPAMI*, 2024.

[Hua *et al.*, 2016] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *3DV*, 2016.

[Huang *et al.*, 2024] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *ECCV*, 2024.

[Kerr *et al.*, 2023] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *ICCV*, 2023.

[Kirillov *et al.*, 2023] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *ICCV*, 2023.

[Kundu *et al.*, 2020] Abhijit Kundu, Xiaoqi Yin, Alireza Fathi, David Ross, Brian Brewington, Thomas Funkhouser, and Caroline Pantofaru. Virtual multi-view fusion for 3d semantic segmentation. In *ECCV*, 2020.

[Landrieu and Obozinski, 2017] Loic Landrieu and Guillaume Obozinski. Cut pursuit: Fast algorithms to learn piecewise constant functions on general weighted graphs. *Journal on Imaging Sciences*, 2017.

[Landrieu and Simonovsky, 2018] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018.

[Liu *et al.*, 2023] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv*, 2023.

[Mao *et al.*, 2022] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. In *NeurIPS*, 2022.

[Mo *et al.*, 2019] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019.

[Nguyen *et al.*, 2024] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3DIS: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *CVPR*, 2024.

[Peng *et al.*, 2023] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, 2023.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[Ravi *et al.*, 2024] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv*, 2024.

[Richardson *et al.*, 2023] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *SIGGRAPH*, 2023.

[Schult *et al.*, 2023] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *ICRA*, 2023.

[Straub *et al.*, 2019] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, Anton Clarkson, Mingfei Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke M. Strasdat, Renzo De Nardi, Michael Goesele, Steven Lovegrove, and Richard Newcombe. The replica dataset: A digital replica of indoor spaces. *arXiv*, 2019.

[Takmaz *et al.*, 2023] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-vocabulary 3d instance segmentation. In *NeurIPS*, 2023.

[Takmaz *et al.*, 2025] Ayca Takmaz, Alexandros Delitzas, Robert W Sumner, Francis Engelmann, Johanna Wald, and Federico Tombari. Search3d: Hierarchical open-vocabulary 3d segmentation. *Robotics and Automation Letters*, 2025.

[Umam *et al.*, 2024] Ardian Umam, Cheng-Kun Yang, Min-Hung Chen, Jen-Hui Chuang, and Yen-Yu Lin. Partdistill: 3d shape part segmentation by vision-language model distillation. In *CVPR*, 2024.

[Xiang *et al.*, 2020] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. Sapien: A simulated part-based interactive environment. In *CVPR*, 2020.

[Xiao *et al.*, 2024] Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. In *CVPR*, 2024.

[Xue *et al.*, 2025] Feng Xue, Wenzhuang Xu, Guofeng Zhong, Anlong Ming, and Nicu Sebe. Cues3d: Unleashing the power of sole nerf for consistent and unique instances in open-vocabulary 3d panoptic segmentation. *Information Fusion*, 2025.

[Yang *et al.*, 2024] Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In *CVPR*, 2024.

[Yeshwanth *et al.*, 2023] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *ICCV*, 2023.

[Yuksel, 2015] Cem Yuksel. Sample elimination for generating poisson disk sample sets. In *Computer Graphics Forum*, 2015.

[Zeng *et al.*, 2023] Yihan Zeng, Chenhan Jiang, Jiageng Mao, Jianhua Han, Chaoqiang Ye, Qingqiu Huang, Dit-Yan Yeung, Zhen Yang, Xiaodan Liang, and Hang Xu. Clip2: Contrastive language-image-point pretraining from real-world point cloud data. In *CVPR*, 2023.

[Zhou *et al.*, 2023] Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. Uni3d: Exploring unified 3d representation at scale. *arXiv*, 2023.