OpenIns3D: Snap and Lookup for 3D Open-vocabulary Instance Segmentation

Zhening Huang[1] Xiaoyang Wu[2] Xi Chen[2]
Hengshuang Zhao[2⋆] Lei Zhu[3,4] Joan Lasenby[1]

[1]University of Cambridge [2]The University of Hong Kong
[3]Hong Kong University of Science and Technology
[4]The Hong Kong University of Science and Technology (Guangzhou)

https://zheninghuang.github.io/OpenIns3D/

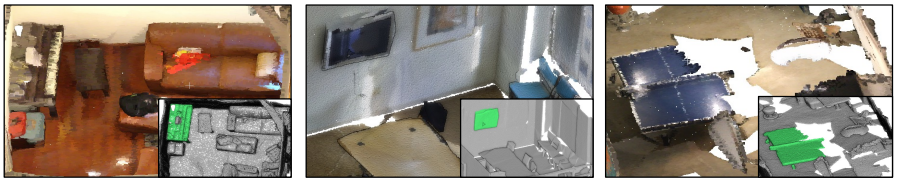


Fig. 1: Complex Queries 3D Instance Segmentation with OpenIns3D.

Abstract. In this work, we introduce OpenIns3D, a new 3D-input-only framework for 3D open-vocabulary scene understanding. The OpenIns3D framework employs a "Mask-Snap-Lookup" scheme. The "Mask" module learns class-agnostic mask proposals in 3D point clouds, the "Snap" module generates synthetic scene-level images at multiple scales and leverages 2D vision-language models to extract interesting objects, and the "Lookup" module searches through the outcomes of "Snap" to assign category names to the proposed masks. This approach yet simple, achieves state-of-the-art performance across a wide range of 3D open-vocabulary tasks, including recognition, object detection, and instance segmentation, on both indoor and outdoor datasets. Moreover, OpenIns3D facilitates effortless switching between different 2D detectors without requiring retraining. When integrated with powerful 2D open-world models, it achieves excellent results in scene understanding tasks. Furthermore, when combined with LLM-powered 2D models, OpenIns3D exhibits an impressive capability to comprehend and process highly complex text queries that demand intricate reasoning and real-world knowledge.

Keywords: Open-Vocabulary Understanding · 3D Scene Understanding · Vision-Language Model

1 Introduction

3D scene understanding plays a critical role in various domains, such as autonomous driving, robotic sensing, AR/VR, and manufacturing, among others.

⋆ Corresponding author.

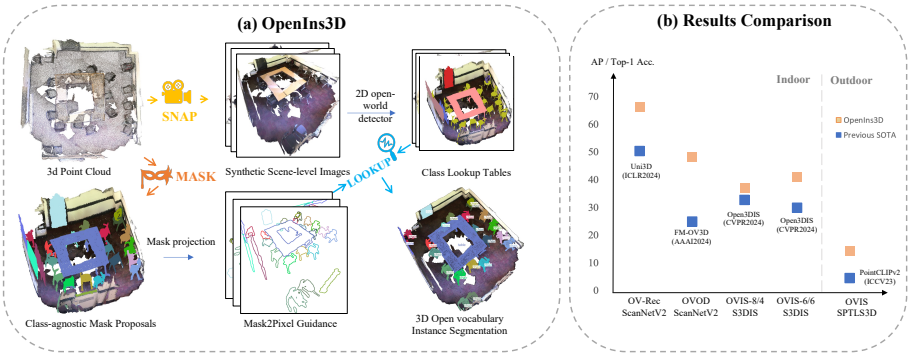


Fig. 2: High-level Illustrations of OpenIns3D and Quantitative Results. (a) OpenIns3D follows the *"Mask-Snap-Lookup"* steps for open-vocabulary scene understanding. (b) A list of SOTA results has been achieved on both indoor and outdoor datasets. OV-Rec: open-vocabulary object recognition. OVOD: open-vocabulary object detection. OVIS: open-vocabulary instance segmentation. PointCLIPV2 [49]; Uni3D [45]; Open3DIS [25]; FM-OV3D [41]

.

While the development of 3D closed-set understanding is relatively mature, scene understanding in an open-vocabulary setting is still in its infancy. Closed-set understanding can only handle a predefined set of concepts and scenarios but fails to provide valid responses when faced with unfamiliar concepts or variations in language usage. This limitation impacts its performance in dynamic and ever-changing contexts.

Thanks to internet-scale image-text datasets, significant progress has been made in 2D image open-vocabulary understanding [4, 7, 11, 17, 27, 38, 43, 44]. However, unlike 2D data that can be easily collected from the internet, constructing a large-scale 3D-text dataset poses a challenge. As a result, the most viable approach to achieving 3D open-vocabulary understanding involves leveraging 2D images to bridge language and 3D data. In this direction, there have been several notable works, such as OpenScene [26], PLA-family [9, 10, 39], and CLIP2Scene [6]. These works leverage well-aligned 2D images and 3D point clouds to conduct feature distillation or employ 2D caption models to construct 3D-text pairs. One prerequisite of these methods, however, is the availability of well-aligned 2D images and 3D point clouds. This means that posed 2D images, associated depth maps and camera models need to be accessible as inputs to the network. In real-life scenarios, there are numerous cases where 2D images or the information required to align 2D and 3D data are unavailable. For instance, to save storage space, point clouds generated with LiDAR are often stored without accompanying 2D images (example datasets include [13, 29, 36]). In cases where point clouds are obtained from the registration of multiple scans from different sensors or are converted from 3D simulations/CAD models [12, 24], 2D images are often non-existent.

We believe that developing a 3D open-vocabulary framework without relying on well-aligned 2D images is meaningful, as this will simplify deployment pre-

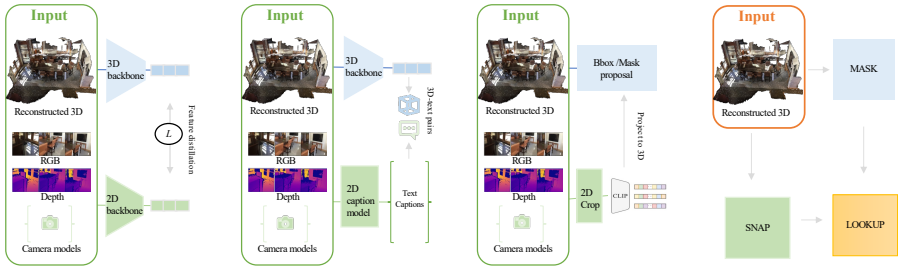
requisites and enhance its applicability across a wide range of scenarios. To this end, we introduce *OpenIns3D*, a framework designed to effectively perform 3D open-vocabulary scene understanding tasks without relying on 2D aligned images. Overall, OpenIns3D comprises three core steps: *Mask*, *Snap*, and *Lookup*. An overall illustration of OpenIns3D is presented in Figure 2a.

Mask: Given a 3D point cloud, the first part of OpenIns3D learns class-agnostic mask proposals with a *Mask Proposal Module* (MPM). This process is trained without any classification labels. To control the quality of the mask, MPM proposes a learnable *Mask Scoring* module to predict the quality of each mask output and implements a list of *Mask Filtering* techniques to discard invalid, low-quality masks. MPM outputs a list of class-agnostic masks in the scene.

Snap: Multiple synthetic scene-level images are generated with calibrated and optimized camera poses and intrinsic parameters. These images are specifically designed to encompass part or all of the relevant masks, aiming to minimize the need for multiple renderings. Instead of individually predicting the category of each mask proposal [22, 42, 49], the scene-level images are input into 2D open-vocabulary models for the simultaneous understanding of all interesting objects present in the scene. A Class Lookup Table (CLT) is then constructed to store all the detected object categories alongside their respective pixel locations.

Lookup: To precisely determine the positions of mask proposals in each image, Mask2Pixel maps are constructed. These maps project all 3D mask proposals onto 2D images with identical camera parameters used in *Snap*. In the *Lookup* phase, OpenIns3D searches through the CLT with the help of Mask2Pixel maps to precisely assign category names to 3D mask proposals. Results from multiple views are combined to establish initial mask classification outcomes. For remaining masks, a similar *Lookup* procedure is carried out on a local scale to facilitate classification. Lastly, the 3D mask proposals are refined by removing masks lacking class assignments after both global and local *Lookup*.

OpenIns3D demonstrates promising performance in extensive comparisons with other methods, as summarized in Figure 2b. This simple and flexible approach achieves a range of state-of-the-art (SOTA) results on various 3D open-vocabulary tasks. Specifically, SOTA results are achieved on open-vocabulary instance segmentation for the indoor S3DIS [1] dataset and the outdoor STPLS3D [5] dataset. OpenIns3D even outperforms OpenMask3D [35], a concurrent work that heavily utilizes 2D images, for instance, segmentation on the challenging Replica [33] dataset. We also find that the "Snap and Lookup" serves as a powerful open-vocabulary object recognition engine, which achieved SOTA object recognition tasks on ScanNet [8], outperforming the previous SOTA, a 1-billion parameter 3D foundation model [45]. Lastly, when converting mask proposals into 3D bounding boxes, OpenIns3D also achieved state-of-the-art results in open-vocabulary object detection (OVOD) on ScanNet, outperforming previous image-dependent methods [41]. The design of OpenIns3D also allows 2D detectors to be changed without the need for retraining. This provides the model with the capability to evolve alongside the latest development of the 2D open-vocabulary models. Moreover, when 2D detectors are coupled with Large Lan-



(a) 3D Feature Distillation (b) Building 3D-text pair (c) CLIP and Projection (d) Mask-Snap-Lookup (ours)

Fig. 3: Four Categories of Open-Vocabulary 3D Scene Understanding Models. a) 3D feature distillation frameworks, where 2D images are used as a bridge to distil language-aligned features into 3D, with typical works including OpenScene [26] and Clip2Scene [6]. b) Building 3D-text pairs, where 2D captioning models are used to build 3D-text pairs for feature learning, with typical works including the PLA-family [9, 10, 39] c) CLIP and Projection, where objects are cropped out of 2D images before being processed by CLIP, and the results are directly projected into 3D, including OpenMask3D [35], OV-3DET [22], CLIP^2 [40] and Open3DIS [25]. d) OpenIns3D

guage Models (LLMs), OpenIns3D can enable complex query understanding capability. When integrated with LISA [19], an LLM-powered reasoning segmentation model, OpenIns3D exhibits a strong ability to comprehend highly intricate language queries and performs reasoning segmentation in 3D, as illustrated in Figure 1. In summary, our contributions are:

- OpenIns3D employs a distinct pipeline that operates without the need for well-aligned images. This approach achieves state-of-the-art results across a range of benchmarks and possesses the ability to comprehend highly complex input queries.
- The proposed "Snap and Lookup" combination can serve as a powerful 3D object recognition engine, especially for noisy 3D objects extracted from scene-level scans. This capability is demonstrated by achieving state-of-the-art results with a large margin.

2 Related work

3D open-vocabulary understanding. Progress in 3D open-vocabulary understanding has been relatively slow compared to that of images. In the domain of 3D object classification tasks, methods like PointCLIP [42], PointCLIPV2 [49], and CLIP2Point [15] project 3D point clouds into depth maps and link them with 2D models for classification. However, these methods lack performance in scene-level understanding, where points are often overlapped and incomplete. For scene-level understanding, most work has primarily focused on leveraging well-aligned 2D posed images, depth maps, and point clouds [6, 9, 10, 26, 30, 39, 40]. One notable example is OpenScene [26], which takes posed 2D images, depth maps, and 3D data as input, and feature distillation is performed to transfer 2D language-aligned features from images to 3D point clouds. Similarly,

Clip2Scene [6] builds dense pixel-point pairs by calibrating the LiDAR point cloud with corresponding images captured by six cameras. However, achieving instance-level understanding is challenging with these methods as they focus solely on semantic-level understanding. In contrast, PLA [9] and its follow-up work RegionPLC [39] and Lowis3D [10] utilize a 2D caption model to construct 3D-text pairs to learn features. However, the PLA-family works rely on a binary head to classify the input object into base categories or novel categories, and the transferability of this binary head to different base-novel splits is very limited, posing a challenge for flexible applications. One current work, OpenMask3D [35], utilizes well-aligned 2D images to learn features for mask proposals, leading to impressive results in open-vocabulary instance segmentation. Open3DIS [25] follows the same procedure, enhancing both mask proposals and mask classification with images, yielding better performance. A common issue with these methods is their reliance on well-aligned 3D and 2D pairs in the input, which may not always be available in real applications. We summarise the key difference between these methods in Figure 3. Simplifying input requirements improves method flexibility and compatibility. In this work, we explore how to conduct 3D open-vocabulary understanding without relying on 2D images.

Image generation from 3D. Projection-based methods have been extensively explored in the past for 3D understanding and have proven to be beneficial for obtaining complementary features. For instance, MVCNN [34] projects 3D objects to different views to aid in feature learning, while LAR [3] introduces object centre projection methods to generate images for 3D objects from various angles, assisting visual grounding tasks. Additionally, Virtual View Fusion [18] employs the original camera pose but enlarges the field of view, resulting in enhanced 2D feature transfer. However, these methods encounter challenges like best view selection, object occlusion, information loss during projection, and long rendering times. In the context of open-vocabulary settings, the quality of the projected image plays a crucial role in model performance. In our work, we evaluate different projection methods, along with their compatibility with 2D open-vocabulary models, to identify an optimal solution that achieves good results and is efficient to implement.

3 OpenIns3D

3.1 Baseline and Challenges

Baseline. We build a naive baseline by adopting the recent 3D instance segmentation backbone Mask3D [32] to generate mask proposals. To make the Mask Proposal Module (MPM) fit for the open-vocabulary setting, we remove all components in Mask3D that use the classification labels. Later, PointCLIP [49] is adopted for mask understanding. This naive approach, although satisfying the requirement of 3D inputs only, has long rendering times and unsatisfactory performance (See in Table 7.) There are several problems in this baseline model.

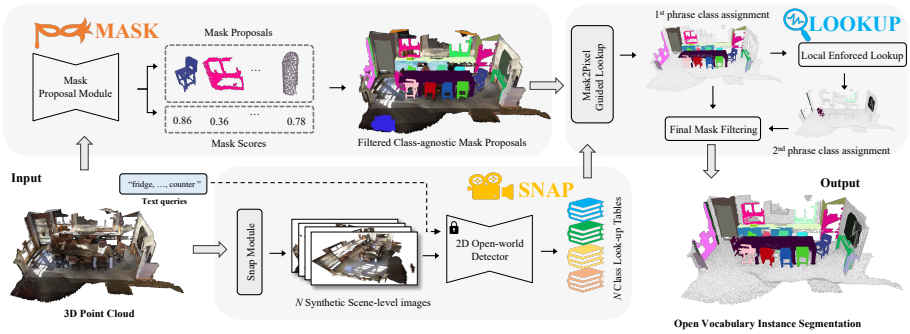


Fig. 4: General Pipeline of OpenIns3D OpenIns3D first processes point clouds with MPM to generate 3D mask proposals and mask scores. The *Snap* module (detailed in Figure 5) then renders N synthetic scene-level images, which are later passed into the 2D open-world model along with the input text queries. The detection results from the 2D model are stored in the *Class Lookup Table* (CLT). Finally, both the mask proposals and CLT are fed into the *Lookup* module, where *Mask2Pixel Guided Lookup* (detailed in Figure 6) is performed at the global level, followed by a *Local Enforced Lookup* at the local level to unlock the semantic meaning of mask proposals. The final mask filtering refines the mask proposals and obtains the final results.

Challenge 1: excessive mask proposals. in Mask3D [32], mask proposals are filtered by the mask classification logit, which is removed in the class-agnostic setting. Therefore, an effective mask filtering scheme is needed.
Challenge 2: low quality of 3D instances. 3D instances extracted from scene scans are typically broken, uncompleted, and sparse. Therefore, generated images by simple projection are not easily understood by 2D VL models.
Challenge 3: lack of context information. Humans recognize imperfect 3D point cloud objects through scene comprehension. However, isolated instance point cloud projections lack this context. A solution is to project both the mask and background onto images, yet this introduces distracting elements, potentially confusing classification-level models like CLIP.
Challenge 4: domain gap between projected images and natural images. Rendered images often differ significantly from natural images used in training, posing a challenge for 2D visual language models to comprehend.

3.2 Overall Framework

In this section, we present our design of OpenIns3D, which targets the aforementioned four problems. The pipeline of OpenIns3D is shown in Figure 4.

Mask: class-agnostic mask proposal

Within the Mask Proposal Module (MPM), we introduce two straightforward designs aimed at filtering low-quality masks generated in the baseline model.
Mask scoring. Inspired by [7, 16, 17], We feed the instance queries generated from the Mask Module into a shallow MLP module to predict the quality, i.e

IoU of the mask. The predicted IoU (IoU_m) is supervised by the ground truth IoU (IoU_{gt}) value during the training stages, which is calculated between the predicted mask and its matched ground truth mask in the Bipartite Matching. For unmatched prediction masks (IoU_u), we label the ground truth IoU value as zero. The loss is computed using $L2$. To avoid overly low IoU predictions, a hyper-parameter γ is introduced to reduce the weight of loss for unmatched masks. Therefore, the total loss function for the MPM is:

$$\mathcal{L}_{\text{total}} = \gamma \sum (IoU_u)^2 + \sum (IoU_m - IoU_{\text{gt}})^2 \tag{1}$$

Mask filtering. To enhance mask quality, three filters are applied. Firstly, we retain masks with a model-predicted IoU score above a threshold of β, ensuring that only high-quality masks are kept. Secondly, drawing inspiration from SAM [17], we focus on stable masks by comparing two binary masks derived from the same underlying soft mask using different threshold values. Specifically, we introduce an offset value α and select masks where the IoU between the pair of thresholded masks (one with $-\alpha$ and the other with $+\alpha$) exceeds 80%. Lastly, small objects in the scene often lead to invalid proposals, so we filter out mask proposals that have a point number lower than N_{min}. By employing these techniques, the quantity of mask proposals will decrease, resulting in cleaner and higher-quality masks for subsequent mask understanding tasks (*Challenge 1*).

Snap: Synthetic Scene-level View Generation

Rendering images from points can be a time-consuming task, especially when the number of rendering jobs is high. We propose a multi-scale synthetic scene-level image scheme.

Camera pose selection. The Snap module captures scene-level images at three scales: global, corner, and wide-angle, as shown in Figure 5. For global-level images, cameras are positioned above the scene and point directly toward the centre of the scene. For corner images, cameras are positioned above the centre and point toward the corner, while for the wide-angle images, cameras are positioned at the 3 by 3 grid interaction points, pointing toward the furthest corner. Using the camera position coordinates P_{cam}, target coordinates P_{target}, and the up axis of the scene U, the *Lookat* function can be employed to determine the pose matrix *Pose*. A more detailed mathematical formulation of this is presented in the supplementary materials.

Camera intrinsic calibration. Once the camera extrinsic matrix is established, the fields of view for the captured images are adjusted by modifying the camera's intrinsic parameters. The goal is to ensure that the entire scene or a specific part of it is encompassed within the captured images. To achieve this goal, we initialize an arbitrary camera intrinsic matrix and then adjust the focal lengths (f_x and f_y) and the principal point coordinates (c_x and c_y) through scaling. The scaling is performed by readjusting the projected areas in the image coordinate space. For example: if the projected points of the pre-defined area

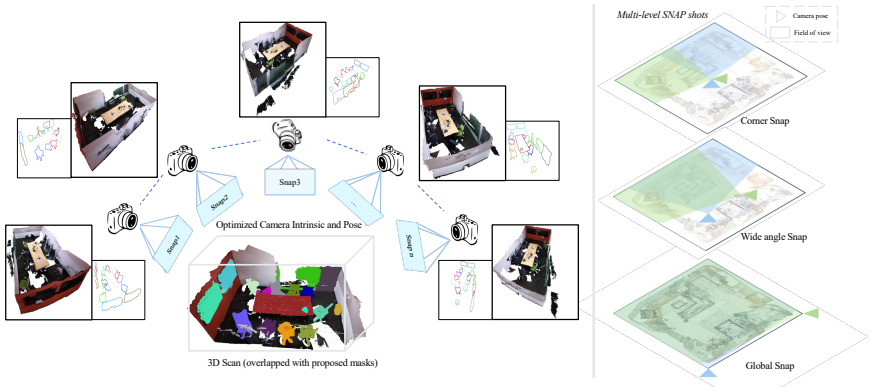


Fig. 5: *Snap* and Mask2Pixel Maps. Multiscale snaps are conducted to render images with different levels of detail for scene understanding, including wide-corner snaps, wide-angle snaps, and global snaps. Cameras are positioned on the top of the scene and point towards the centre or corners, and the field of view is determined with the calibrated intrinsic matrix. With the defined camera models, Mask2Pixel maps are built to store the location of each 3D mask in the 2D image (using the same colour to represent 2D-3D correspondences) to guide the search for category names.

were located in image coordinates within the range of $[-1000, -192]$ in the x-domain, our calibrated intrinsic parameters transform this range to $[0, 1000]$ in x. Importantly, we preserve the aspect ratio between the x and y coordinates to maintain the proportions of the final image without any distortion. This procedure ensures that each captured image is fully utilized and encompasses all regions of interest within the scene (*Challenge 2 & 3*).

Class lookup table. Upon obtaining N synthetic scene-level images, we input them into a 2D open-vocabulary detector. With text queries provided for interested classes, a list of detected objects in synthetic images can be obtained. Subsequently, information about detected objects, including their location and class, are stored in a designated *Class Lookup Table* (CLT). This table will later be retrieved to allocate class categories to 3D mask proposals (*Challenge 4*).

Lookup: Mask Classification through Searching

We conduct a multi-level search to assign category labels to the mask proposals generated from *Mask* step.

Mask2Pixel guided lookup. We introduce a *Mask2Pixel Guided Lookup* (MGL) to search within CLT. The concept involves projecting each 3D mask proposal onto a 2D plane using the same camera extrinsic and intrinsic matrices that are utilized to generate the 2D image, as depicted in Figure 5. With knowledge of the precise pixel locations of each mask in images, we can conduct an accurate search through the CLT to identify the most likely class for each mask. The development of MPM takes into account occlusion by integrating depth information. To accomplish the matching, we follow a three-step approach: 1. based

on the mask's projection onto the 2D plane, we select the best-matched class categories in terms of IoU values; 2. if the IoU value of the best-matched object on the 2D plane is below 20%, the match is disregarded. 3. we aggregate results from multiple views to formulate the final prediction, calculating probability scores using their normalized average IoU values, as illustrated in Figure 6.

Local enforced lookup. While the *Mask2Pixel Guided Lookup* assigns class categories to mask proposals, some masks may not correspond to objects in the CLT. To address this, we introduce a *Local Enforced Lookup* (LEL) approach. We crop out the remaining masks from 2D scene-level images using enlarged bounding boxes and process them with the 2D detector to encourage detection. To select the best views, we introduce an *Occlusion Report* method to assess occlusion conditions for each mask in each projection, and then choose the top K views for LEL. More details for *Occlusion Report* can be found in the supplementary materials.

Final mask refinement. With the previous lookup approaches, a large proportion of mask proposals obtain a category prediction. All masks that have no category predictions after the MGL and LEL stages are eliminated.

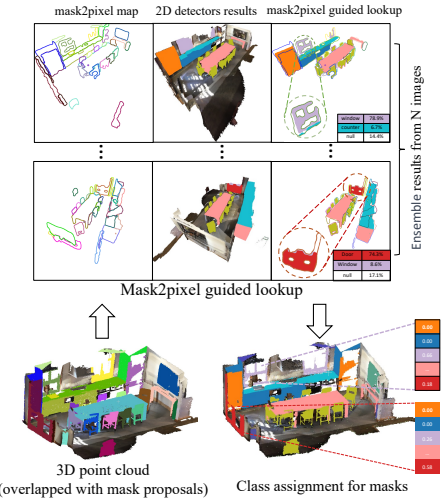


Fig. 6: Mask2Pixel Guided Lookup Illustration. IoUs between the 2D detection results and the projected masks are the guidance to assign class names to 3D masks. Multiple image results are ensembled.

4 Experiments

Datasets and evaluation scheme. We tested OpenIns3D on five datasets, including four indoor datasets, namely S3DIS [2], ScanNetv2 [8], ScanNet200 [30], Repica [33], and one outdoor dataset, STPLS3D [5]. Among them, S3DIS, ScanNetv2, and ScanNet200 are indoor point cloud datasets generated from RGB-D images, Replica is a photo-realistic 3D indoor scene reconstruction, while STPLS3D is an aerial photogrammetry-constructed outdoor dataset. *We exclusively used the 3D data with colour from these datasets and did not employ any 2D images, poses, or depth maps.* Following the settings of prior work [9], we excluded the "other furniture" class in ScanNetv2 and the "clutter" class in S3DIS due to their vague meanings. Replica and ScanNet200 are evaluated by following the settings of OpenMask3D [35]. For STPLS3D, we merged the low, medium, and high vegetation classes into one "vegetation" class and kept all the rest.

Implementation details. For the S3DIS, ScanNetv2, Scannet200, and STPLS datasets, the MPM module is trained without utilizing any category labels, and λ is set to 0.1 to reduce the weight of zero-IoU. The mask proposal of Replica

Table 1: Zero-shot object classification on ScanNetv2. OpenIns3D's Snap and Lookup approach for mask classification, surpasses all previous methods, including the latest language-aligned large-scale 3D foundation model [45].

Method	Avg.	Bed	Cab	Chair	Sofa	Tabl	Door	Wind	Bksf	Pic	Cntr	Desk	Curt	Fridg	Bath	Showr	Toil	Sink
PointCLIP [49]	6.3	0.0	0.0	0.0	0.0	0.7	0.0	0.0	91.8	0.0	0.0	0.0	15.0	0.0	0.0	0.0	0.0	0.0
PointCLIP V2 [42]	11.0	0.0	0.0	23.8	0.0	0.0	0.0	0.0	7.8	0.0	**90.7**	0.0	0.0	0.0	0.0	64.4	0.0	0.0
CLIP2Point [15]	24.9	20.8	0.0	85.1	43.3	26.5	69.9	0.0	20.9	1.7	31.7	27.0	0.0	1.6	46.5	0.0	22.4	25.6
PointCLIP w/ TP.	26.1	0.0	**55.7**	72.8	5.0	5.1	1.7	0.0	**77.2**	0.0	0.0	51.7	0.3	0.0	0.0	40.3	85.3	49.2
CLIP2Point w/ TP.	35.2	11.8	3.0	45.1	27.6	10.5	**61.5**	2.6	71.9	0.3	33.6	29.9	4.7	11.5	**72.2**	92.4	86.1	34.0
$CLIP^2$ [40]	38.5	32.6	67.2	69.3	42.3	18.3	19.1	4.0	62.6	1.4	12.7	52.8	40.1	9.1	59.7	41.0	71.0	45.5
Uni3D [45]	45.8	58.5	3.7	78.8	**83.7**	**54.9**	31.3	39.4	70.1	**35.1**	1.9	27.3	**94.2**	13.8	38.7	10.7	88.1	47.6
OpenIns3D	**60.8**	**85.2**	27.4	**87.6**	77.3	46.9	54.8	**64.2**	71.4	9.9	80.8	**82.7**	71.6	**61.4**	38.7	0.0	**87.9**	**85.7**

is trained on ScanNet200, followed by OpenMask3D [35]. The Snap module captures images with a size of 1000×1000 including 16 global snaps, 4 corner snaps, and 4 wide-angle snaps. The top 0.5 m of the scene is removed for S3DIS, as the rooms are enclosed. For STPLIS3D, we followed Mask3D to split the large outdoor scene into patches of $50m \times 50m$ and lifted the camera up to 10m. More implementation details are presented in the supplementary materials.

Open-vocabulary point cloud recognition. We first evaluate OpenIns3D performance among all existing 3D open-vocabulary models with the same setting, i.e. **using only 3D inputs**. These models are most commonly tested on recognition tasks, including PointCLIP(v1&v2) [49] [42], Clip2Point [15], $CLIP^2$ [40]. We also compare with large-scale 3D foundation models, such as Uni3D [45], which has 1-billion parameters, and was trained with large-scale 3D shapes and image-text pairs. We followed their evaluation scheme and reported the Top-1 accuracy of instance classification on ScanNetv2. The results are presented in Table 1.

Open-vocabulary instance segmentation. We adopted various comparison schemes to align with existing methods. For 3D Open-vocabulary Instance Segmentation, we compared with PLA [9], and its follow-up works RegionPLC [39] and Lowis3D [10]. For a fair comparison, we followed their category splits and compared our results on novel classes, as demonstrated in Table 2. For STPLS3D, we compared OpenIns3D with baseline models whose classification module is PointCLIP and PointCLIPV2 [49] (Table 5). We also explored the performance of OpenIns3D on a more challenging dataset with more class categories. Specifically, we compared the performance of OpenIns3D with OpenMask3D, OpenScene, on Replica (Table 4) as well as ScanNet200 (Table 6). Following OpenMask3D, we used Mask3D to generate mask proposals for OpenScene for evaluation.

Open-vocabulary object detection. Since there were limited works on 3D open-vocabulary instance segmentation at the time of conducting this work, we also selected some of the latest methods in the 3D open-world object detection domain for a more comprehensive evaluation. The evaluation is performed by converting generated masks into axis-aligned bounding boxes, and the results are shown in Table 3.

Table 2: 3D Open-vocabulary Instance Segmentation Results on S3DIS and ScanNetv2. We compare our zero-shot performance on the novel categories defined in the PLA-family work. Significant improvements are achieved on the S3DIS dataset, and competitive results are observed on ScanNetv2 (B/N: Base/Novel).

OVIS	S3DIS			ScanNetv2			
Method	B/N	AP50	AP25	B/N	AP50	AP25	require 2D
PLA [9]	8/4	08.6	-	10/7	21.9	-	✓
RegionPLC [39]	8/4	-	-	10/7	32.3	-	✓
Lowis3D [10]	8/4	13.8	-	10/7	31.2	-	✓
Open3DIS [25]	8/4	26.3	-	10/7	-	-	✓
Mask3d+PointClip [49]	–/4	05.4	10.3	–/7	04.5	07.8	✗
OpenIns3D	–/4	**37.0**	**39.3**	–/7	27.9	**42.6**	✗
improvement	–/4	(+10.7)	(+29.0)	–/7	–	(+34.8)	✗
PLA [9]	6/6	09.8	-	8/9	25.1	-	✓
RegionPLC [39]	6/6	-	-	8/9	32.2	-	✓
Lowis3D [10]	6/6	15.8	-	8/9	38.1	-	✓
Open3DIS [25]	6/6	29.0	-	8/9	-	-	✓
Mask3d+PointClip [49]	–/6	08.5	10.6	–/9	05.6	06.7	✗
OpenIns3D	–/6	**33.0**	**38.9**	–/9	19.5	**27.9**	✗
improvement	–/6	(+4.0)	(+28.3)	–/9	–	(+21.2)	✗
Mask3d+PointClip [49]	–/12	08.6	09.3	–/17	04.5	14.4	✗
OpenIns3D	–/12	**28.3**	**29.5**	–/17	**28.7**	**38.9**	✗
improvement	–/12	(+19.7)	(+20.2)	–/17	(+24.2)	(+24.5)	✗

Table 3: Open-vocabulary Object Detection (AP_{25}) on unseen classes in ScanNet.

Methods	mean	toilet	bed	chair	sofa	dresser	table	cabinet	bookshelf	pillow	sink
OV-PointCLIP [42]	3.1	6.6	2.3	6.3	3.9	0.7	7.2	0.7	2.1	0.6	0.8
OV-Image2Point [37]	0.8	0.2	0.8	1.0	1.4	0.2	2.8	1.0	0.9	0.0	0.1
Detic-ModelNet [46]	1.7	4.2	1.0	4.6	1.2	0.2	3.2	0.6	1.2	0.0	0.7
Detic-ImageNet [46]	0.4	0.0	0.0	0.2	0.0	0.5	1.8	0.5	0.3	0.0	0.7
OV-3DETIC [22]	12.7	49.0	2.6	7.3	18.6	2.8	14.3	2.4	4.5	3.9	21.1
L3Det [48]	24.6	56.3	36.2	16.1	23.0	8.1	23.1	14.7	17.3	23.4	27.9
FM-OV3D [41]	21.5	55.0	38.8	19.2	41.9	23.8	3.5	0.4	6.0	17.4	8.8
OpenIns3D	**43.7**	**79.5**	**70.5**	**76.9**	15.8	0.0	**53.1**	**40.1**	**41.2**	7.1	**53.1**
improvement	(+19.1)	(+23.2)	(+31.7)	(+57.7)	–	–	(+30)	(+25.4)	(+23.9)		(+25.2)

5 Results and Discussion

5.1 Comparison with SOTA

First of all, OpenIns3D demonstrates impressive performance in open-vocabulary point cloud recognition, surpassing all previous methods, including the large-scale 3D foundation model by 15%. This proves the effectiveness of the zero-shot "Snap" and "lookup" scheme. With the enhanced recognition capability, the performance of 3D open-vocabulary Object Detection among the ScanNet dataset has also achieved state-of-the-art results by a large margin. For 3D instance segmentation, compared to works in the PLA family [9, 10, 39] and the latest work Open3DIS [25], OpenIns3D does not require aligned images as input, (still need RGB information for points), achieves higher results on the S3DIS dataset, both in the 4 novel categories split and the 6 novel categories split. In SPTLS3D, OpenIns3D outperforms the baseline model PointCLIPV2 by 9.3 % in AP. On

Table 4: OVIS on the indoor Replica dataset.

Model	2D	AP	AP_{50}	AP_{25}
OpenScene [26] (2D Fusion)	✓	10.9	15.6	17.3
OpenScene [26] (2D/3D Ens.)	✓	8.2	10.4	13.3
OpenMask3D [35]	✓	13.1	18.4	24.2
OpenScene [26] (3D Distill)	✗	8.2	10.5	12.6
OpenIns3D	✗	**13.6**	**18.0**	**19.7**
improvement	✗	(+5.4)	(+7.5)	(+7.1)

Table 5: OVIS on outdoor STPLS3D dataset

Model	AP	AP_{50}	AP_{25}
PointCLIP [49]	02.0	02.6	04.0
PointCLIPv2 [42]	02.1	03.1	05.2
OpenIns3D	**11.4**	**14.2**	**17.2**
improvement	(+9.3)	(+11.9)	(+12.0)

Table 6: 3D instance segmentation results on the ScanNet200 validation set. OpenIns3D demonstrates robust performance when compared to 2D-input-free models. However, notable limitations emerge when dealing with small objects in the common and tail classes.

Model	use 2D	AP_{head}	AP_{common}	AP_{tail}	AP	AP_{50}	AP_{25}
OpenScene (2D Fusion) [26]	✓	13.4	11.6	9.9	11.7	15.2	17.8
OpenScene (2D/3D Ens.) [26]	✓	11.0	3.2	1.1	5.3	6.7	8.1
OpenMask3D	✓	17.1	14.1	14.9	15.4	19.9	23.1
OpenScene (3D Distill)	✗	10.6	2.6	0.7	4.8	6.2	7.2
OpenIns3D	✗	**16.0**	**6.5**	**4.2**	**8.8**	**10.3**	**14.4**
improvement	✗	(+5.4)	(+3.9)	(+3.6)	(+4.0)	(+4.1)	(+7.2)

the Replica dataset, OpenIns3D even outperformed OpenMask3D, which relies on well-aligned images for mask understanding. In the case of ScanNet200, OpenIns3D attains the highest performance compared to all other 3D input baselines. However, we have observed a decline in performance on tail and common classes within ScanNet200. The decrease in performance can be attributed to the low-quality and unclear reconstruction of smaller objects in the ScanNet200 scene. This is a noticeable limitation of OpenIns3D, but for better reconstructions, such as in Replica, and for Head classes within ScanNet200, OpenIns3D still demonstrates decent performance.

In summary, OpenIns3D demonstrates the best performance among all existing methods if only 3D data is used as input and outperforms many existing state-of-the-art methods that require 2D images. It also shows certain limitations on small objects that are not well-reconstructed in 3D scenes.

5.2 Ablation study

Mask quality ablation. Following the evaluation on ScanNetv2, we assessed the class-agnostic mask quality using the average precision (AP) score. We treated all classes as universal since the predictions are class-agnostic. The evaluation was conducted on the ScanNetv2 validation set. Table 8 demonstrates the effectiveness of the *Mask Scoring* and *Mask Filtering* designs.

Table 7: Rendering and Inference Time Ablations. Results tested on typical ScanNet scenes with 50 masks. OpenIns3D requires less rendering and inference time.

Rendering	Num of Img needed	2D backbone	Img size (w × h)	T_{render} (s/scene)	T_{infer} (s/scene)	T_{total} (s/scene)	AP25 (%)
PointCLIP	250	CLIP	128^2	5.2	15.3	20.5	9.3
LAR	250	CLIP	128^2	14.3	18.7	33.0	10.5
Mask rendering	250	CLIP	128^2	42.6	19.5	62.1	7.3
OpenIns3D (ours)	8	G-DINO	1000^2	2.3	6.2	8.5	29.8
OpenIns3D (ours)	8	ODISE	1000^2	2.3	8.2	10.5	35.1

Table 8: MPM Ablation. MS: Mask scoring. MF: Mask filtering.

Method	AP50	AP25
Mask3d-Supervised	74.7	80.9
CA-Mask3d	47.5	49.2
CA-Mask3d + MS	50.2 (+02.7)	53.3 (+04.1)
CA-Mask3d + MF	61.6 (+14.1)	71.0 (+21.8)
CA-Mask3d + MS + MF	64.6 (+17.0)	73.4 (+24.2)

Table 9: Number of Views Ablation. LEL: Local Enforced Lookup

IDX	4	8	16	LEL	AP50	AP25
i	✓				18.3	27.1
ii		✓			22.7	35.1
iii			✓		24.8	37.5
iv			✓	✓	28.7	38.9

Multi-view ablation. We also studied the effects of using different numbers of views (Table 9). Increasing the number of views used in the *Lookup* module leads to better results. Additionally, Look Enforced Lookup provided a final improvement to the results.

Projection and 2D backbone ablation. We conducted a comprehensive study on various rendering methods and their interaction with the 2D backbone to identify a suitable approach. We report the rendering time and inference performance for each method (Table 7), and more details can be found in the supplementary materials. The key observation is that the scene-level rendering and understanding approach excels in speed while also demonstrating strong performance. Switching from Grounding Dino [21] to the latest ODISE [38] also brings gains in performance, indicating that the OpenIns3D framework can easily benefit from the rapid development of 2D open-world detectors.

Cross-domain analysis. To evaluate the generalization capability of MPM across different domains, we trained and tested OpenIns3D on two different datasets, as shown in Table 10. The cross-domain models also demonstrate impressive performance on both datasets when compared with the baseline. Notably, within the 17 classes of ScanNetv2, 11 classes do not exist in S3DIS. OpenIns3D, trained on S3DIS, still achieves decent performance among these unseen classes.

Free-flow language capability. The snap and lookup scheme outsources the mask understanding tasks to a 2D vision-language model. Therefore when integrated with a 2D model powered by large language models (LLMs) like LISA [19], OpenIns3D can perform reasoning-based segmentation tasks (as depicted in Figure 7). For instance, when given the query "pencilling down ideas during brain-

storming," OpenIns3D accurately segments the whiteboard, while for "furniture offers recreational enjoyment with friends," it precisely identifies and segments the pool table.

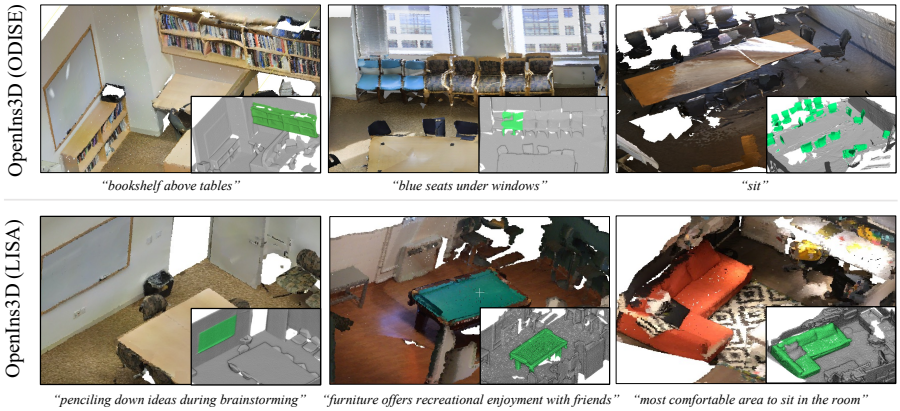


Fig. 7: Qualitative Results from OpenIns3D. OpenIns3D (ODISE) demonstrates the ability to manage a versatile vocabulary. OpenIns3D (LISA) can conduct 3D reasoning segmentation.

Table 10: Cross-domain Ablation. We trained and tested OpenIns3D on two different datasets to examine its cross-domain capability. While S3DIS and ScanNetV2 have non-overlapping classes, OpenIns3D demonstrates decent generalization capability.

Test	Model	Training Data	AP50	AP25
ScanNetv2 [8]	Mask3D-Pointclip [49]	ScanNetv2	04.5	14.4
	OpenIns3D	ScanNetv2	28.7	38.9
	OpenIns3D	S3DIS	<u>21.5</u>	<u>33.6</u>
S3DIS [1]	Mask3D-Pointclip [49]	S3DIS	03.5	06.8
	OpenIns3D	S3DIS	28.3	29.5
	OpenIns3D	ScanNetv2	<u>14.2</u>	<u>19.8</u>

6 Conclusion

Achieving 3D open-vocabulary scene understanding is a challenging task, primarily due to the lack of extensive 3D-text data. Currently, most work in this domain focuses on using 2D images to bridge the gap between 3D and language. This, however, not only requires a good alignment between 2D and 3D but also evolves slowly due to the significant effort needed for retraining when changing the 2D backbone. OpenIns3D introduces a new pipeline, i.e. Mask-Snap-Lookup, for this task. The **Mask** module generates authentic masks in the 3D domain, while **Snap** renders scene-level images in 2D domains, and the **Lookup** module links the results from 2D to 3D precisely. This pipeline requires no image input, achieves better performance, and can evolve seamlessly with 2D models without training. We hope our work will provide a fresh perspective for researchers working towards open-world 3D scene understanding.

Acknowledgments

This work is supported by the Girton College Graduate Research Awards at the University of Cambridge, School of Technology, National Highways sponsored through EPSRC Centre for Doctoral Training, the InnoHK funding launched by Innovation and Technology Commission, Hong Kong SAR, the National Natural Science Foundation of China No. 62201484, HKU Startup Fund, and HKU Seed Fund for Basic Research. We would like to express our gratitude to Yunhan Yang for his assistance in exploring rendering techniques and to Chengyao Wang for sharing his implementation of Mask3D.

Appendix

In a nutshell, OpenIns3D is a new pipeline for 3D open-world scene understanding that consumes only 3D coloured point clouds as input, making it easier to deploy in a wide range of scenarios. We also present detailed per-category results on how OpenIns3D performs on the ScanNetv2 [8], S3DIS [1], and STPLS3D [5] datasets for both Instance Segmentation and Object Detection, for future work to compare with. In the development process, we tested a wide range of rendering methods and documented their different performances, which demonstrates how scene-level rendering stands out from other rendering methods. More detailed implementation details and methodologies of OpenIns3D are also presented. The section structure is listed as follows:

- Section A: More Details on Methodologies
- Section B: Implementation Details
- Section C: Per-Categories Results
- Section D: Other Attempts for Image Rendering
- Section E: Limitation and future work
- Section F: More Visualization

A More Details on Methodologies

Class-agnostic mask proposal module. We modified modules that require classification labels in Mask3D [32] to make it a class-agnostic setting. This includes 1. removing semantic probability components in Hungarian Matching, 2. eliminating semantic classification loss, 3. discarding classification logits-based ranking, and 4. getting rid of classification logits-based filtering. Instead, we added the *Mask Scoring* and *Mask Filtering* module to acquire high-quality mask proposals.

Local enforced lookup. Here, we provide a detailed explanation of the *Occlusion Report* module that we proposed to effectively evaluate the occlusion condition of masks in all synthetic images. Specifically, the following four steps are executed:

- **Step 1. Point Count Array:** We initiate the process by constructing a 3D array with dimensions $W \times H \times (M+1)$, where M represents the number of masks, and $+1$ is for the background points. This array will be denoted as PC, *i.e.* point count, as it is designed to store the number of points of the 3D mask projected onto each pixel in the images. For example, if the pixel at coordinates i, j is occupied by two points from the 3D mask k during the projection, $PT_{i,j,k}$ will be assigned the value 2.
- **Step 2. Foremost Point Identification:** Utilizing the depth map generated during the projection process, we construct a 2D array named FP with dimensions $W \times H$, which is used to identify the foremost point in each pixel and indicate the originating mask number. For example, if pixel i,j's foremost point is projected from Mask k, we denote $FP_{i,j} = k$.
- **Step 3. Occlusion Rate Calculation:** To evaluate the occlusion rate (OR) for mask k within specific images, we compute the following formula:

$$OR_k = \frac{\sum_{i=1}^{W} \sum_{j=1}^{H} PC_{i,j,k} \cdot (FP_{i,j} = k)}{T_k}$$

  where T represent the total number of point in mask k.
- **Step 4. All Images Report:** Finally, we repeat steps 1-3 for all images to obtain an overall report of the occlusion rate of each mask across all images, forming the final *Occlusion Report*.

After selecting the best view with *Occlusion Report*, synthetic scene-level images are cropped to focus on a specific mask proposal and then reprocessed by 2D detectors. The results are also searched with the help of Mask2Pixel, in this case the binary mask, maps to form the final classification prediction for the mask, as shown in Figure 8.

B Implementation Details

Zero-shot Object Recognition

Zero-shot object recognition results are obtained by employing the "Snap and Lookup" modules to assign category names to the ground truth masks. The Snap module takes 24 images, and the Lookup module uses ODISE [38] to extract potentially interesting objects and assign labels to the masks. For masks that are not assigned a label or are assigned the wrong label, the top-1 classification is marked as a false negative.

Most other methods in this comparison use an object-centred rendering approach where a depth map or point cloud is projected into images for classification. Uni3D [45] is pre-trained on a large amount of image and text pairs, as well as 3D shapes. OpenIns3D's scene-level image rendering, with enhanced results, proves to be much more effective in object recognition tasks.

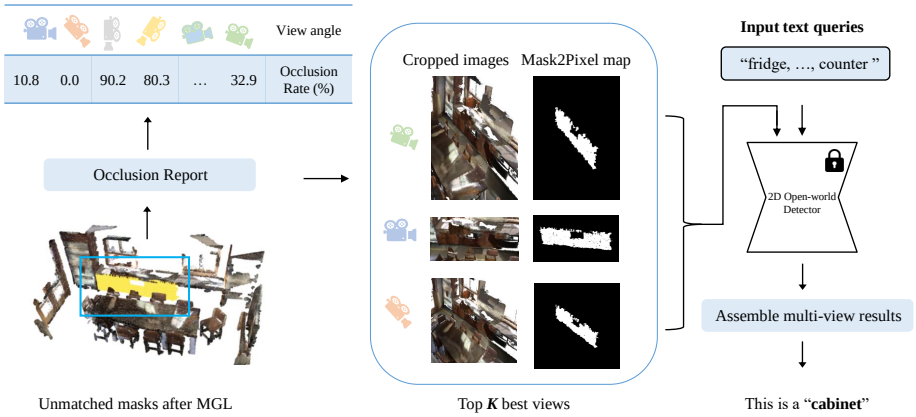


Fig. 8: Illustration of Local Enforced Lookup. The remaining masks from phase one first go through the *Occlusion Report* module to select the best K views. The selected images are cropped before being processed by the 2D detectors to encourage a classification result.

Open-Vocabulary Instance Segmentation

Mask. The Mask Proposal Module is built upon a lightweight version of Mask3D [32] with three decoder layers. For the mask quality scoring module, we set λ to 0.1 to down-weight zero IOU masks. The Mask Proposal Module is trained using the ADAM optimizer with a learning rate of 0.0003, and the one-cycle scheduler is applied. For the ScanNetv2, S3DIS, and STPLS3D experiments, the mask proposal module is trained on all-category class-agnostic masks to learn to propose masks. For Replica and ScanNet200, we followed OpenMask3D [35] and used ScanNet200 pre-trained weights for mask proposals.

Snap. We captured 24 images of the scene, including 16 global, 4 corner, and 4 wide-angle images. We used the PyTorch3D Rasterization Renderer to render images. For all datasets, we captured images with dimensions of 1000 x 1000 for a great trade-off between speed and performance. Additionally, to avoid the occlusion effect caused by the ceiling, we discarded the top 0.3m points in the S3DIS and ScanNet datasets. As a result, the ceiling categories in the S3DIS dataset are completely discarded and assigned 0 in the AP results. For STPLS3D, the camera position is located 5m higher than the top of the scene to acquire a better view.

Lookup. During the *Lookup* stage, we only assign a classification label to each mask if the results have been verified in at least two views. In the case of *Local Enforced Lookup*, we crop the images using bounding boxes that are twice the size of the target masks. The cropped images are then fed into 2D detectors to refine the results. Mask2Pixel maps, in this case, binary maps, are used to accurately search for the detection results, as shown in Figure 8.

Open-vocabulary Object Detection

Open-vocabulary object detection is carried out by converting the mask proposals into axis-aligned bounding boxes. We followed the same Snap and Lookup implementation details in Open-vocabulary Instance Segmentation Setting for bounding box understanding.

C Detailed Results

Open-vocabulary Instance Segmentation

Tables 11 and 12 provide the per-class results of OpenIns3D on the S3DIS and ScanNetv2 datasets. The novel (unseen) classes in PLA are highlighted in blue. Table 13 represents the per-category results for the STPLS3D dataset, compared with PointCLIP and PointCLIPV2.

In **S3DIS**, OpenIns3D consistently achieves high results in these novel classes. We attribute this to the high quality of 3D point data in S3DIS, which ensures favorable conditions for object detection in 2D Snap images. However, for classes like columns and beams, OpenIns3D struggles to produce desirable results. Ceiling results are marked as 0, as explained in the implementation detail section.

In **ScanNetv2**, the point cloud data quality is not very high. Consequently, the Snap output quality is limited, resulting in slightly lower performance.

In **STPLS3D**, OpenIns3D outperforms PointCLIP and PointCLIPV2 by a significant margin. This is as expected, as already demonstrated in the zero-shot classification tasks. However, the performance on very small objects, such as bikes, motorbikes, signs, and light poles, is not as strong. This is because the Snap module positions the camera at a high angle, resulting in a limited number of pixels available for these smaller objects.

Open-vocabulary Object Detection.

We also present the object detection results at Table 14 for all categories, followed by the OV-3DETIC categories. OpenIns3D shows strong performance across most categories.

Cross-domain analysis.

Table 15 presents the per-category results for the cross-domain OpenIns3D model, trained on S3DIS and tested on ScanNetv2. This table comparison is conducted to demonstrate the generalization capability of the mask module.

D Other Attempts for Image Generation

Figures 9 and Table 16 illustrate the alternative 2D image rendering approaches we explored before concluding that synthetic scene-level images offer the optimal solution. We document the process here for future reference.

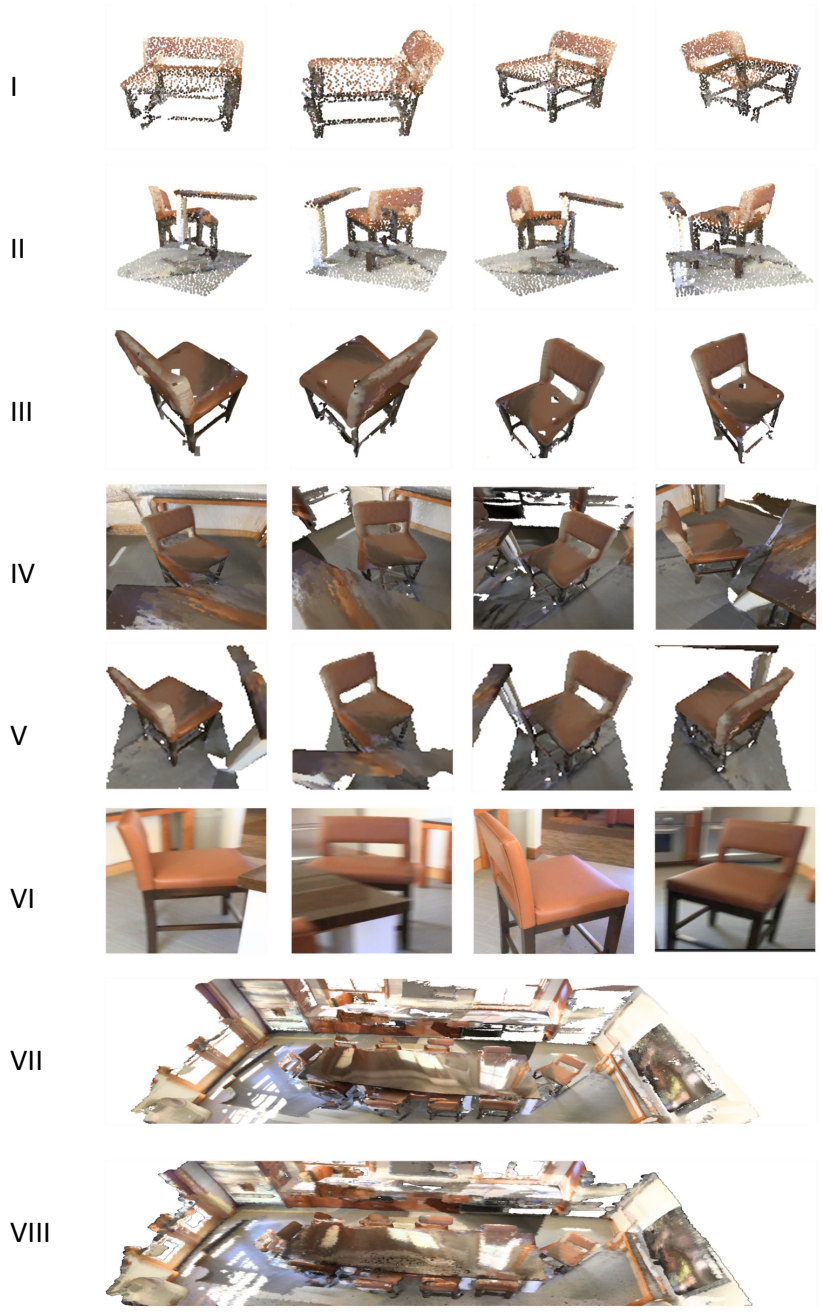


Fig. 9: Visualization of Attempts Made to Generate 2D Images from 3D. I: LAR-point projection; II: LAR-point-bg-project; III: Mesh rendering; IV Mesh-in-scene Rendering; V: Mesh-bg-Rendering; VI: Cropped from Original 2D images; VII: Scene Level Rendering from Mesh; VIII: Scene Level Rendering from Point. Performance can be found in Table 16.

Table 11: Per-class Results of 3D Open-vocabulary Instance Segmentation on S3DIS AP50. Performance on novel classes is marked in blue.

Methods	Partition	ceiling	floor	wall	beam	column	window	door	table	chair	sofa	bookcase	board
PLA [9]	B8/N4	89.5	100.0	50.8	00.0	35.3	36.2	60.5	00.1	84.6	01.9	00.8	59.4
	B6/N6	89.5	60.2	17.9	00.0	41.5	10.2	02.1	00.6	86.2	45.1	00.1	02.2
OpenIns3D	–/N12	00.0	84.4	29.0	00.0	00.0	62.6	25.2	25.5	52.0	60.0	00.0	00.0

Table 12: Per-class Results of 3D Open-vocabulary Instance Segmentation on ScanNet AP50. Performance on novel classes is marked in blue.

Methods	Partition	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	fridge	shower c.	toilet	sink	bathtub
PLA [9]	B13/N4	50.5	77.0	82.9	43.4	75.4	49.0	46.0	43.7	46.5	33.7	23.2	54.1	49.6	56.0	97.8	47.5	85.8
	B10/N7	53.7	62.7	11.2	70.5	27.2	47.7	45.7	30.0	01.5	39.9	40.8	50.6	68.6	84.6	92.9	24.6	00.0
	B8/N9	45.1	77.4	82.2	84.2	74.2	48.9	51.0	30.0	00.5	02.1	16.8	44.9	28.3	35.1	94.3	16.6	00.0
OpenIns3D	–/N17	24.3	52.5	75.7	61.6	40.6	39.7	45.5	54.8	0.5	33.5	16.7	48.1	18.5	4.3	50.1	16.8	7.6

Attempts I, II: Inspired by the success of LAR [3], we positioned the camera around the object and projected point clouds to generate multi-view images for each mask. However, these approaches produced images beyond the recognition capability of the CLIP model, especially for masks that were not well-segmented or reconstructed, leading to unsatisfactory results.

Attempts III, IV, V: We redirected our attention to the mesh of the scene, using rasterization rendering methods rather than simple point projecting. Although these methods brought some improvement, they still proved to have strong limitations when the mask proposed was not perfect. Masks of undesired quality made up a large portion of the masks, making this an inadequate solution. Moreover, per-mask rendering required a significant amount of time, making it impractical for deployment.

Attempt VI: We then experimented with using original images and cropping out masks in the images for evaluation (VI). We believed this would offer the best quality of images, making them most likely to be recognizable with 2D models. We used *Occlusion Reports* methods to select the top K views from all frames and crop out mask pixels with an enlarged bounding box. This approach achieved notable performance, primarily due to the high quality of 2D images. However, we ultimately abandoned this approach due to concerns about its applicability in general scenarios.

Attempts VII, VIII: Shifting our focus to scene-level rendering, our model began to produce high-quality results. By observing all broken instances from a distance and incorporating a large amount of contextual information, objects became clear and recognizable. As a result, the scene-level images had a small domain gap with the images used to train 2D Vision-Language models.

Table 13: Per-class Results of 3D Open-vocabulary Instance Segmentation on STPLS3D AP50. All models are tested in a zero-shot manner.

Methods	mean	building	veg	vehicle	truck	aircraft	mil-veh	bike	motorbike	lightpole	signs	clutter	fence
PointCLIP [49]	2.7	15.3	0.4	10.2	06.6	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0
PointCLIPV2 [42]	3.2	20.3	0.2	12.3	5.8	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0
OpenIns3D	**14.1**	**40.4**	**01.2**	**54.2**	**24.2**	**30.0**	**05.5**	**02.1**	**03.0**	00.0	00.0	00.0	**08.3**

Table 14: Detailed results on 3D open vocabulary object detection. We present all results on ScanNet20, followed by OV-3DETIC [22].

Methods	**mean**	toilet	bed	chair	sofa	dresser	table	cabinet	bookshelf	pillow	sink	bathtub	fridge	desk	nightstand	counter	door	curtain	box	lamp	bag
3DETIC [47]	15.0	53.3	24.9	15.8	31.4	**11.5**	09.1	02.1	09.4	17.0	29.2	27.5	20.0	13.7	00.0	00.0	00.0	17.7	04.8	03.0	**09.5**
CLIP-3D [28]	12.7	44.8	23.8	17.5	12.6	04.9	13.2	01.9	04.0	11.4	17.6	32.2	14.9	11.4	02.4	00.5	14.5	08.6	**07.5**	05.1	04.7
OV-3DETIC [22]	18.8	57.3	42.3	27.1	**31.5**	08.2	14.2	03.0	05.6	**23.0**	31.6	**56.3**	11.0	19.7	00.8	00.3	09.6	10.5	03.8	02.1	02.7
OpenIns3D	**37.1**	**79.5**	**70.5**	**76.9**	15.8	00.0	**53.1**	**40.1**	**41.2**	07.1	**53.1**	14.3	**32.1**	**29.1**	**04.8**	**55.6**	**40.4**	**41.1**	02.6	**48.0**	06.2

Table 15: Cross-domain Analysis of OpenIns3D on OVOD on ScanNetv2 AP25. OpenIns3D achieves competitive results on the cross-domain dataset, even on categories at are not available on the training dataset, highlighted in blue. Compared with other SOTA models on OVOD, cross-domain OpenIns3D still has competitive performance. MPM-SC: MPM trained on ScanNetv2; MPM-S3: MPM trained on S3DIS.

Methods	BBox Prop	cabinet	bed	chair	sofa	table	door	window	bookshelf	picture	counter	desk	curtain	fridge	shower c.	toilet	sink	bathtub
OpenIns3D	MPM-SC	17.1	57.5	74.5	59.2	36.9	29.3	47.5	26.4	0.0	31.1	32.2	55.4	39.1	0.0	57.4	42.1	6.6
OpenIns3D	MPM-S3	16.1	43.5	45.7	41.8	28.6	17.7	18.3	31.9	1.2	1.0	29.3	23.1	20.1	8.0	63.6	16.4	1.7
SOTA models																		
PointCLIP	P-3DE	6.0	4.8	45.2	4.8	7.4	4.6	2.2	-	-	1.0	4.0	-	-	-	-	13.4	6.5
PointCLIPV2	P-3DE	19.3	21.0	61.9	15.6	23.8	13.2	17.4	-	-	12.4	21.4	-	-	-	-	14.5	16.8
OV-3DET	P-2DE	3.0	42.3	27.1	31.5	14.2	9.6	-	5.6	-	0.3	19.7	10.5	11.0	-	57.3	31.6	56.3

E Limitations and Future Work

There are some limitations of OpenIns3D that need further investigation in future studies.

– Reliance on ground truth instance masks: Similar to SAM [17], OpenIns3D still relies on ground truth mask supervision. While it does prove to have the capability to generalize masks that have never been seen before, providing a vast amount of class-agnostic masks can be helpful. Approaches like UnScene3D [31], Segment3D [14] might serve as alternative methods for mask proposal, linking it with *Snap* and the *Lookup* module for open-vocabulary understanding. This requires further investigation.

Table 16: Evolution of *Snap* and *Lookup* Module. The corresponding image visualization is shown in Figure 9. Scene-level rendering not only requires fewer images but also achieves superb results when compared to other pre-mask levels of rendering. *: The image sizes of VI are not fixed as it depends on the size of the mask area on the original images.

Idx	Methods	Job intensity	Imgs needed	use 2D	Img size	2D backbone	AP50	AP25
I	LAR-point projection	per mask	250	✗	128^2	CLIP	5.3	8.6
II	LAR-point-bg-projection	per mask	250	✗	128^2	CLIP	6.3	10.5
III	mesh-rendering	per mask	250	✗	128^2	CLIP	6.8	7.2
IV	mesh-scene-rendering	per mask	250	✗	128^2	CLIP	6.7	7.3
V	mesh-bg-rendering	per mask	250	✗	128^2	CLIP	4.3	5.3
VI	crop-original2d	per mask	250	✓	$-^*$	CLIP	24.3	29.6
VII	scene-mesh-rendering	per scene	8	✗	1000^2	ODISE	28.7	38.9
VIII	scene-point-rendering	per scene	8	✗	1000^2	ODISE	21.5	33.6

Table 17: Comparison with OpenScene and other Frameworks on Semantic Segmentation. Our framework prioritises mask quality and sacrifice overall semantic segmentation results.

Semantic Seg.	mIoU					mAcc				
Methods	Bookshelf	Desk	Sofa	Toilet	Mean	Bookshelf	Desk	Sofa	Toilet	Mean
3DGenZ [23]	6.3	3.3	13.1	8.1	7.7	13.4	5.9	5.9	26.3	12.9
MSeg Voting [20]	47.8	40.3	56.5	68.8	53.3	50.1	67.7	67.7	81.0	66.6
OpenScene-LSeg [26]	**67.1**	**46.4**	60.2	**77.5**	**62.8**	**85.5**	**69.5**	69.5	90.0	78.6
OpenScene-OpenSeg [26]	64.1	27.4	49.6	63.7	51.2	73.7	73.4	73.4	95.3	79.0
OpenIns3D	54.8	16.7	**61.6**	50.6	45.9	59.0	32.3	**76.7**	79.8	61.9

– Limited performance in semantic segmentation: OpenIns3D heavily relies on filtering to refine the mask proposals, discarding masks with low quality directly. While this approach benefits instance segmentation by reducing false positive instances, it may limit its performance in semantic segmentation. We have also calculated the semantic segmentation results of OpenIns3D on four categories, as reported by OpenScene [26], as shown in Table 17. Our method still exhibits a gap compared to OpenScene in terms of semantic segmentation.
– Small object performance: The performance of OpenIns3D is ultimately closely linked to the quality of the point cloud itself. Masks that are very small or made of sparse point clouds would be difficult to recognize in the rendered images, as they either occupy a small portion of the image pixels or are too fragmented to be detected by the 2D models.

F Visualization

Zero-shot performance on other dataset. OpenIns3D is able of deploying on any colored 3D scans, regardless of the availability of 2D image counterparts. To demonstrate this, we provide some demos deploying OpenIns3D on Lidar-based datasets like ArkitScene Lidar, whose 2D images are not available, and

Mattport3D, which has a different style of indoor rooms. The results are shown in Figure 11 and Figure 10. We present both mask proposals (pre-trained on ScanNet 200) as well as the final detection results.

Mask proposal. Figure 12 and 13 present a qualitative evaluation of the mask proposal module. The learned mask proposals exhibit great similarity to the ground truth masks, often capturing additional unlabeled masks. This demonstrates the effectiveness of our class-label-free learning scheme in producing high-quality class-agnostic mask proposals. Moreover, through the application of *Mask Scoring* and *Mask Filtering* techniques, we are able to connect fragmented or fragile masks, resulting in a substantial improvement in mask quality. These advancements provide a strong foundation for the Snap and Lookup understanding scheme.

Snap visualization. Figure 14, 15 and 16 demonstrate the capability of the Snap module, we present mostly the global snap. With the proposed pose and intrinsic optimization scheme, the Snap module is capable of generating decent-quality images from point clouds, regardless of whether the dataset is indoor or outdoor.

Lookup results visualization. The Lookup module effectively links 2D results with 3D. Here, we present visualizations of its outcomes from all three datasets (Figure 17, 18, 19).

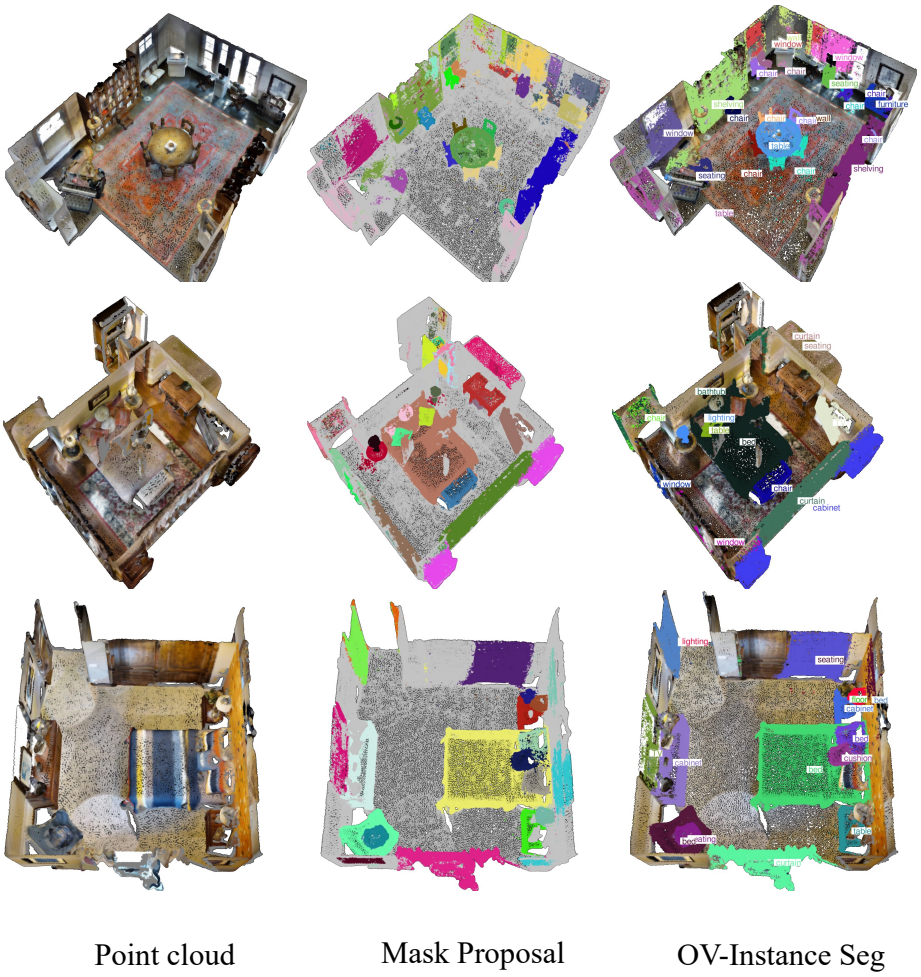


Fig. 10: Zero-shot Open world Instance Segmentation on Matterport3D.

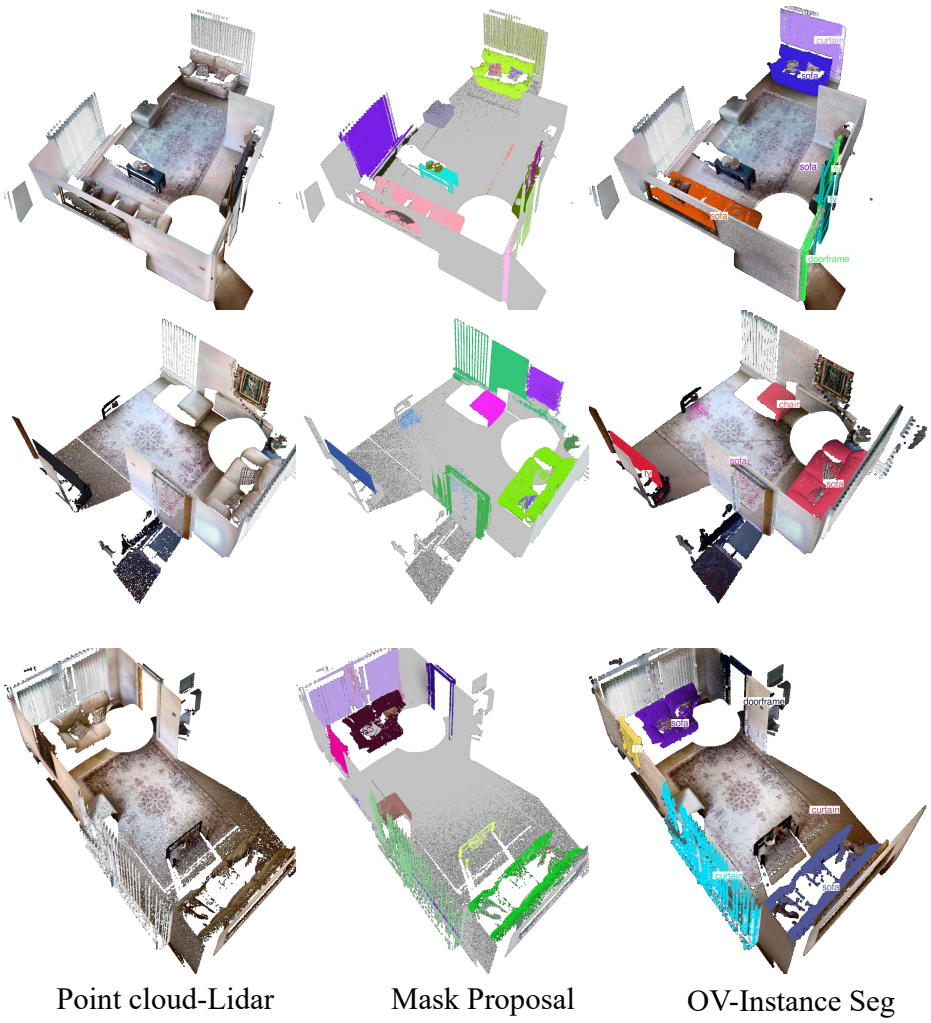


Fig. 11: Zero-shot Open world Instance Segmentation on ArkitScene-Lidar.

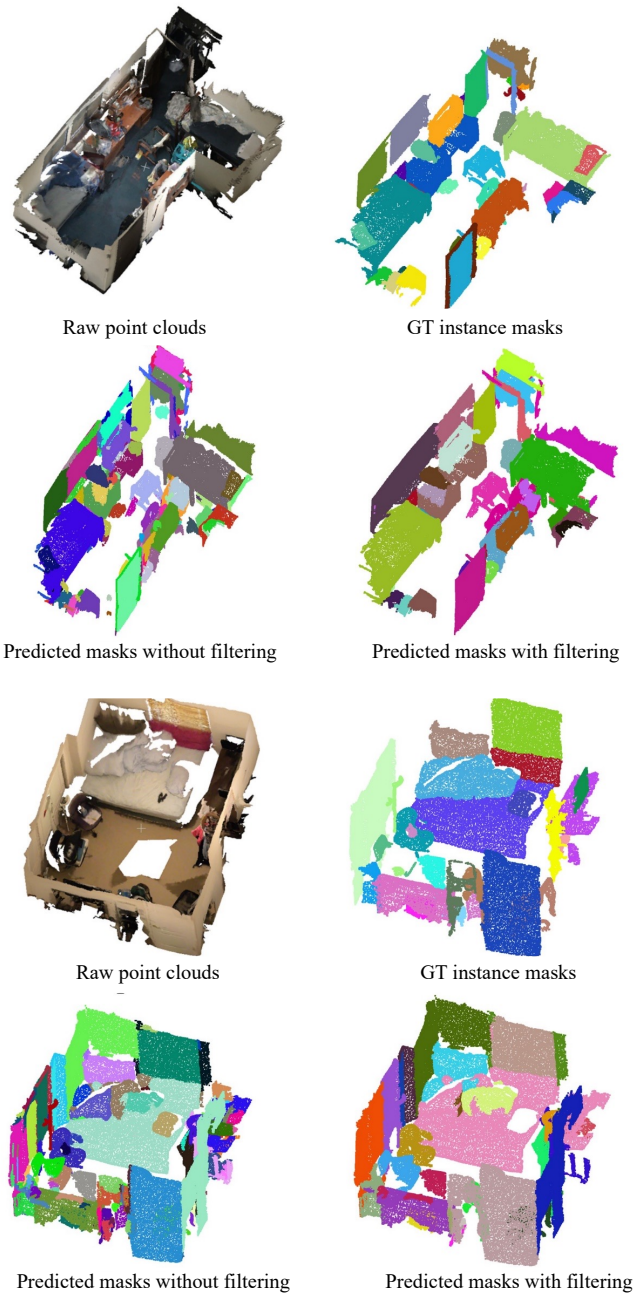


Fig. 12: Qualitative Evaluation of the Mask Proposals. Our class-label-free approach produces high-quality masks that closely resemble the ground truth. Additionally, the incorporation of *Mask Scoring* and *Mask Filtering* further enhances the overall quality of the masks.

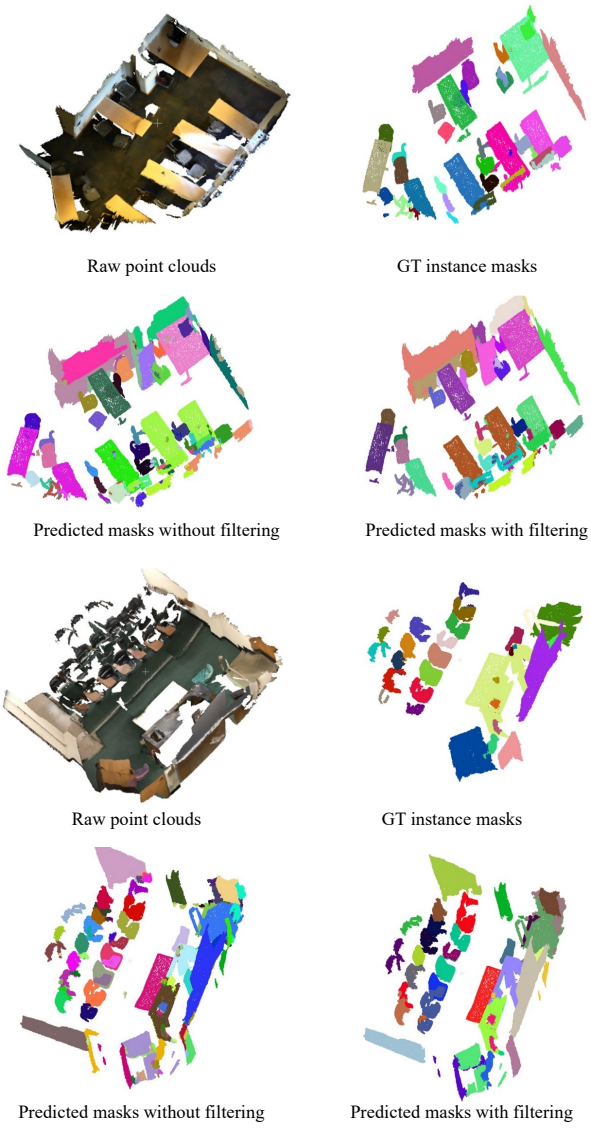


Fig. 13: Qualitative Evaluation of the Mask Proposals. Our class-label-free approach produces high-quality masks that closely resemble the ground truth. Additionally, the incorporation of *Mask Scoring* and *Mask Filtering* further enhances the overall quality of the masks.

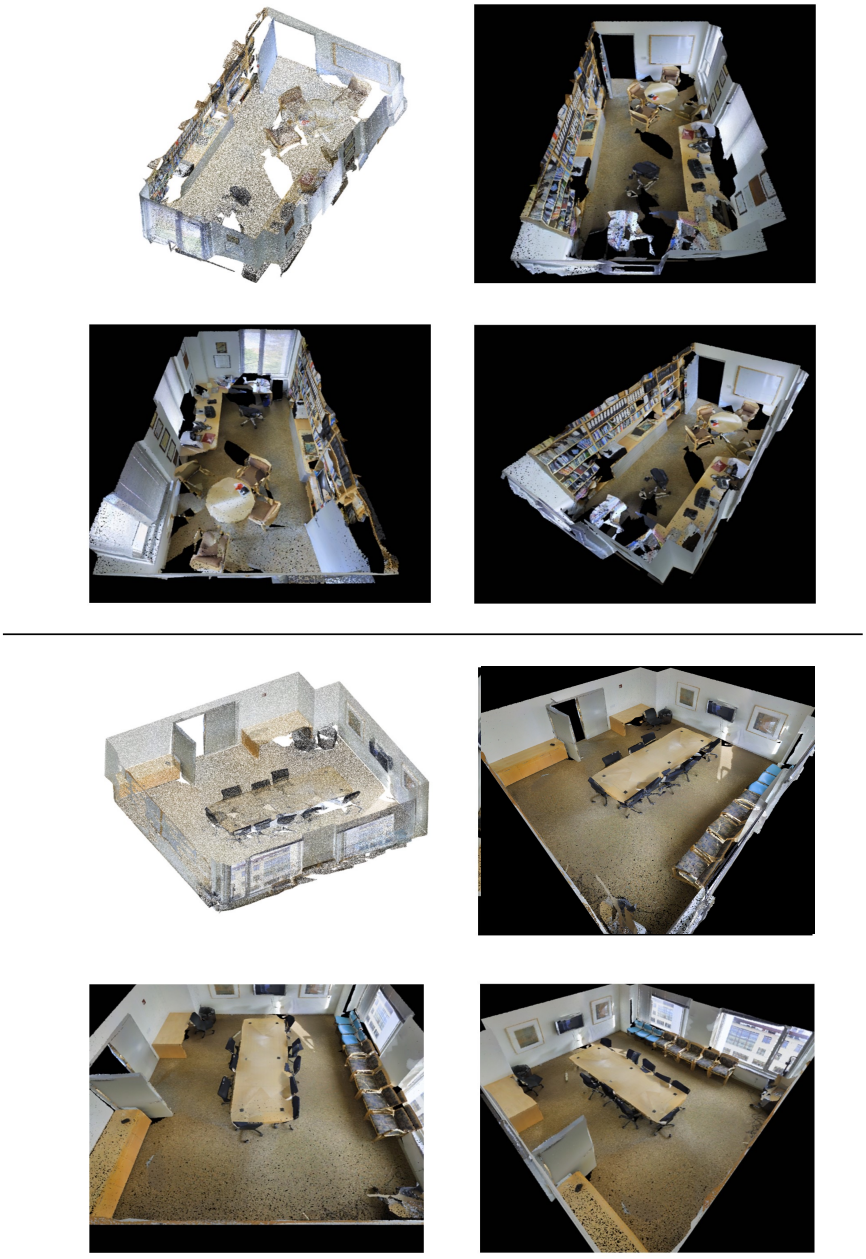


Fig. 14: Synthetic Scene-level Images of S3DIS Generated by *Snap*. The first image is the original spare point cloud, and the following three images are outcomes of the *Snap* module.

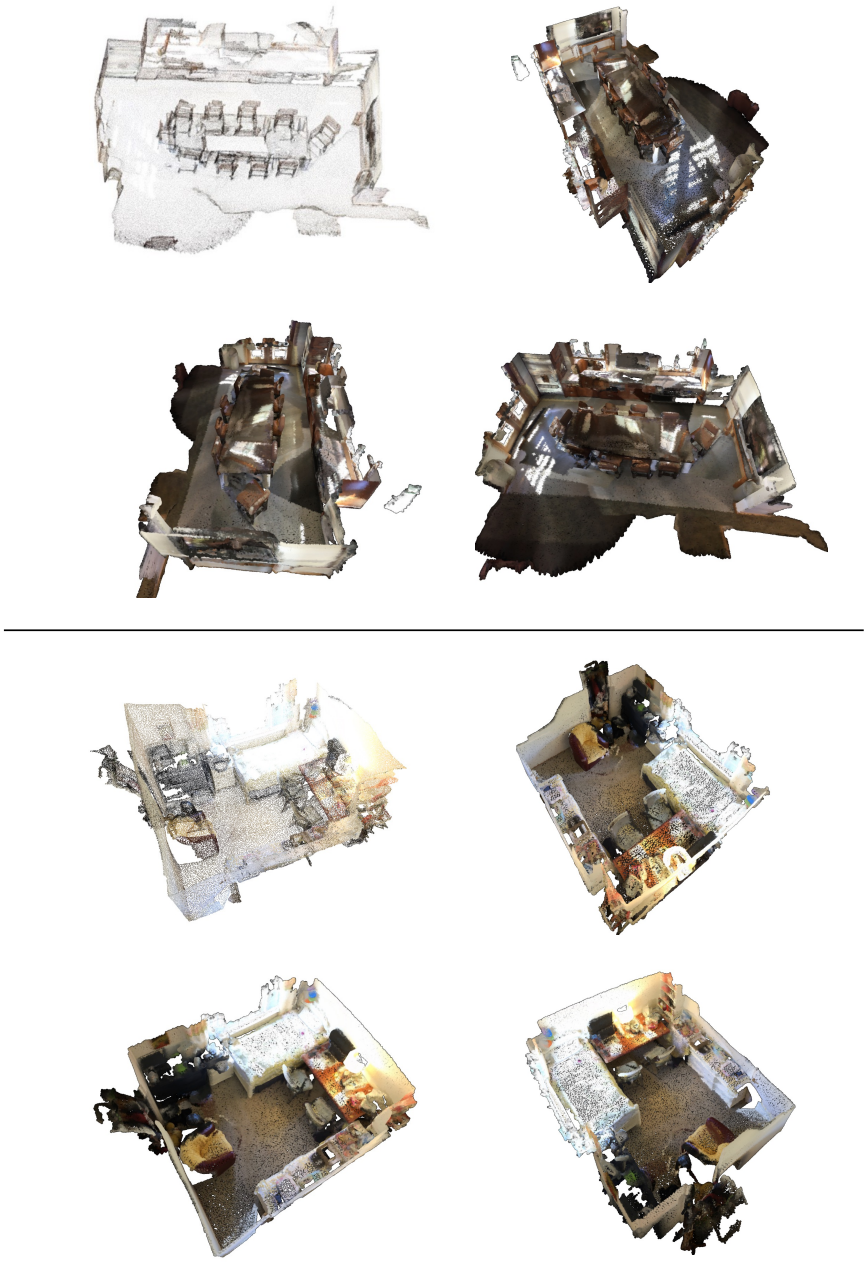


Fig. 15: Synthetic Scene-level Images of ScanNetv2 Generated by *Snap*. The first image is the original spare point cloud, and the following three images are outcomes of the *Snap* module.

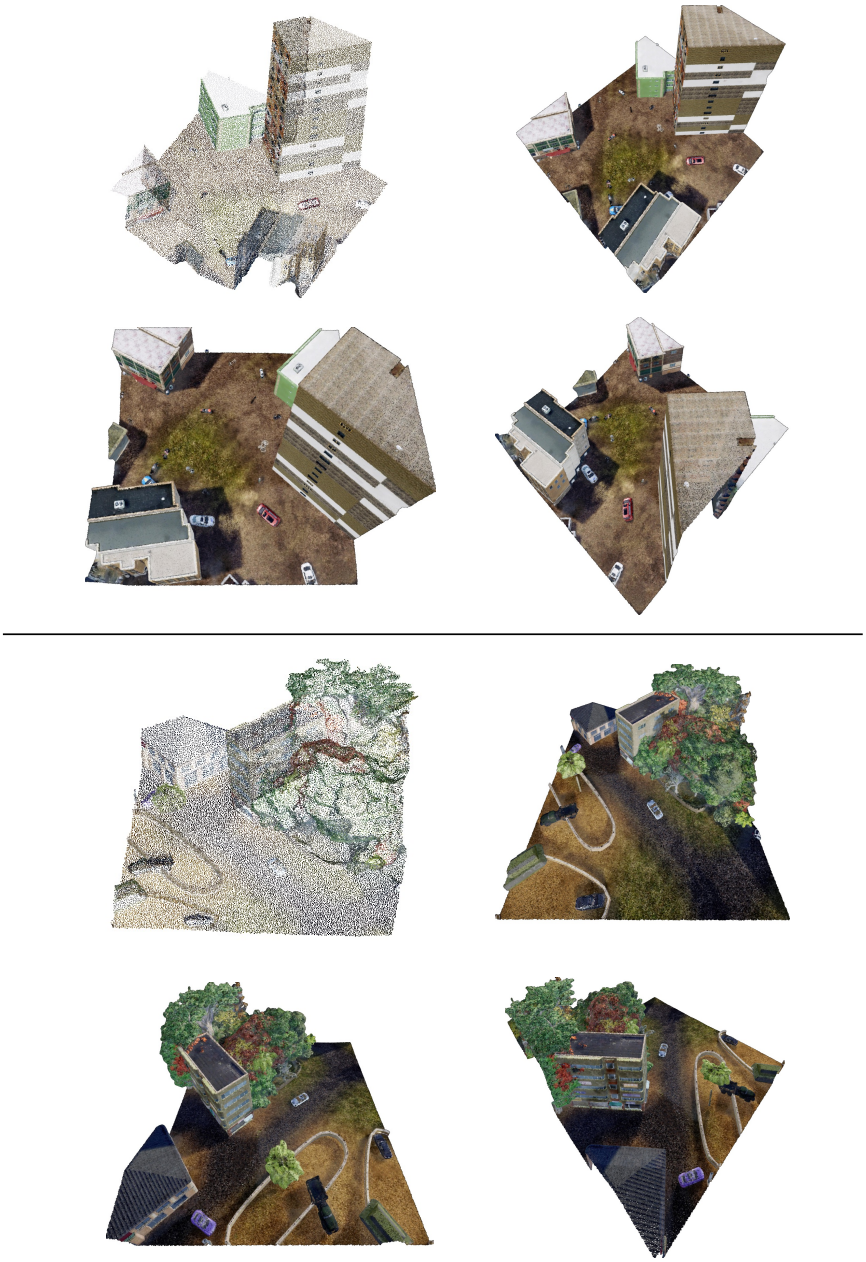


Fig. 16: Synthetic Scene-level Images of STPLS3D Generated by *Snap*. The first image is the original spare point cloud, and the following three images are outcomes of the *Snap* module.

Input Queries: *'ceiling', 'floor', 'wall', 'beam', 'column', 'window', 'door', 'table', 'chair', 'sofa', 'bookcase', 'board'*

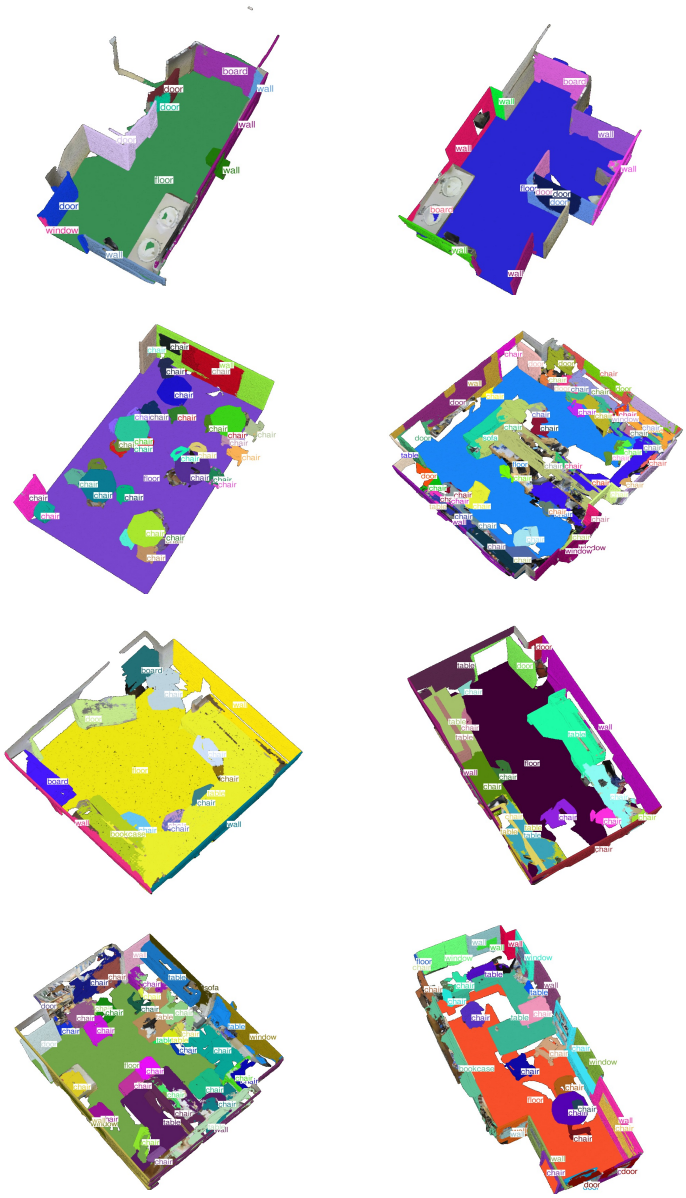


Fig. 17: Open-vocabulary Instance Segmentation Results of S3DIS by OpenIns3D (ODISE). Instance and class labels are presented in the same color.

Input Queries: *'cabinet', 'bed', 'chair', 'sofa', 'table', 'door', 'window', 'bookshelf', 'picture', 'counter', 'desk', 'curtain', 'refrigerator', 'shower curtain', 'toilet', 'sink', 'bathtub'*

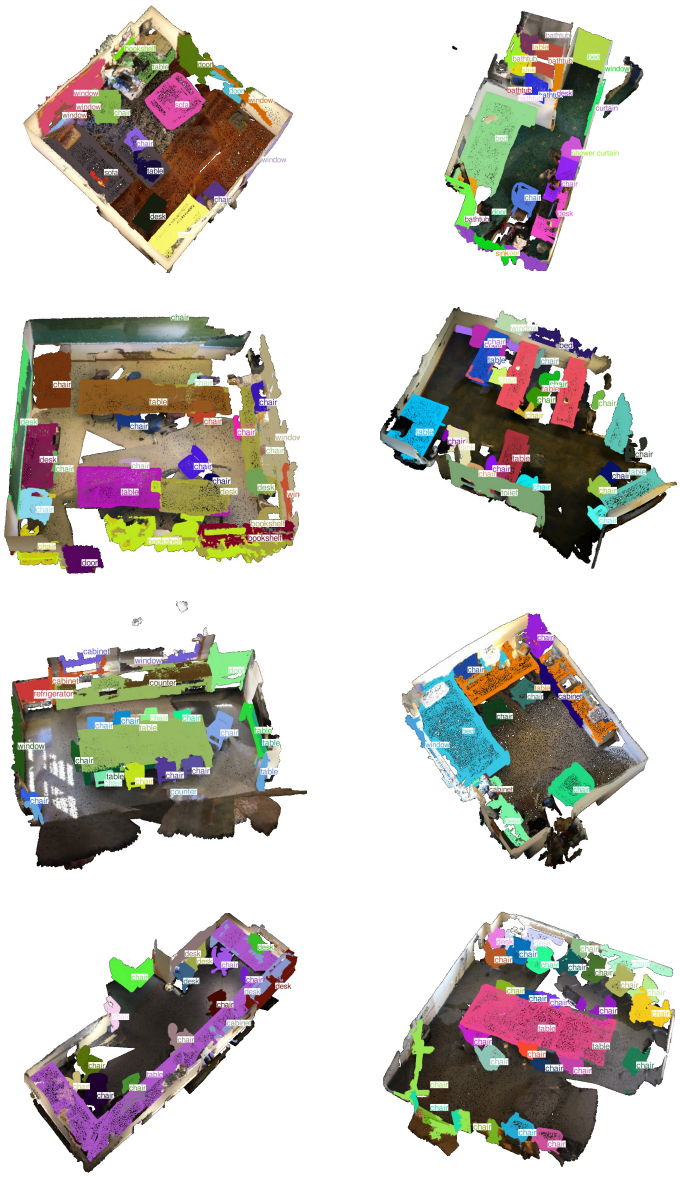


Fig. 18: Open-vocabulary Instance Segmentation Results of ScanNetv2 by OpenIns3D (ODISE). Instance and class labels are presented in the same color.

Input Queries: *'building', 'vegetation', 'vehicle', 'truck', 'Aircraft', 'military vehicle', 'bike', 'motorcycle', 'light pole', 'street sign', 'clutter', 'fence'*

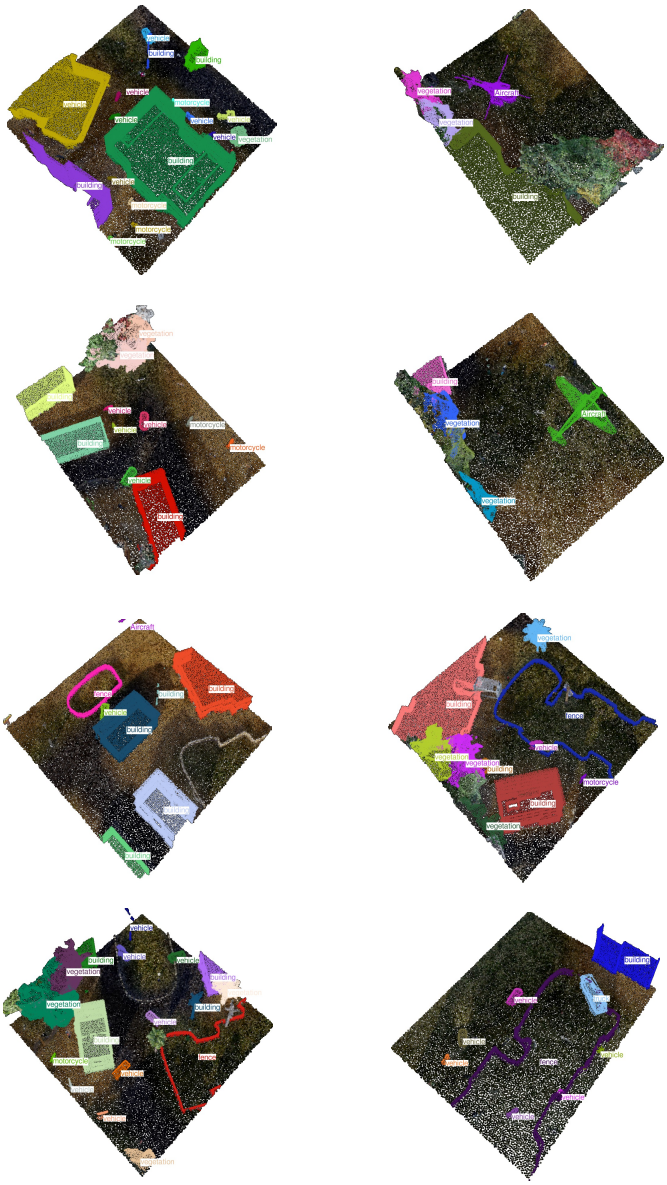


Fig. 19: Open-vocabulary Instance Segmentation Results of STPLS3D by OpenIns3D (ODISE). Instance and class labels are presented in the same color.

References

1. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: CVPR (2016)
2. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: CVPR (2016)
3. Bakr, E.M., Alsaedy, Y.Y., Elhoseiny, M.: Look around and refer: 2d synthetic semantics knowledge distillation for 3d visual grounding. In: NeurIPS (2022)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: NeurIPS (2020)
5. Chen, M., Hu, Q., Yu, Z., THOMAS, H., Feng, A., Hou, Y., McCullough, K., Ren, F., Soibelman, L.: Stpls3d: A large-scale synthetic and real aerial photogrammetry 3d point cloud dataset. In: BMVA (2022)
6. Chen, R., Liu, Y., Kong, L., Zhu, X., Ma, Y., Li, Y., Hou, Y., Qiao, Y., Wang, W.: Clip2scene: Towards label-efficient 3d scene understanding by clip. In: CVPR (2023)
7. Chen, X., Li, S., Lim, S.N., Torralba, A., Zhao, H.: Open-vocabulary panoptic segmentation with embedding modulation. In: ICCV (2023)
8. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
9. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: CVPR (2023)
10. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Lowis3d: Language-driven open-world instance-level 3d scene understanding. In: TPAMI (2024)
11. Ding, Z., Wang, J., Tu, Z.: Open-vocabulary universal image segmentation with maskclip. In: ICML (2023)
12. Griffiths, D., Boehm, J.: SynthCity: A large-scale synthetic point cloud. In: arXiv (2019)
13. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: Semantic3D.net: A new large-scale point cloud classification benchmark. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2017)
14. Huang, R., Peng, S., Takmaz, A., Tombari, F., Pollefeys, M., Song, S., Huang, G., Engelmann, F.: Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. In: ECCV (2024)
15. Huang, T., Dong, B., Yang, Y., Huang, X., Lau, R.W., Ouyang, W., Zuo, W.: Clip2point: Transfer clip to point cloud classification with image-depth pre-training. In: ICCV (2023)
16. Huang, Z., Huang, L., Gong, Y., Huang, C., Wang, X.: Mask scoring R-CNN. In: CVPR (2019)
17. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., Girshick, R.: Segment anything. In: ICCV (2023)
18. Kundu, A., Yin, X., Fathi, A., Ross, D.A., Brewington, B., Funkhouser, T.A., Pantofaru, C.: Virtual multi-view fusion for 3d semantic segmentation. In: ECCV (2020)
19. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: Lisa: Reasoning segmentation via large language model. In: CVPR (2024)

20. Lambert, J., Liu, Z., Sener, O., Hays, J., Koltun, V.: Mseg: A composite dataset for multi-domain semantic segmentation. In: TPAMI (2021)
21. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In: arXiv (2023)
22. Lu, Y., Xu, C., Wei, X., Xie, X., Tomizuka, M., Keutzer, K., Zhang, S.: Open-vocabulary point-cloud object detection without 3d annotation. In: CVPR (2023)
23. Michele, B., Boulch, A., Puy, G., Bucher, M., Marlet, R.: Generative zero-shot learning for semantic segmentation of 3D point cloud. In: 3DV (2021)
24. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: CVPR (2019)
25. Nguyen, P.D.A., Ngo, T.D., Gan, C., Kalogerakis, E., Tran, A., Pham, C., Nguyen, K.: Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In: CVPR (2024)
26. Peng, S., Genova, K., Jiang, C.M., Tagliasacchi, A., Pollefeys, M., Funkhouser, T.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)
27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
28. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
29. Roynard, X., Deschaud, J.E., Goulette, F.: Paris-lille-3d: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. In: The International booktitle of Robotics Research (2018)
30. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: ECCV (2022)
31. Rozenberszki, D., Litany, O., Dai, A.: Unscene3d: Unsupervised 3d instance segmentation for indoor scenes. In: arXiv (2023)
32. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. In: ICRA (2023)
33. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan, X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler, E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M., Lovegrove, S., Newcombe, R.: The replica dataset: A digital replica of indoor spaces. In: arXiv (2019)
34. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.G.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV (2015)
35. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: Openmask3d: Open-vocabulary 3d instance segmentation. In: NeurIPS (2023)
36. Tan, W., Qin, N., Ma, L., Li, Y., Du, J., Cai, G., Yang, K., Li, J.: Toronto-3D: A large-scale mobile lidar dataset for semantic segmentation of urban roadways. In: CVPRW (2020)
37. Xu, C., Yang, S., Galanti, T., Wu, B., Yue, X., Zhai, B., Zhan, W., Vajda, P., Keutzer, K., Tomizuka, M.: Image2point: 3d point-cloud understanding with 2d image pretrained models. In: ECCV (2022)
38. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-Vocabulary Panoptic Segmentation with Text-to-Image Diffusion Models. In: CVPR (2023)

39. Yang, J., Ding, R., Wang, Z., Qi, X.: Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In: CVPR (2024)
40. Zeng, Y., Jiang, C., Mao, J., Han, J., Ye, C., Huang, Q., Yeung, D.Y., Yang, Z., Liang, X., Xu, H.: Clip^2: Contrastive language-image-point pretraining from real-world point cloud data. In: CVPR (2023)
41. Zhang, D., Li, C., Zhang, R., Xie, S., Xue, W., Xie, X., Zhang, S.: Fm-ov3d: Foundation model-based cross-modal knowledge blending for open-vocabulary 3d detection. In: AAAI (2024)
42. Zhang, R., Guo, Z., Zhang, W., Li, K., Miao, X., Cui, B., Qiao, Y., Gao, P., Li, H.: Pointclip: Point cloud understanding by CLIP. In: CVPR (2022)
43. Zhou, C., Loy, C.C., Dai, B.: Denseclip: Extract free dense labels from clip. In: CVPR (2022)
44. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2022)
45. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3d: Exploring unified 3d representation at scale. In: ICLR (2024)
46. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: ECCV (2022)
47. Zhou, X., Girdhar, R., Joulin, A., Krähenbühl, P., Misra, I.: Detecting twenty-thousand classes using image-level supervision. In: ECCV (2022)
48. Zhu, C., Zhang, W., Wang, T., Liu, X., Chen, K.: Object2scene: Putting objects in context for open-vocabulary 3d detection. In: arXiv (2023)
49. Zhu, X., Zhang, R., He, B., Guo, Z., Zeng, Z., Qin, Z., Zhang, S., Gao, P.: Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In: ICCV (2023)