

OnlineAnySeg: Online Zero-Shot 3D Segmentation by Visual Foundation Model Guided 2D Mask Merging

Supplementary Material

1. Overview

In the supplementary material, the sections are briefly introduced as follows:

- We provide more detailed analyses for certain modules of our method in Sec. 2.
- Additional experimental results, including both quantitative and qualitative analyses, are presented in Sec. 3.
- In Sec. 4 we present a video demo showcasing the process of our online segmentation as well as visual comparisons to other online segmentation methods.

2. Method Details

2.1. Discussion about Overlap Ratio

As introduced in Sec. 3.3 of the main paper, we define the **Overlap Ratio**, which quantifies the overlap between a pair of masks in 3D space, with the overlap ratios for all mask pairs stored in the matrix I_t . Another straightforward way to measure the overlap between two masks m_a and m_b is to calculate the proportion of overlapping voxels relative to the voxel count of the mask, for example:

$$or^*(_{a,b}) = \frac{|V_a \cap V_b|}{|V_b|} \quad (1)$$

without considering the visual part of the visible part of mask m_b in the frame set of mask m_a . However, we find that this approach is not feasible for our method. A detailed analysis is provided here.

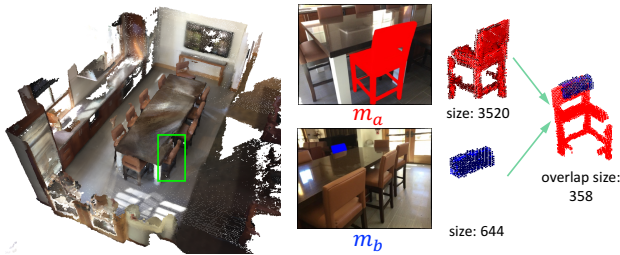


Figure 1. An example of computing overlap ratio without considering "visible part".

An example of two masks belonging to the same 3D instance is given in Fig. 1. These masks represent observations of the same chair, captured from different viewpoints (as highlighted by the green bounding box). The voxel sizes of the lifted 3D masks are 3520 and 644 respectively, with

an intersection size of 358 voxels. According to the definition in Eq. (1), the overlap ratio of m_a to m_b is calculated as $or^*_{(a,b)} = \frac{358}{644} = 0.556$, which is considerably lower than what intuition might suggest. This is because, with the significant difference between the two viewpoints, the two masks cannot align perfectly in 3D space, due to depth noise and occlusion. On the other hand, if we set the threshold for the overlap ratio too low (e.g., 0.5), it may lead to other problems. For instance, if a chair is located very close to a table, and a mis-segmentation occurs in a 2D frame where a large portion of the chair is incorrectly segmented as part of the table, the merging strategy may erroneously combine them into a single instance, making our method highly sensitive to under-segmentation.

By incorporating the "visible part", the overlap ratio of m_a to m_b is $or_{(a,b)} = 1.0$ (following the definition of the overlap ratio in the main paper Sec. 3.3), which aligns with their identity in 3D space. A qualitative comparison of the two different methods for calculating the overlap ratio is provided in Tab. 1.

	AP	AP_{50}	AP_{25}
w/o "Visible Part"	13.6	26.9	40.3
w "Visible Part"	18.6	36.1	53.5

Table 1. Comparison of different methods for calculating the **Overlap Ratio**.

2.2. Extraction of Geometric Feature

In Sec. 3.3 of the main paper, we describe the extraction of geometric features for each detected mask using FCGF [4]. A visualization of the extracted feature point clouds by FCGF is presented in Fig. 2, where points with similar colors indicate higher feature similarity. In our method, the per-point geometric features of the latest reconstructed point cloud S_t are extracted, and the features for each mask are aggregated based on their corresponding scene points, as illustrated in Fig. 3 (b). Compared to the naive method of directly feeding the back-projected point cloud of each detected mask into the geometric feature extractor, as shown in Fig. 3 (a), our approach is both more accurate and time-efficient.

First, as an input sequence typically contains thousands of masks in total, extracting geometric features for each mask individually can result in significant time overhead. Additionally, since FCGF is a fully convolutional network

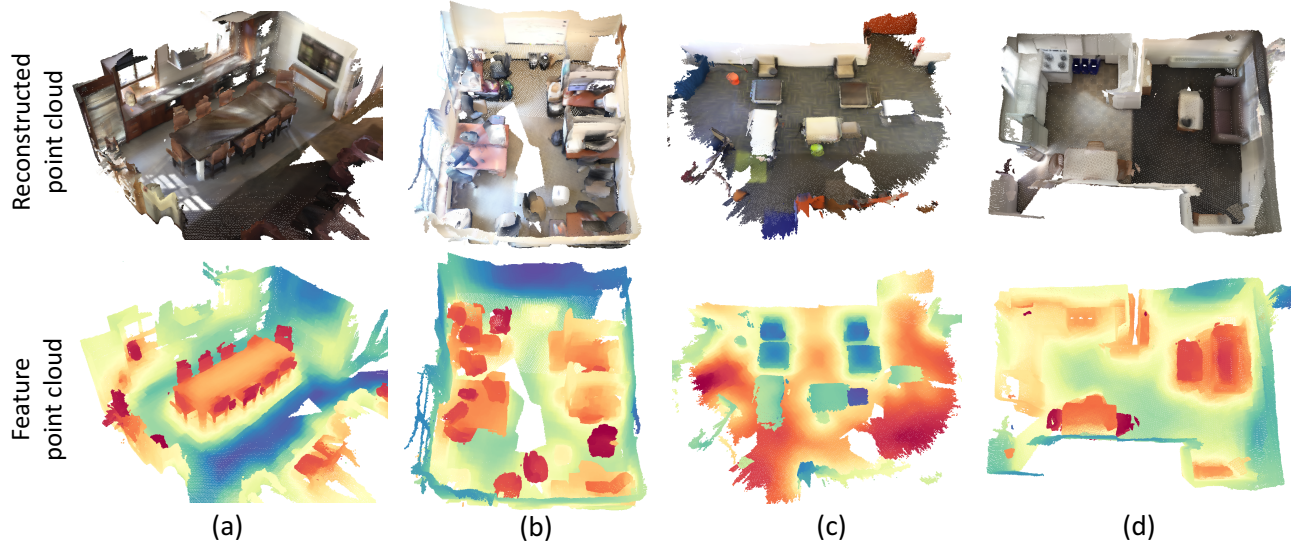


Figure 2. **Visualizations of feature point clouds output by FCGF [4].** The first row shows the reconstructed point clouds from our method. The second row displays their corresponding feature point clouds, colored based on the extracted per-point features, where points with similar colors indicate high feature similarity within the same feature point cloud.

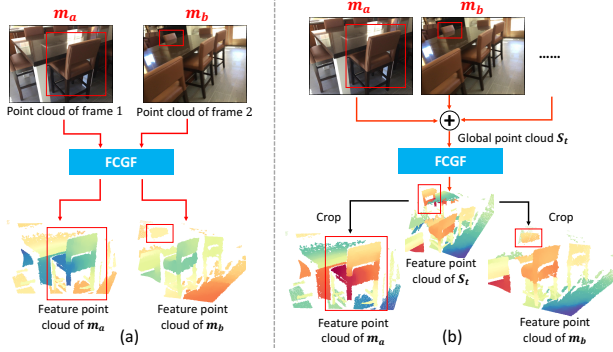


Figure 3. **Comparison of different approaches for extracting geometric features of masks.** (a) Extracting point features separately for each point cloud. (b) Extracting point features from the global point cloud and cropping the resulting feature point cloud to obtain sub-feature point clouds.

capable of capturing broad spatial context, using a more complete input point cloud produces higher-quality output features. Visualizations of the output feature point clouds are presented in Fig. 3, where more similar colors indicate higher feature similarity. For a pair of masks (m_a, m_b) corresponding to the same 3D instance observed from different viewpoints (denoted as frame 1 and frame 2), the significant disparity between the viewpoints can lead to notable feature dissimilarities when extracted separately (Fig. 3, a). In contrast, cropping the complete feature point cloud of S_t ensures globally consistent features for the masks (Fig. 3, b).

2.3. Comparison with Frame-by-frame Mask Merging Strategy

We describe our online mask merging strategy in Sec. 3.4 of the main paper, which establishes mask associations by overall similarities and third-view support. In contrast, some previous methods adopt a “frame-by-frame” merging strategy to process sequential inputs. For example, OVIR-3D [7] focuses on finding instance correspondences between newly detected masks in incoming frames and all existing masks from previous frames. Even though various post-processing operations can be applied to remove redundant instances, the “frame-by-frame” merging strategy can lead to significant issues, particularly in the following scenarios. A typical example is shown in Fig. 4.

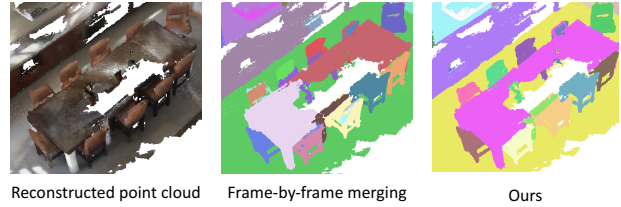


Figure 4. **Visual comparison of different merging strategies.** The “frame-by-frame” mask merging strategy struggles to handle segmentation for large instances and significant viewpoint disparity.

First, for a large object observed partially from different viewpoints with minimal overlap, the “frame-by-frame” merging strategy struggles to correctly associate the newly

observed part with previous observations. For instance, as illustrated in Fig. 4, a long dining table is segmented into two separate instances by the "frame-by-frame" merging strategy. Second, if a previously detected object is scanned again from a completely different viewpoint, the newly detected instance is often not successfully matched to the existing one, resulting in it being treated as a new instance. This issue is exemplified by the chairs in Fig. 4.

Our mask merging strategy, on the other hand, fully leverages all previous observations to ensure global consistency. Additionally, by incorporating third-view supporting to establish extra associations between masks with limited overlap, these issues can be significantly mitigated. A qualitative comparison of these two merging strategies is presented in Tab. 2.

	AP	AP_{50}	AP_{25}
Frame-by-frame	14.7	29.0	44.3
Ours	18.6	36.1	53.5

Table 2. Comparison of different mask merging strategies.

2.4. Comparison without using the mapping table

We introduce the mapping table in Sec. 3.2 of the main paper, which maps the IDs of the original 2D masks to the IDs of the current 3D masks, enabling the tracking of each mask throughout every merging process. Instead of directly updating the mask ID lists in the hashed voxel volume, the re-assignment of mask IDs during the merging stage (Section 3.4 of the main paper) only triggers a synchronous update of the mapping table, keeping the hash table append-only. This design significantly accelerates the updating of the mask bank G_t , as frequent voxel modifications in the volume are highly time-consuming.

A speed comparison of the scanning process with and without the mapping table is shown in Fig. 5, demonstrating the effectiveness of this design. Without the mapping table, the process is approximately 10 times slower, and the slowdown can even reach up to 20 times as the number of masks increases.

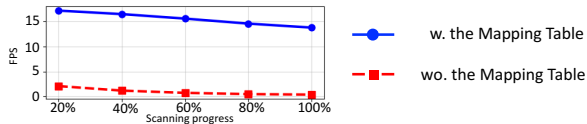


Figure 5. Comparison of speed with and without the mapping table.

3. Additional Experiments

3.1. More Full-sequence Results

The qualitative results and visual comparison of full-sequence instance segmentation are provided in the Sec. 4.2 and Sec. 4.3 of the main paper. To facilitate a more detailed comparison, we present the per-sequence results of our method and other online methods on SceneNN [6], as shown in Tab. 3.

SAM3D [11] performs poorly overall due to its naive mask merging strategy. Compared with EmbodiedSAM [9], our method achieves higher performance under low and medium IoU thresholds but performs worse under high IoU thresholds. The reason is that as a learning-based method, EmbodiedSAM trains a classification head to distinguish between foreground and background in advance. Background instances, such as walls and floors, are filtered out, leaving only foreground instances to be processed and merged. In contrast, as zero-shot methods, both our method and MaskClustering [10] lack pre-filtering operations and process all detected instances equally. This phenomenon is evident in Fig. 6 of the main paper, where the background is painted uniformly in gray. Since, under high IoU thresholds, the predicted walls and floors often fail to match the ground truth accurately, zero-shot methods may experience a drop in performance.

3.2. More Intermediate Results

In addition to the final segmentation results, we evaluate the intermediate segmentation outputs of EmbodiedSAM and our method, with results presented in Fig. 4 and Tab. 2 of the main paper. Furthermore, we conduct more detailed experiments to explore the core reasons behind the superior real-time segmentation performance of our method.

The real-time segmentation results at different quarters of the input sequence are presented in Tab. 4. Each sub-table shows the AP scores at the completion of 25%, 50%, and 75% of the sequence respectively. Our method outperforms EmbodiedSAM in most sequences, demonstrating the effectiveness of our global-consistent merging strategy. This also indicates that our approach has less reliance on post-processing steps, such as smoothing, to achieve high-quality segmentation results.

Meanwhile, we also provide a detailed visual comparison of instance segmentation results on progressively scanned scenes in Fig. 7. Several challenging scenes from ScanNet200 [8] and SceneNN [6] are showcased to highlight the performance of our method in complex environments. The intermediate segmentation results are directly presented on the reconstructed point cloud (EmbodiedSAM) or reconstructed mesh (ours), with backgrounds painted in gray. We observe that while EmbodiedSAM performs well on ScanNet200 (the first and fourth scenes), the

Scene	SAM3D* [11]			EmbodiedSAM** [9]			Ours*		
	AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}	AP	AP_{50}	AP_{25}
005	9.1	25.6	54.8	15.5	25.9	48.4	21.0	49.5	63.4
011	29.1	44.5	53.0	44.1	53.0	57.9	33.3	54.5	64.9
015	6.4	16.9	32.7	17.8	32.3	43.3	16.9	35.7	45.1
030	4.1	10.0	38.2	28.0	42.7	52.6	17.1	29.2	50.0
054	9.3	27.0	50.7	23.5	41.4	61.4	16.9	31.7	61.2
080	9.0	22.4	54.3	7.5	17.6	30.0	13.6	27.8	54.1
089	5.9	17.2	48.2	9.7	22.3	42.3	7.8	16.1	50.4
093	12.5	23.8	46.2	24.7	37.9	47.3	19.8	41.1	65.5
096	9.9	22.5	55.8	21.7	36.2	44.5	16.4	29.2	60.6
243	4.5	14.2	40.9	16.3	27.6	51.4	12.7	31.7	69.1
263	14.8	39.2	59.0	27.9	38.9	47.3	26.6	52.6	66.7
322	16.4	31.8	51.3	33.1	41.1	52.8	31.7	56.9	77.3
Overall	9.1	21.3	43.4	20.1	32.5	46.3	18.1	35.3	59.5

Table 3. **Full-sequence instance segmentation results on SceneNN [6].** We present the per-sequence results of the online methods. *: Zero-shot method, **: Learning-based method.

dataset on which it is trained, it tends to output noisy segmentation results when transferred to other datasets (the other four scenes), as highlighted by the red bounding boxes in Fig. 7.

The reason for this is that the mask merging strategy used by EmbodiedSAM is essentially a "frame-by-frame" approach. It utilizes the encoded features output by the trained model, along with the IoU of mask bounding boxes, to evaluate the similarity between newly detected masks and previous instance masks. This approach avoids calculating precise 3D spatial overlap between mask pairs, which may lead to issues when transferring to new datasets with different characteristics. In contrast, our mask merging strategy incorporates more global information during each merging step, fundamentally differing from the "frame-by-frame" approach, as discussed in Sec. 2.3.

3.3. More Ablation Studies

We also test different τ_{weight} values and the results are shown in Tab. 5, demonstrating the robustness of our method and confirming $\tau_{\text{weight}} = 5$ as the optimal parameter choice.

τ_{weight}	3	4	5	6	7	8
AP_{25}	52.1	53.5	53.5	53.5	53.0	52.6
AP_{50}	32.7	36.1	36.1	36.1	36.3	34.1
AP	17.9	18.6	18.6	18.3	18.0	17.4

Table 5. Ablation study on τ_{weight} on ScanNet200.

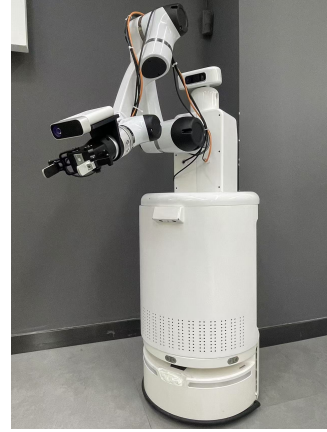


Figure 6. **Equipment setup for real-world experiments.** The experimental setup consists of an E05 robotic arm [1] guided by the Water II vehicle [3], with a Microsoft Azure Kinect DK RGB-D sensor [2] mounted at the arm's end.

4. Online Video Demo

We also provide a video demonstration, showcasing the real-time reconstruction and segmentation process of our method on several challenging scenes from ScanNet200 [5, 8] and SceneNN [6]. In addition, we deploy our online segmentation method on an E05 robotic arm [1], guided by the Water II vehicle [3], with a Microsoft Azure Kinect DK RGB-D sensor [2] mounted at the end, as illustrated in Fig. 6. A real-world demo is also presented in the video.

References

- [1] Elfin collaborative robot e05. <https://www.hansrobot.net/elfin-collaborative-robot>. Accessed: 2024-11-21. 4
- [2] Microsoft azure kinect dk. <https://learn.microsoft.com/en-us/azure/kinect-dk/>. Accessed: 2024-11-21. 4
- [3] Water ii. <https://www.yunji.ai/en/autonomous-robot-platform-chassis.html>. Accessed: 2024-11-21. 4
- [4] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8958–8966, 2019. 1, 2
- [5] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 4
- [6] 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 *2016 fourth international conference on 3D vision (3DV)*, pages 92–101. Ieee, 2016. 3, 4, 5
- [7] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Conference on Robot Learning*, pages 1610–1620. PMLR, 2023. 2
- [8] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *European Conference on Computer Vision*, pages 125–141. Springer, 2022. 3, 4
- [9] Xiuwei Xu, Huangxing Chen, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Embodiedsam: Online segment any 3d thing in real time. *arXiv preprint arXiv:2408.11811*, 2024. 3, 4, 5, 7
- [10] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28274–28284, 2024. 3
- [11] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 3, 4

(a) Intermediate results at 25% sequence completion.

Scene	EmbodiedSAM** [9]			Ours*		
	AP	AP ₅₀	AP ₂₅	AP	AP ₅₀	AP ₂₅
005	7.1	17.5	51.0	8.6	26.2	47.1
011	11.5	25.7	42.9	34.7	52.5	61.3
015	14.7	34.1	47.3	41.6	61.5	81.1
030	12.2	20.9	37.4	9.3	25.7	48.2
054	14.3	45.0	64.3	13.0	36.7	65.6
080	7.8	13.0	52.4	22.0	49.1	49.3
089	6.6	23.8	40.4	3.9	9.7	44.9
093	17.8	46.6	63.2	32.5	51.2	72.6
096	11.7	36.8	38.3	10.4	19.1	59.5
243	11.1	22.9	62.5	18.6	46.5	59.6
263	26.5	52.2	61.5	42.5	73.2	86.6
322	9.3	22.2	35.0	18.2	36.7	61.1

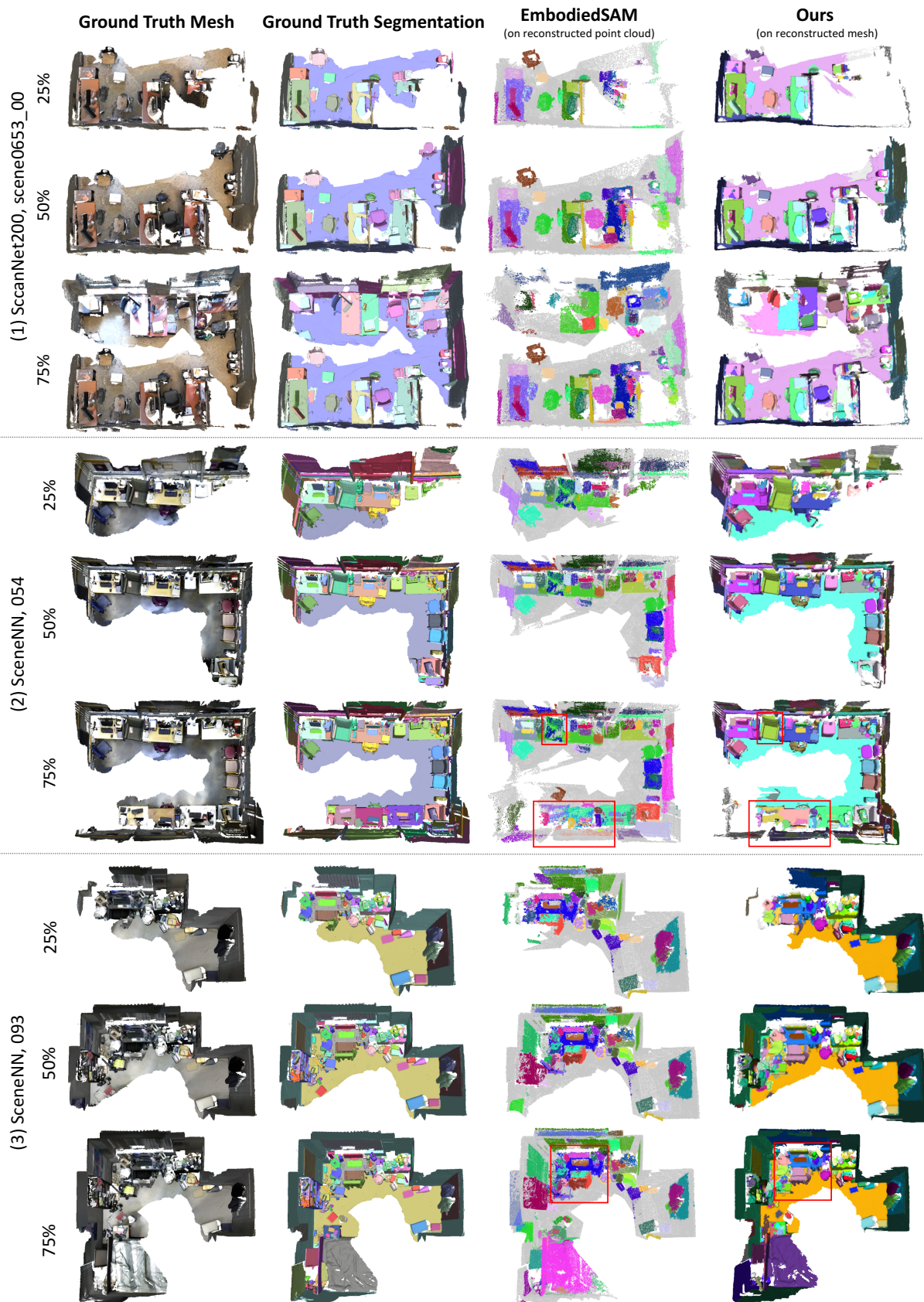
(b) Intermediate results at 50% sequence completion.

Scene	EmbodiedSAM** [9]			Ours*		
	AP	AP ₅₀	AP ₂₅	AP	AP ₅₀	AP ₂₅
005	8.5	17.4	42.3	12.9	34.8	60.2
011	15.8	35.1	55.2	32.0	61.2	68.1
015	17.2	37.9	38.1	20.4	41.1	52.3
030	14.3	22.7	40.1	17.1	30.0	48.5
054	19.7	60.6	76.3	20.4	43.0	69.4
080	5.1	16.5	35.6	17.8	40.6	57.2
089	4.8	18.9	41.0	6.1	16.9	49.0
093	16.7	39.9	59.7	21.0	38.6	63.2
096	10.0	35.3	42.8	15.3	26.4	61.1
243	9.1	20.2	63.8	10.9	31.9	60.8
263	29.4	52.2	69.2	38.6	68.3	76.7
322	11.2	24.7	37.5	16.9	42.9	74.1

(c) Intermediate results at 75% sequence completion.

Scene	EmbodiedSAM** [9]			Ours*		
	AP	AP ₅₀	AP ₂₅	AP	AP ₅₀	AP ₂₅
005	9.3	20.2	53.5	15.5	37.7	58.5
011	17.8	33.3	59.1	37.0	65.1	70.1
015	12.8	27.1	41.3	26.1	35.4	45.6
030	18.3	30.6	47.9	15.9	25.0	41.7
054	16.2	41.1	67.5	19.4	40.2	61.6
080	3.1	7.2	33.9	14.2	30.8	51.2
089	5.0	19.0	45.5	5.1	18.3	52.6
093	13.6	31.0	58.8	19.8	34.7	63.0
096	11.5	41.5	49.0	13.8	25.4	66.0
243	11.7	23.1	66.7	16.1	40.5	66.7
263	22.4	39.2	57.1	35.2	62.2	74.2
322	18.3	38.0	47.2	26.3	49.8	77.1

Table 4. **Intermediate instance segmentation results on SceneNN [6].** The instance segmentation results are evaluated by mapping from the reconstructed point cloud or mesh to ground truth point cloud through point correspondences. *: Zero-shot method, **: Learning-based method.



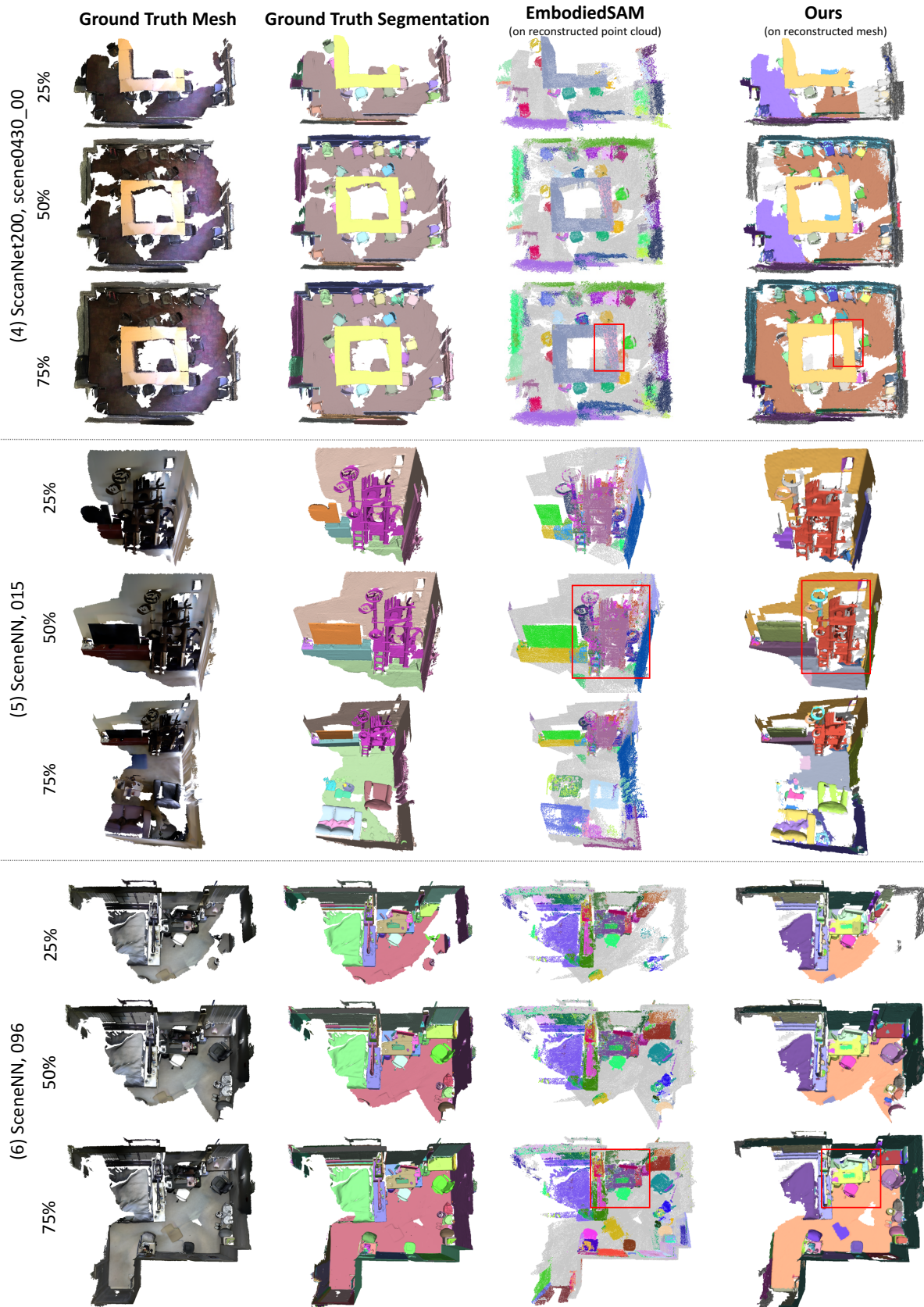


Figure 7. Visual comparison of intermediate segmentation results with EmbodiedSAM [9].