Supplementary Materials: Details Matter for Indoor Open-vocabulary 3D Instance Segmentation

1. Algorithm on 3D Proposal Merge and Refinement

We present a more detailed algorithm for merging and refining 3D proposals in Alg. 1. This algorithm refines and consolidates a list of tracklets and their associated 3D proposals. It iteratively evaluates the pairwise similarity of proposals using an Intersection-over-Union (IOU) cost matrix and merges those exceeding a defined similarity threshold. During merging, the 3D points and corresponding tracklets are combined, followed by a refinement step that removes low-visibility 3D superpoints from the merged proposal based on their visibility in the tracked 2D masks. A dynamic validity list tracks unmerged proposals and is updated after each merge iteration. The process continues until no proposals meet the merging criteria. This iterative method effectively consolidates overlapping proposals, enhancing the overall accuracy and coherence of 3D proposals.

Algorithm 1 3D Proposal Merge and Refinement

```
Input: A list of $K$ tracklets $\{\mathbf{T}_k\}_{k=1}^{K}$ and associated 3D proposals $\{\mathbf{m}_k\}_{k=1}^{K}$, $k = 1, 2, \ldots, K$
Output: A list of filtered tracklets and 3D proposals

$K \leftarrow$ # of 3D proposals
$\mathbf{m} \leftarrow [\mathbf{m}_1, \ldots, \mathbf{m}_K] \in \{0, 1\}^{K \times N}$
$\mathbf{T} \leftarrow [\mathbf{T}_1, \ldots, \mathbf{T}_K]$
$V \leftarrow [1, 2, \ldots, K]$                          ▷ Initialize valid proposal indices
$\mathbf{C}^{\text{merge}} \leftarrow$ getIoUCostMatrix($\mathbf{m}$)
should_merge $\leftarrow$ Any($\mathbf{C}^{\text{merge}} > \tau^{\text{merge}}$)
while should_merge = True do
    visited $\leftarrow$ hashmap                        ▷ Track visited proposals
    for row $r = 1, 2, \ldots, K$ do
        if visited[$r$] = True then
            continue
        end if
        for col $c = 1, 2, \ldots, K$ do
            if $r = c$ or visited[$c$] or $\mathbf{C}^{\text{merge}}[r, c] \leq \tau^{\text{merge}}$ then
                continue
            end if
            $\mathbf{m}_r \leftarrow \mathbf{m}_r \cup \mathbf{m}_c$                    ▷ Merge 3D proposals
            $\mathbf{T}_r \leftarrow \mathbf{T}_r \cup \mathbf{T}_c$                      ▷ Merge tracklets
            $\mathbf{m}_r \leftarrow$ refine3DProposal($\mathbf{m}_r, \mathbf{T}_r$)  ▷ Refine 3D proposal
            $V \leftarrow V \setminus \{c\}$                   ▷ Remove merged proposal
            visited[$c$] $\leftarrow$ True
        end for
        visited[$r$] $\leftarrow$ True
    end for
    $K \leftarrow$ length($V$)                             ▷ Update # of proposals
    $\mathbf{m} \leftarrow [\mathbf{m}_{i_1}, \ldots, \mathbf{m}_{i_K}],\ i_k \in V$        ▷ Update 3D proposal list
    $\mathbf{T} \leftarrow [\mathbf{T}_{i_1}, \ldots, \mathbf{T}_{i_K}],\ i_k \in V$            ▷ Update tracklet list
    $V \leftarrow [1, 2, \ldots, K]$             ▷ Re-initialize valid proposal indices
    $\mathbf{C}^{\text{merge}} \leftarrow$ getIoUCostMatrix($\mathbf{m}$)
    should_merge $\leftarrow$ Any($\mathbf{C}^{\text{merge}} > \tau^{\text{merge}}$)
end while
```

2. More Implementation Details

We leverage Alpha-CLIP [10] with SAM [6] for instance classification based on text queries. For the ScanNet200 and Replica datasets, we use instance bounding boxes as queries for mask retrieval from SAM, while for the S3DIS dataset, we utilize subsampled points. This choice is driven by the presence of “stuff” classes in S3DIS, where subsampled points introduce less noise compared to bounding boxes. For proposal filtering, we adopt different SMS thresholds (τ^{SMS}) tailored to each experiment. Performance remains stable within a reasonable range ($\tau^{\text{SMS}} \in [-1.0, 1.0]$), corresponding to a standard deviation range, as elaborated in Sec. 4.1. For instance classification, we select the top 20 visible images for ScanNet200 and Replica and the top 40 images for S3DIS. Following OpenMask3D [11], we use a confidence value of 1.0 to evaluate Average Precision (AP) and Average Recall (AR) metrics. The inference time for our method on the Replica dataset is approximately 597 seconds per scene, closely matching the 547 seconds reported for OpenMask3D [2, 11]. This similarity arises from both methods utilizing SAM and CLIP for instance feature extraction. In our analysis, instance classification takes up to around 456 seconds, which is 76% of the computational cost. All inference times were measured on a single NVIDIA RTX 4090 GPU. Additionally, we employ a text query template, “*a blurry photo of* $\{$*CLASS_NAME*$\}$ *in a room*,” adapted from CLIP [8]. For Top-K evaluations, we use $K = 300$ for 2D-only and 3D-only experiments, and adopt $K = 600$ for 2D+3D experiments, following Open3DIS. OpenYOLO3D adopted $K = 600$ for the 3D-only experiment on the ScanNet200 dataset.

3. Analysis of Computational Cost

Fig. 1 presents the analysis of the impacts of various factors on the computation time. We plot graphs to demonstrate the correlation of 1) the number of points in the point cloud, 2) the number of image frames, 3) the number of instances, and 4) the number of different semantic classes present in the scene. At last, we show the stage-wise computation time of our method. As shown, we can see meaningful correlations between those factors and computation time. Also, the 2D grounding step takes the longest computation time in our method, followed by instance classification, 3D pro-

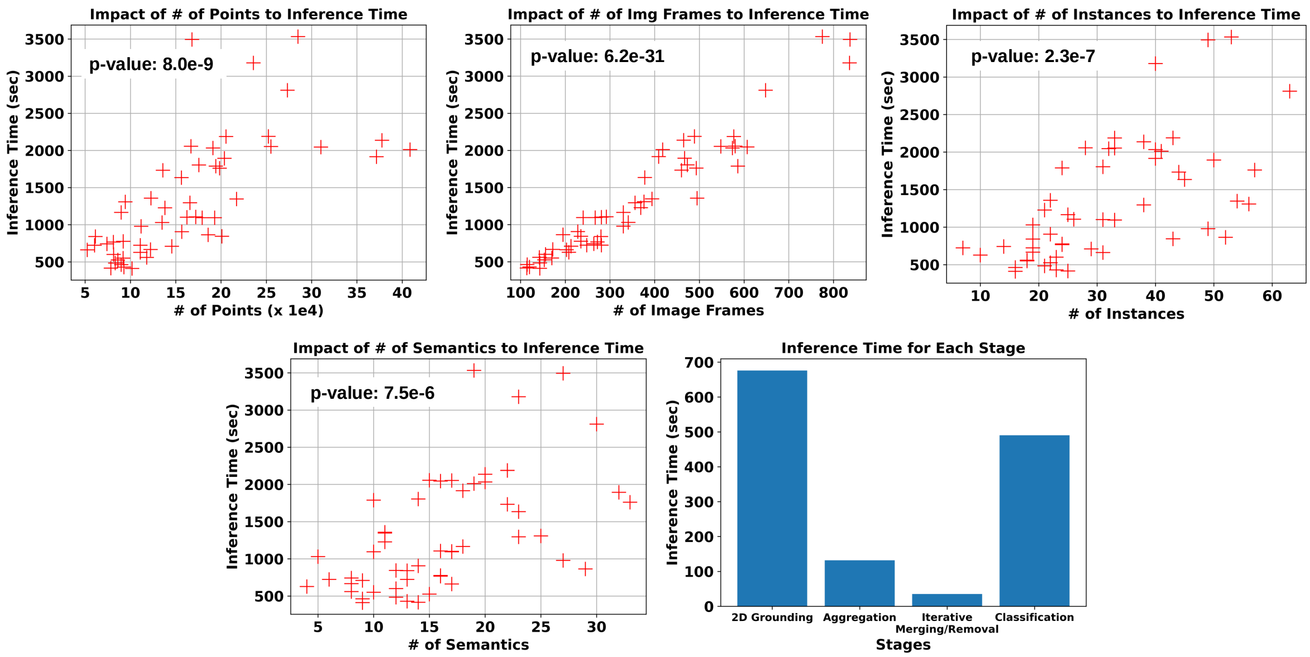

Figure 1. **Computation time analysis of various factors on the subset of ScanNet200 validation set.**

posal aggregation, and iterative merging and removal.

4. Additional Experiments

This section presents additional quantitative and qualitative results not included in the main paper.

4.1. Quantative Results

Class-agnostic Evaluation Results on ScanNet++ We further evaluate our method on the ScanNet++ dataset. Unlike other datasets, we only evaluate our image-based proposal generation pipeline, excluding point cloud-based 3D proposals. This is because we experience a non-trivial amount of distributional gap when we apply 3D instance segmentation models trained on other datasets such as ScanNet200. As reported in Table 1, our method demonstrates on-par evaluation results on the AP metric with Open3DIS [7] + SAM for 2D object grounding. Under the same VFM (Grounded SAM) for 2D grounding, our method shows better performance in AP_{50} and AP_{25} by 1.7% and 3.7%, respectively. SAI3D [14] presents superior results in the AP_{25} metric, surpassing all other methods by a large gap. We found that our iterative merging/removal step does not contribute to precise 3D proposal generation as much as it used to in other datasets. We conjecture that this is because the ScanNet++ dataset includes more small and fine objects that may get removed by merging and removing overlapped and included proposals. However, our method still maintains reasonable performance, showing on-par results with SoTA methods.

Method	AP	AP_{50}	AP_{25}
SAM3D [12]	7.2	14.2	29.4
SAM-guided Graph Cut [4]	12.9	25.3	43.6
Segment3D [5]	12.0	22.7	37.8
SAI3D [14]	17.1	31.1	**49.5**
Open3DIS (G-SAM)† [7]	18.2	30.7	40.6
Open3DIS (SAM) [7]	**18.5**	**33.5**	44.3
Ours (2D Only)	18.2	32.4	44.3

Table 1. **Class-agnostic evaluation on the ScanNet++ dataset [13].** †numbers are obtained from their official code.

S3DIS Results Including "stuff" Classes. We present results that include ”stuff” classes—specifically floor, ceiling, and wall—for evaluation. These classes were excluded from the main paper's evaluation, as our task focuses on segmenting instances, and the notion of instances hardly applies to those classes. As reported in Table 2, our method consistently outperforms baselines in 2D-only and 2D+3D groups. However, in the 3D-only group, our method falls slightly behind OpenYOLO3D in the mAP_{50} and mAP_{25} metrics, primarily due to weaker performance on "stuff" classes in these metrics. Nevertheless, our objective is to improve performance on "thing" classes, which does not necessarily correlate with gains on "stuff" classes. While this gap could be addressed by incorporating panoptic segmentation methods to handle both types of classes, such exploration is beyond the scope of this work. Importantly, our method achieves SoTA results in all AR metrics across all three groups.

Ablation Study on SMS filtering. Fig. 3 illustrates the effect of varying SMS filtering thresholds on the AP met-

Methods	3D Proposals Image-based	 Point cloud-based	mAP	mAP$_{50}$	mAP$_{25}$	mAR	mAR$_{50}$	mAR$_{25}$
Open3DIS [7]	✓	✗	17.1	27.1	36.7	24.7	37.5	49.0
Ours (2D Only)	✓	✗	**22.5**	**35.0**	**47.6**	**32.0**	**48.7**	**63.7**
Open3DIS [7]	✗	✓	24.7	30.6	35.9	34.4	41.5	47.3
OpenYOLO3D [2]	✗	✓	**37.4**	**49.4**	**56.9**	45.6	56.6	62.6
Ours (3D Only)	✗	✓	**37.4**	46.6	54.7	**47.5**	**57.2**	**64.6**
Open3DIS [7]	✓	✓	27.8	33.9	39.3	44.8	53.6	60.6
Ours (2D + 3D)	✓	✓	**33.5**	**42.4**	**47.9**	**53.6**	**66.0**	**72.7**

Table 2. **OV-3DIS results on the S3DIS dataset [1].** The numbers are obtained by using 12 classes, including stuff classes such as floor, ceiling, and wall. Top-1 evaluation protocol is used.

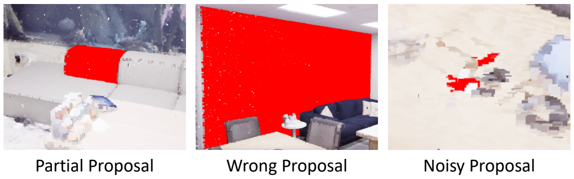

Figure 2. **Visualization of filtered proposals by using the Standardized Maximum Similarity (SMS) score.** The SMS score effectively filters out partial proposals (e.g., only part of a sofa is covered), incorrect proposals that do not match any text queries (e.g., "wall" class not included in the evaluation set), and noisy proposals lacking meaningful object representation.

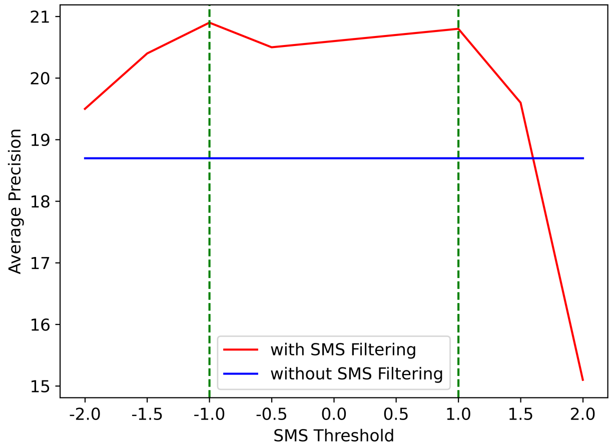

Figure 3. **Impacts of varying SMS filtering thresholds on the AP metric.** The red line denotes the AP values across different SMS filtering thresholds, and the blue line indicates AP without using SMS filtering. The green vertical lines indicate a desirable range of SMS filtering thresholds. The numbers are measured on the Replica dataset.

ric using the Replica dataset. Within the standard deviation range of [-1, 1], the variance remains minimal compared to the outer ranges, with a maximum gap of only 0.4%. Notably, applying SMS filtering consistently outperforms the baseline experiment conducted without filtering.

Impact of Hyper-parameters. Table 3 demonstrates the impacts of different hyper-parameter values on the generated 3D proposal quality. As reported, our algorithm is sensitive to τ^{img} and τ^{inst} values because they are applied in the first step of our generation and also provide the basis for later operations. Nevertheless, our method remains less sensitive to other hyper-parameters.

τ^{img}	AP	τ^{inst}	AP	τ^{ref}	AP	τ^{merge}	AP	τ^{incl}	AP
0.0	26.4	0.0	28.0	0.0	33.5	0.1	**36.1**	0.5	**36.3**
0.1	**35.1**	0.1	33.9	0.2	34.0	0.3	35.1	0.7	35.6
0.3	27.9	0.3	**35.1**	0.4	**35.1**	0.5	33.6	0.9	35.0
0.5	16.2	0.5	29.9	0.6	33.8	0.7	33.0	0.99	35.1

Table 3. **Impact of hyper-parameters on the subset of the ScanNet200 validation set.** Class-agnostic APs are reported.

Alpha CLIP vs CLIP Table 4 demonstrates impacts of using Alpha-CLIP and SMS filtering for instance classification. In all three datasets, using Alpha-CLIP brings meaningful performance gains consistently across all datasets and metrics. Applying SMS filtering further improves this, achieving SoTA performance.

4.2. Qualitative Results

Qualitative Results with Dataset-Provided Text Queries. We present more qualitative comparisons on the ScanNet200 dataset [3] in Figs. 4 and 5. As shown, both Open3DIS and OpenYOLO3D fail to detect certain instances, primarily due to missing image-based 3D proposals or incorrect instance classifications. In contrast, our method not only generates accurate proposals but also classifies them correctly. Open3DIS, in particular, occasionally misidentifies the "floor" as an instance (see third/fourth and fourth rows of Figs. 4 and 5, respectively), reflecting imperfections in their image-based proposal generation. We also provide qualitative results on the S3DIS and Replica datasets in Fig. 6, demonstrating that our method accurately retrieves most proposals, with only a few instances missed. We attribute these missed instances to either the domain gap between real-world data and the synthetic data from Replica or the training nature of CLIP, which emphasizes foreground regions over background elements such as

Dataset	Method	mAP	mAP$_{50}$	mAP$_{25}$	mAR	mAR$_{50}$	mAR$_{25}$
ScanNet200	Ours w/ CLIP	27.5	34.7	38.2	52.4	65.6	71.8
	+ Alpha-CLIP	30.5 (+3.0)	37.6 (+2.9)	41.1 (+2.9)	57.6 (+5.2)	70.5 (+4.9)	76.5 (+4.7)
	+ SMS Filtering	**32.7 (+5.2)**	**41.4 (+6.7)**	**45.3 (+7.1)**	**61.4 (+9.0)**	**76.9 (+11.3)**	**83.5 (+11.7)**
Replica	Ours w/ CLIP	22.4	30.0	35.6	42.8	57.1	67.7
	+ Alpha-CLIP	25.1 (+2.7)	33.7 (+3.7)	41.7 (+6.1)	47.6 (+4.8)	63.8 (+6.7)	78.0 (+10.3)
	+ SMS Filtering	**25.7 (+3.3)**	**34.9 (+4.9)**	**42.3 (+6.7)**	**48.8 (+6.0)**	**66.3 (+9.2)**	**79.7 (+12.0)**
S3DIS	Ours w/ CLIP	29.4	39.9	45.4	45.4	60.5	67.2
	+ Alpha-CLIP	31.0 (+1.6)	43.1 (+3.2)	49.9 (+4.5)	47.9 (+2.5)	64.7 (+4.2)	72.5 (+5.3)
	+ SMS Filtering	**31.3 (+1.9)**	**43.5 (+3.6)**	**50.4 (+5.0)**	**48.2 (+2.8)**	**65.1 (+4.6)**	**72.9 (+5.7)**

Table 4. **Impact of Alpha-CLIP in instance classification on the ScanNet200, Replica, and S3DIS datasets.**

floors, ceilings, walls, and columns. Also, in the case of S3DIS, some instances have incomplete masks for large objects, which could be a side effect of our refinements. This is the limitation of our method, and solving this problem remains our future work.

Qualitative Results with the New Text Queries. We visualize more examples of OV-3DIS using new text queries on the ScanNet200 dataset in Fig. 7. Our method successfully retrieves corresponding instances based on functional descriptions and object attributes such as color, brand name, and other features.

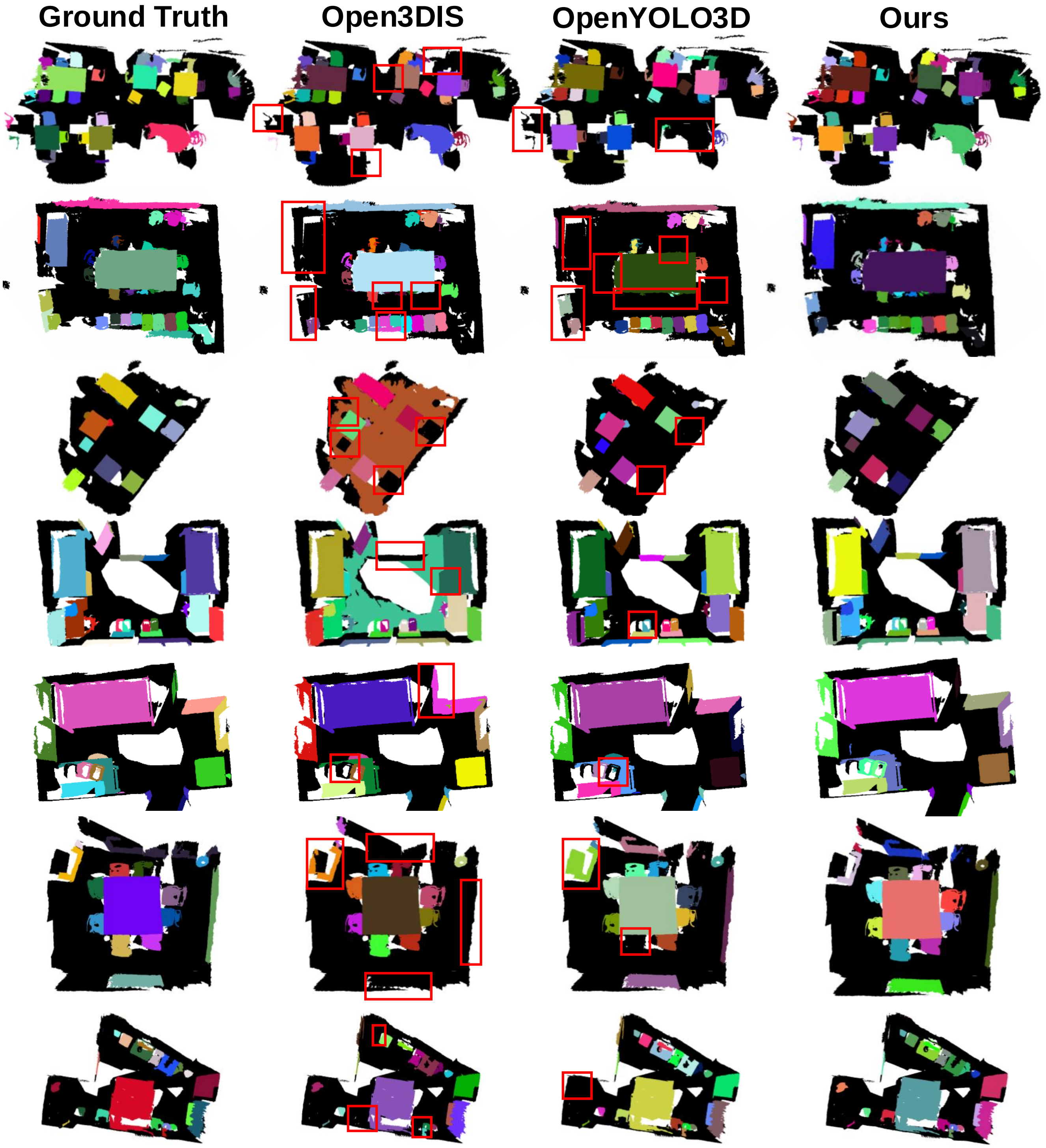

Figure 4. **Exteneded qualitative comparisons on the ScanNet200 dataset.** Black regions indicate empty predictions (*no object*), while red boxes highlight objects missed by other methods but successfully detected by ours. 3D instance masks are colored randomly.

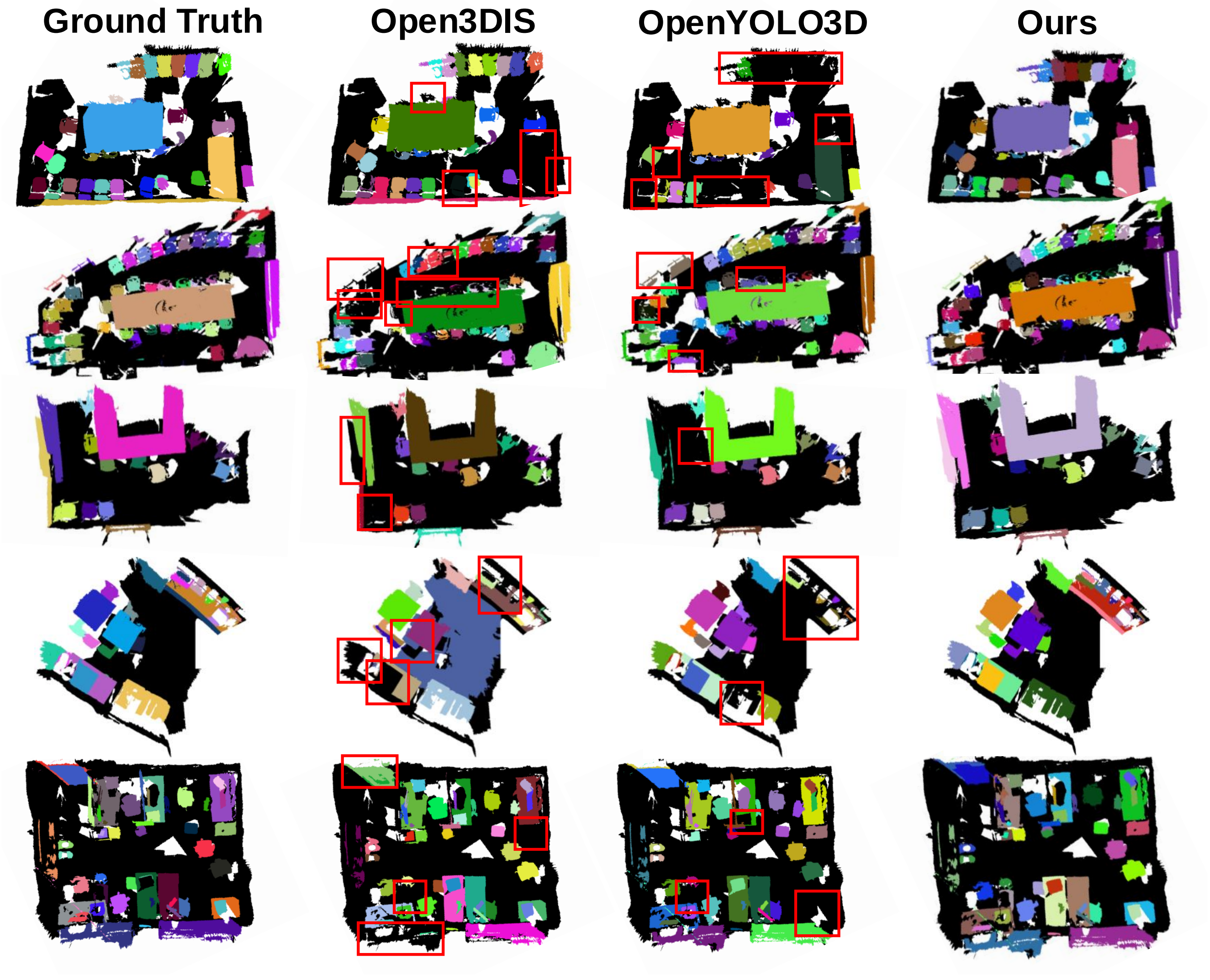

Figure 5. **Exteneded qualitative comparisons on the ScanNet200 dataset.** Black regions indicate empty predictions (*no object*), while red boxes highlight objects missed by other methods but successfully detected by ours. 3D instance masks are colored randomly.

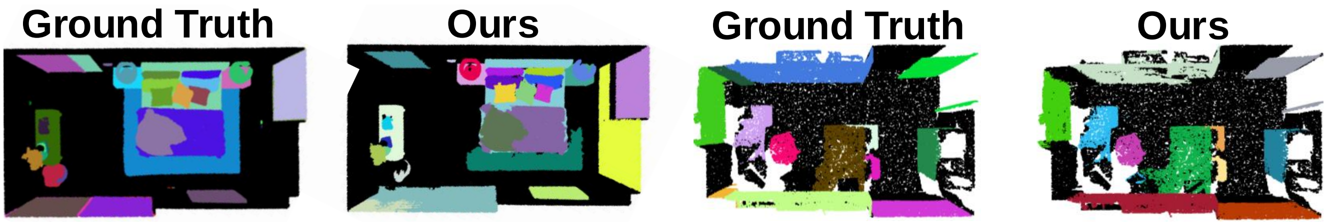

Figure 6. **Qualitative results of our method on the Replica [9] (left) and S3DIS [1] (right) datasets.** Black regions indicate empty predictions (*no object*). 3D instance masks are colored randomly.

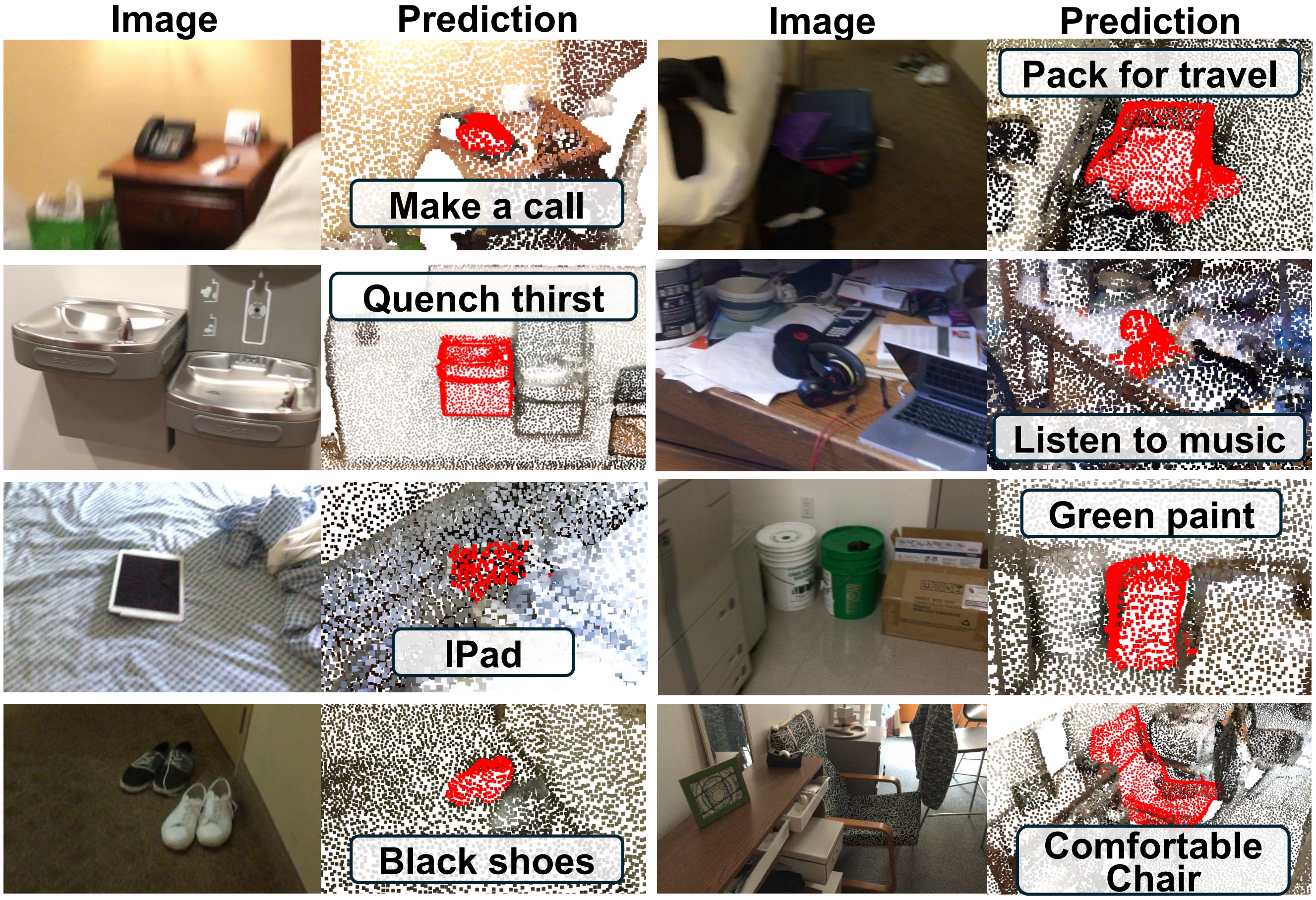

Figure 7. **Exteneded OV-3DIS results with new text queries on the ScanNet200 dataset.** Our method effectively retrieves instances based on functional descriptions and object attributes.

References

[1] Iro Armeni, Ozan Sener, Amir R Zamir, Helen Jiang, Ioannis Brilakis, Martin Fischer, and Silvio Savarese. 3d semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1534–1543, 2016. 3, 6

[2] Mohamed El Amine Boudjoghra, Angela Dai, Jean Lahoud, Hisham Cholakkal, Rao Muhammad Anwer, Salman Khan, and Fahad Shahbaz Khan. Open-yolo 3d: Towards fast and accurate open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2406.02548*, 2024. 1, 3

[3] 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. 3

[4] Haoyu Guo, He Zhu, Sida Peng, Yuang Wang, Yujun Shen, Ruizhen Hu, and Xiaowei Zhou. Sam-guided graph cut for 3d instance segmentation. In *European Conference on Computer Vision*, pages 234–251. Springer, 2024. 2

[5] Rui Huang, Songyou Peng, Ayca Takmaz, Federico Tombari, Marc Pollefeys, Shiji Song, Gao Huang, and Francis Engelmann. Segment3d: Learning fine-grained class-agnostic 3d segmentation without manual labels. In *European Conference on Computer Vision*, pages 278–295. Springer, 2024. 2

[6] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 1

[7] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4018–4028, 2024. 2, 3

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[9] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. 6

[10] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alpha-clip: A clip model focusing on wherever you want. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13019–13029, 2024. 1

[11] Ayça Takmaz, Elisabetta Fedele, Robert W Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. *arXiv preprint arXiv:2306.13631*, 2023. 1

[12] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023. 2

[13] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2023. 2

[14] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3292–3302, 2024. 2