MaskClustering: View Consensus based Mask Graph Clustering for Open-Vocabulary 3D Instance Segmentation

Supplementary Material

1. Overview

In this supplementary material, we begin by detailing the advantages of view consensus-based mask clustering in comparison to geometric overlap and semantic similarity in Sec. 2. Following that, in Sec. 3, we introduce additional clustering baselines to demonstrate the superiority of our iterative clustering algorithm. To offer a more comprehensive understanding of our approach, we delve into additional experimental details in Sec. 6 and elaborate on implementation details in Sec. 5. Further, in Sec. 4, we present additional qualitative results.

2. Discussion about View Consensus Rate

2.1. Comparison with Geometric Overlap

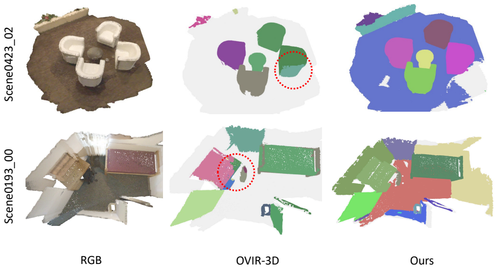


Figure 1. **Failure cases exemplifying over-segmentation in the geometric overlap-based method.**

As stated in Section 3.2.1 of the main paper, we observe that the merging of masks relying on geometric overlap may lead to over-segmentation errors. As illustrated in Fig. 1, such errors are evident, including the over-segmentation of the side of an armchair in the first row and the corner of a desk in the second row. In this section, we provide comprehensive statistics to explain why our proposed method is more effective in addressing these specific scenarios.

2.1.1 Case Study: Armchair Over-segmentation

Let's consider the armchair instance in the first row as a case study. To streamline notation, we will use m_i here to represent each mask instead of the double index $m_{t,i}$ as used in the main paper. Examining Fig. 2, the blue mask m_1 captures the side of the armchair, while the red mask m_2 captures its frontal view. The geometric Intersection-over-Union (IoU) between them is merely 0.012, falling significantly below the 0.25 threshold employed by OVIR-3D, rendering their merger challenging. Despite the inclusion of the third-view green mask m_3, merging the blue and red masks remains challenging because their overlaps with the third-view mask are still low (0.044 and 0.097, respectively).

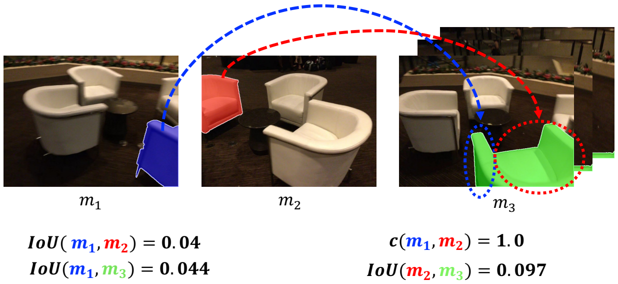


Figure 2. **Case study.** Masks belonging to a same instance may display low geometric overlap but exhibit a high view consensus rate. The blue and red masks represent the side and frontal views, respectively, of the same armchair. Despite their low geometric overlap, both masks are visible in the rightmost frame and are contained by the same green mask, resulting in a high consensus rate.

In contrast, our view consensus metric effectively utilizes third-view observations. In Fig. 2, both masks are visible in the rightmost view, and they are encompassed by the green mask m_3 (highlighted by arrows and circles of matching colors). Consequently, this third view supports for merging these two masks. In total, these masks co-occur in 42 frames, receiving unanimous support, resulting in a perfect 42/ 42 consensus rate.

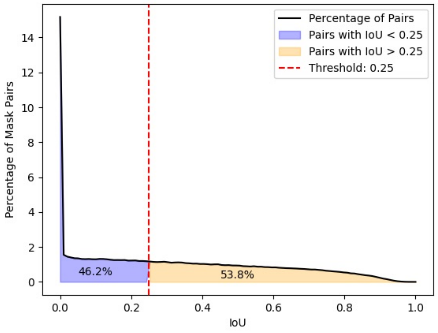


Figure 3. **Distribution of Intersection over Union (IoU) for positive mask pairs**. 46.2% of mask pairs belonging to the same instance exhibit low IoU, contributing to the over-segmentation phenomenon observed in the geometric overlap-based method.

2.1.2 More Statistics

We present additional statistics to illustrate why a geometric overlap-based method tends to result in over-segmentation of objects. Using the validation set of ScanNet, we identify all positive mask pairs, meaning they correspond to the same object based on ground truth annotations. We then calculate the IoU for each pair and depict the distribution in Fig. 3. Notably, while 53.8% of pairs exhibit high geometric overlap, 46.2% have significantly lower IoU. Moreover, 15.2% of positive pairs demonstrate no geometric overlap. This is particularly common for masks corresponding to large objects, such as the two ends of a table, the front and back faces of a chair, or the left and right sides of a bed.

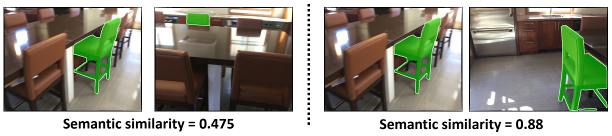
2.2. Comparison with Semantic Similarity

Previous work[2, 4] all use semantic similarity between two masks as a clue to decide whether they belong to a same object. In this section, we introduce an extra experiment to assess the influence of this semantic clue. Specifically, we begin by extracting the CLIP feature from the original RGB image around each mask, considering it as the semantic feature for that mask. Subsequently, we establish a connection between masks only when their consensus rate exceeds τ_{rate} and their semantic similarity surpasses $\tau_{seman} = 0.6$.

Table 1. **Effect of semantic clue on mask clustering.** Incorporating semantic similarity as an additional criterion yields only marginal performance improvement.

	AP	AP_{50}	AP_{25}
Ours	12.0	23.3	30.1
Ours + semantics	**12.1**	**23.5**	**30.2**

Table 1 illustrates the results, indicating that the contribution of semantics is relatively modest: a mere increase of +0.1 in AP, +0.2 in AP_{50}, and +0.1 in AP_{25}. Given that this enhancement is accompanied by a temporal cost, we opt not to include semantic similarity as an additional criterion.


Semantic similarity = 0.475

Semantic similarity = 0.88

Figure 4. **Instances of Semantic Similarity Failures.** On the left, the side and frontal view of the same chair exhibit low similarity. On the right, all chairs in a room appear identical, causing different chairs to have high similarity.

Fig. 4 highlights typical cases where semantic similarity proves unreliable. In line with Fig. 3, we present detailed statistics to elucidate this unreliability. Positive and negative mask pairs are identified based on their correspondence to the same object, according to ground truth annotations. We then calculate the semantic similarity for each pair, illustrating the distributions in Fig. 5. The substantial overlap in these distributions indicates that negative pairs can have high similarity, while positive pairs may exhibit low similarity. This overlap poses challenges in utilizing semantic similarity as a dependable criterion for determining the relationship between two masks.

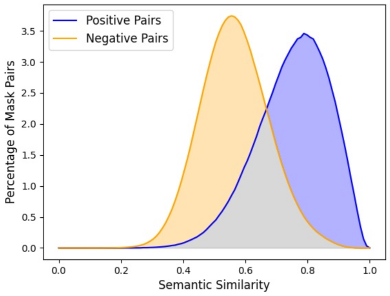


Figure 5. **Semantic Similarity Distribution for Positive and Negative Mask Pairs.** The distributions exhibit substantial overlap, indicating that negative pairs can possess high similarity, while positive pairs may exhibit low similarity. This overlap complicates the use of semantic similarity to reliably determine the relationship between two masks.

3. Discussion about Clustering Methods

In Sec. 3.3 of the main paper, we provide a concise explanation for our adoption of iterative clustering. In this section, we present supplementary experiments to further justify this selection.

For the sake of clarity in our subsequent discussions, let us introduce several key graph theory terms:

- **Connected component** is a set of vertices in a graph that are interconnected by paths.
- **Clique** is a set of vertices in a graph where there exists an edge between every pair of vertices.
- **Clique cover** is a partition of the graph's vertices into cliques.

Table 2. **Comparison of different clustering methods.**

Clustering Algorithm	AP	AP_{50}	AP_{25}
Connected component	11.0	21.2	27.5
Clique	11.3	22.0	29.4
Quasi-Clique (HCS)	11.9	22.9	29.7
Ours w/o approximation	11.8	23.1	**30.4**
Ours	**12.0**	**23.3**	30.1

Here, we present several distinct clustering strategies.

Connected Component. Instead of employing the iterative approach of merging connected components and updating edges, we execute this algorithm just once. As depicted in Table 2, all metrics exhibit a substantial decline when compared to our iterative version. Fig. 6 illustrates that this relaxed connectivity requirement leads to severe under-segmentation, such as predicting the wall and floor as a single instance, the mirror frame mixed into the wall, and the floor drain merged with the floor.

Clique. To mitigate the issue of under-segmentation, a straightforward solution is to elevate the connectivity requirements of a cluster. A clique, representing a graph with maximal connectivity, serves as a potential solution. Consequently, we aim to identify a clique cover for effective clustering and merging of masks. The results presented in Table. 2 provide additional evidence supporting this enhancement of connectivity. However, this extreme requirement in connectivity can lead to over-segmentation at times, as illustrated in Fig. 6, where the wall is over-segmented into two pieces.

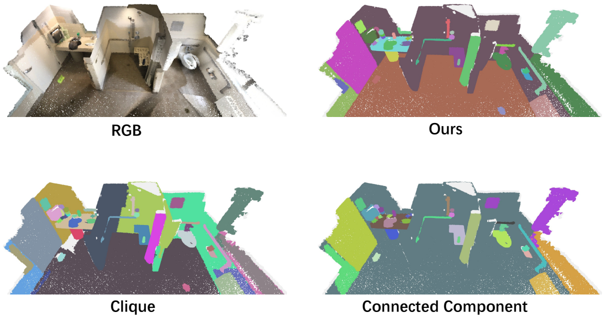


Figure 6. **Qualitative results of different clustering methods.** Clique-based clustering tends to over-segment, and single-time connected component-based clustering tends to under-segment. In contrast, our iterative clustering method yields perfect results.

Quasi-Clique (HCS). The stringent connectivity requirement often results in over-segmentation issues. In response, we explore a relaxation of the clique concept, allowing for a fraction of edges to be absent within each cluster, a condition known as quasi-cliques. The Highly Connected Subgraphs (HCS) clustering algorithm [3] is a standard method for efficiently partitioning a graph into such quasi-cliques. HCS defines a quasi-clique as a subgraph with n vertices, where the minimum cut of the subgraph contains more than n/2 edges. They demonstrate that these quasi-cliques exhibit properties similar to cliques. As illustrated in Table 2, this relaxation of the clique requirement enhances performance, yielding results slightly below those of our final version. Nevertheless, due to the necessity for HCS algorithm to iteratively recompute the minimum cut, its computational cost exceeds more than twice the time required by our algorithm.

Ours w/o approximation. In Section 3.3 of the main paper, we employ two approximations, namely $F(m_{\text{new}}) \approx F(m_{t_1,i_1}) \cup F(m_{t_2,i_2}) \ldots \cup F(m_{t_s,i_s})$ and $M(m_{\text{new}}) \approx M(m_{t_1,i_1}) \cup M(m_{t_2,i_2}) \ldots \cup M(m_{t_s,i_s})$, to speed up the edge updating process. In this section, we conduct additional experiments to demonstrate the impact of this approximation. Table 2 shows that the approximation has minimal effect on all metrics. However, the average time required to merge masks increases from 2.8 minutes to 6.9 minutes. Consequently, we opt to use the approximation as it provides a balance between speed and performance.

4. Additional Qualitative Results

We present enhanced qualitative results in Fig. 7. Our approach demonstrates the ability to accurately segment small objects, some of which may not be present in the ground truth. Additionally, our method exhibits consistent and robust performance when handling large objects, overcoming challenges faced by the geometric overlap-based method OVIR-3D.

5. Implementation Details

How do we obtain mask point cloud $P_{t,i}$? In this section, we elaborate on the methodology employed to derive the mask point cloud $P_{t,i}$. For every pixel (u, v) within this mask, given the camera intrinsic matrix $K \in \mathbb{R}^{3\times3}$ and extrinsic parameters $R \in \mathbb{R}^{3\times3}, T \in \mathbb{R}^3$, the back-projection of this pixel into 3D world space is accomplished using the following transformation:

$$(x\ y\ z)^T = R^{-1}\left(dK^{-1}\left(u\ v\ 1\right)^T - T\right), \tag{1}$$

where d is the depth value at pixel (u, v).

Subsequently, the obtained 3D point (x, y, z) is projected onto the reconstructed point cloud P. There are two reasons for this:

- **Format Alignment with Ground Truth**: Since the ground truth is annotated on the reconstructed point cloud, aligning the raw back-projected point cloud onto it is essential for accurate evaluation.
- **Efficient Computation**: By leveraging the globally-consistent point cloud P, we can utilize a list of indices within P to represent the masked point cloud. This transformation converts the subsequent time-consuming geometric operation into fast intersection and union operations on lists of indices.

Specifically, we use a ball query to identify all points on the reconstructed point cloud that are sufficiently close to the point (x, y, z) (less than 2cm for ScanNet and 3cm for MatterPort3D). The union of such points forms $P_{t,i}$.

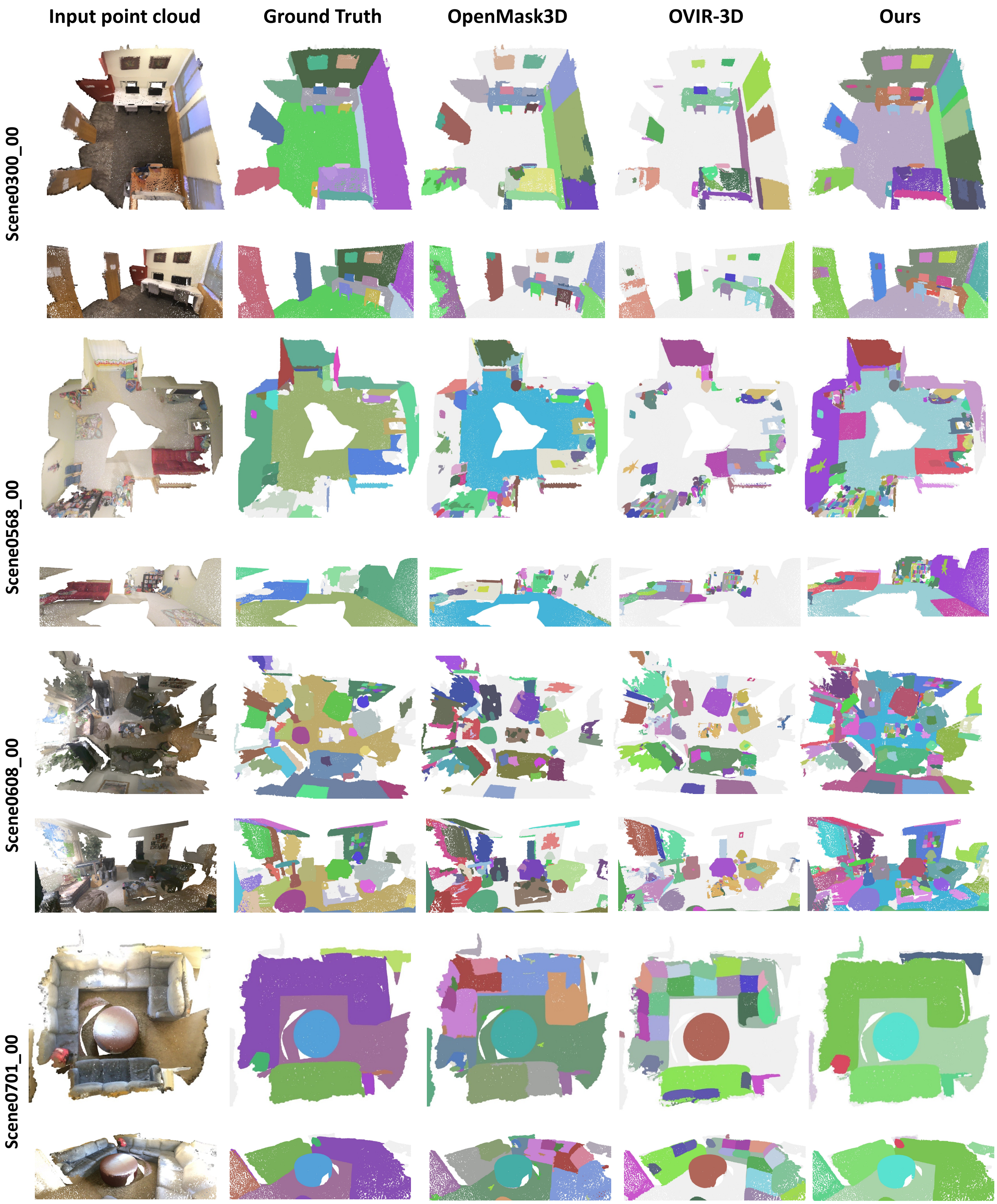


Figure 7. Comparison of 3D zero-shot segmentation performance. We compare our methods with OpenMask3D [6] and OVIR-3D [4] on ScanNet [1].

6. Experimental Details

6.1. Details about MatterPort3D Benchmark

We use the MatterPort3D test set as our benchmark and adopt the 160-category benchmark established by OpenScene [5]. As Mask3D encounters out-of-memory errors in 9 out of the total 17 scenes, our testing is consequently focused on the remaining 8 scenes: 2t7WUuJeko7, gxdoqLR6rwA, WYY7iVyf5p8, YVUC4YcDtcY, ARNzJeq3xxb, gYvKGZ5eRqb, RPmz2sHmrrY, YFuZgdQ5vWj.

To evaluate Mask3D on the MatterPort3D dataset, we map each label in ScanNet200 to its similar label in MatterPort3D, by calculating the similarity score between them using a natural language processing tool spaCy and manually removing the uncorrected matches. Labels that fail to match are tagged invalid. Finally, 164 labels in 200 ScanNet labels are mapped to 115 labels in 160 MatterPort3D labels.

6.2. Details about Hyperparameters

In our main paper, we use mask visibility threshold $\tau_{vis} = 0.3$, the under-segment mask filtering threshold $\tau_{filter} = 0.3$, the consensus rate threshold $\tau_{rate} = 0.9$ and the approximate containment threshold $\tau_{contain} = 0.8$. To demonstrate the robustness of our approach to these hyperparameters, we conduct a series of experiments illustrated in Fig. 8. The results reveal that even when each parameter is varied within the range of ±0.2, the performance remains relatively stable.

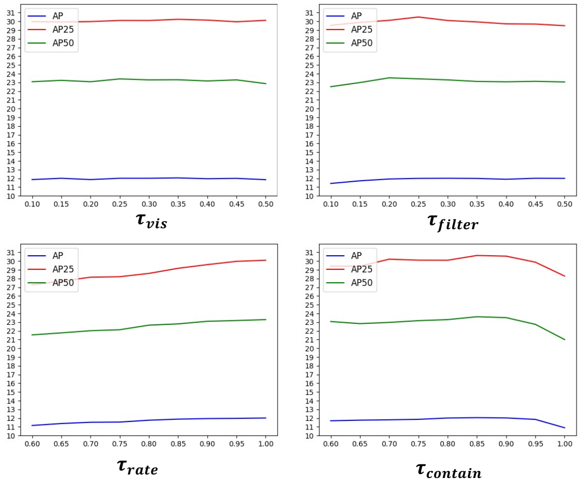


Figure 8. **Performance Variation with Changing Hyperparameters.**.

References

[1] 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

[2] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Ramalingam Chellappa, Chuang Gan, Celso Miguel de Melo, Joshua B. Tenenbaum, Antonio Torralba, Florian Shkurti, and Liam Paull. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. *ArXiv*, abs/2309.16650, 2023. 2

[3] Erez Hartuv and Ron Shamir. A clustering algorithm based on graph connectivity. *Information processing letters*, 76(4-6):175–181, 2000. 3

[4] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas E. Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. *ArXiv*, abs/2311.02873, 2023. 2, 4

[5] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023. 5

[6] 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. 4