PS3: Part level instance segmentation in 3D

Hong-Xuan Yen[1*] Chiamin Chen[1*] Yanqing Wang[1] Yu-Lun Liu[2] Min Sun[1]
[1]National Tsing Hua University [2]National Yang Ming Chiao Tung University
{felix.hongxuan.yen, cjm108061535, s112061518ee}@gapp.nthu.edu.tw
yulunliu@cs.nycu.edu.tw sunmin@ee.nthu.edu.tw

Abstract

Open-vocabulary 3D segmentation allows exploration of 3D environments using unrestricted natural language queries. Current approaches to open-vocabulary 3D instance segmentation largely concentrate on recognizing object-level instances but face difficulties when dealing with more fine-grained elements of a scene, such as object parts. Some previous work constructs hierarchical open-vocabulary 3D scene representations by geometric over-segmentation, which can't identify parts with similar geometry. In this work, we introduce PS3, an approach to generate 3D part proposals from multi-view 2D masks. PS3 outperforms baselines that rely on geometric over-segmentation in scene-scale open-vocabulary 3D part segmentation.

1. Introduction

Understanding the semantic structure of 3D environments is a core requirement for embodied AI and robotic systems operating in unstructured human-centric spaces. Conventional 3D segmentation pipelines, however, are usually trained on datasets with fixed taxonomies and thus remain constrained to closed-set recognition. This reliance on pre-defined labels limits their ability to recognize novel objects or scene elements that naturally occur in diverse environments. For practical applications such as assistive robotics, the ability to segment entities described in free-form language — including unseen objects, object parts, and regions defined by attributes — is essential.

Open-vocabulary 3D segmentation has recently emerged as a solution to this challenge, enabling the use of text queries to discover arbitrary scene entities. Existing approaches vary in how they represent 3D scenes: some adopt compact object-centric representations that align well with open-ended queries at the instance level, while others rely on per-point semantic encodings to achieve finer granularity. While the former are efficient for identifying whole objects, they often lack the flexibility to segment parts or attributes. The latter, despite being more detailed, comes with drawbacks such as high memory cost, noisy features, and limited instance-level reasoning.

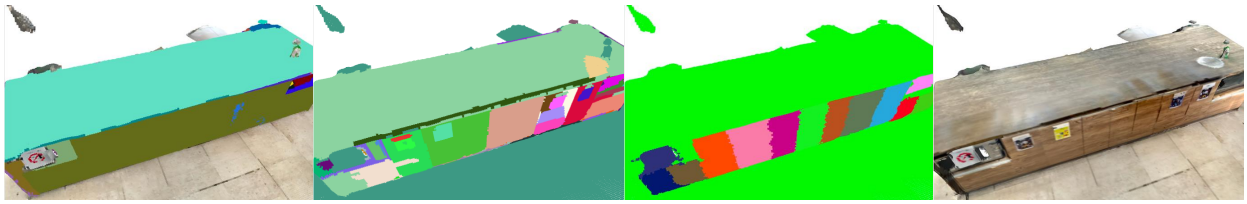
To bridge this gap, Search3D[24] introduced a hierarchical framework that leverages geometric over-segmentation to generate 3D proposals, thereby supporting both object- and part-level queries. This method demonstrated that moving beyond a purely object-centric perspective is feasible and beneficial for open-vocabulary 3D segmentation.

Nevertheless, Search3D's dependence on geometry-based over-segmentation introduces an important weakness: it struggles when different parts share highly similar geometry. For instance, the multiple doors of a cabinet that lie on the same flat surface may collapse into a single proposal, making it impossible to distinguish them individually. Such failures are detrimental in scenarios where robots must interact with specific components, materials, or surfaces, as precise segmentation is often critical for the successful execution of downstream tasks.

In this work, we aim to overcome these limitations by developing a framework that extends open-vocabulary 3D segmentation beyond geometry-driven proposals. Our approach seeks to robustly separate geometrically similar parts — thereby moving closer to the requirements of real-world embodied applications. To summarize our key contributions:

- We introduce a superpoint generation strategy driven by texture cues, which is helpful to distinguish parts that are similar in geometry.
- We propose a masklet-consensus merging strategy to generate part-level proposals instead of only relying on geometric over-segmentation.
- Our approach outperforms baselines on open-vocabulary 3D scene-scale part-level segmentation.

*Equal contribution



(A) Search3D (B) Ours (C) GT (D) Mesh

Figure 1. **Limitation of geometric over-segmentation.** In (A) to (C), each color represents one class-agnostic part-level 3D proposal. Search3D[24] relies on using geometric over-segmentation to generate proposals. However, this fails when multiple parts share similar geometry (e.g., located on the same plane). As shown in the figure, Search3D[24] fails to identify the doors of the cabinet. In contrast, our results are much more similar to the GT annotations.

2. Related work

2.1. Hierarchical 3D scene understanding

Early studies on open-vocabulary 3D understanding reveal two contrasting directions. Some methods emphasize compact, object-level representations that align well with vision-language models, such as OpenMask3D[23], Open3DIS[19], and OpenYOLO3D[2]. These approaches are efficient for retrieving object instances that match arbitrary text queries but are constrained to whole-object predictions and cannot adapt to finer levels of detail. Other approaches instead favor dense, per-point representations or implicit neural fields. Examples include OpenScene[20], ConceptFusion[9], LeRF[10], and OpenNeRF[6]. These methods capture more detailed semantics that can in principle distinguish object parts, yet they lack a structured, layered representation of scenes and often suffer from noise, memory demands, or limited interpretability.

More recently, research has explored multi-granularity and hierarchical reasoning in 3D. N2F2[1] demonstrates that hierarchical features can be embedded in neural fields through Gaussian splatting, although it does not explicitly enable part-instance segmentation. Other frameworks highlight multi-granularity grouping (GARField[11]), task-driven sub-part discovery (SceneFun3D[5]), or interactive segmentation interfaces (AGILE3D[28]). Segment3D[8] extends the success of foundation models like SAM[12] into the 3D domain, offering segmentation at multiple scales, but still without a principled hierarchical organization that supports language-guided querying.

A different line of work targets part-level open-vocabulary segmentation. Methods in this space [3][16][15] have mostly concentrated on single objects, operating on isolated point clouds instead of full-scale indoor scenes. In addition, systems relying on models such as GLIP[14] must define queries at representation-construction time, which forces expensive reprocessing whenever new queries are introduced. By contrast, building intermediate feature hierarchies offers a more efficient solution, as queries can be answered flexibly at inference without retaining all input images.

Closest to our work is Search3D[24], which proposes a hierarchical open-vocabulary 3D segmentation framework that bridges object-level and part-level queries. Its central idea is to generate proposals through geometric over-segmentation and to embed them hierarchically for text-based querying. While effective for separating many objects and parts, this reliance on geometry introduces a key weakness: when multiple parts share similar shapes or surfaces — for example, the identical doors of a cabinet arranged on the same plane — they are not distinguished correctly. Our work seeks to overcome this limitation by moving beyond purely geometry-driven proposals and enabling more robust identification of fine-grained parts under open-vocabulary queries.

2.2. Training-free 3D instance segmentation

A large body of recent work derives 3D instances by lifting 2D masks across views. SAM3D[29] uses SAM[12] to obtain 2D masks, projects them into the reconstructed point cloud, and merges them iteratively in a bottom-up fashion to produce training-free 3D instances. However, inconsistencies in 2D segmentation across RGB-D frames hinder their robustness, as they rely on heuristic merging criteria. This is especially true in using part-level masks(e.g., high-granularity masks generated by Semantic-SAM[13]). MaskClustering[25] reframes multi-view fusion as graph clustering: each 2D mask is a node, edges are weighted by a view-consensus rate that measures how often two masks co-occur inside other views' masks. Clustering this global mask graph yields 3D instances without training. SAI3D[27], and Open3DIS[19] utilize models such as SAM[12] to produce dense mask predictions for every 2D RGB-D frames, which are then aggregated into a unified set of 3D proposals. Any3DIS[18] leverages the power of a strong 2D tracking model, SAM2[21], and lifts the

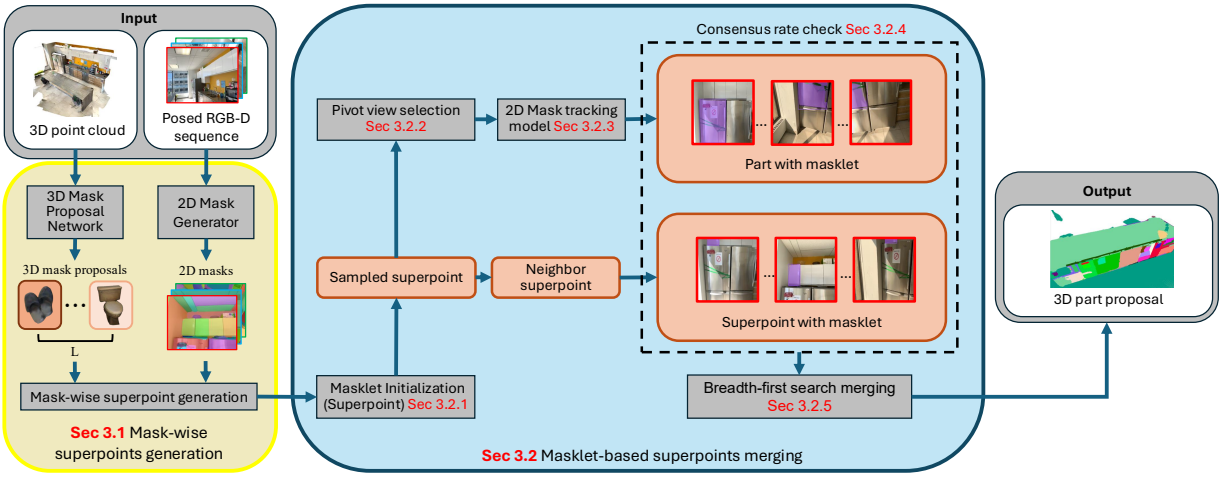


Figure 2. **Overview of PS3.** We propose a novel approach to generate 3D part-level proposals. First, we don't rely on geometric over-segmentation to generate superpoints. In contrast, we rely on 2D masks from Semantic-SAM[13] to generate superpoints that reflect textures. Second, we merge these superpoints based on masklets. We assign each superpoint a masklet that represents its corresponding 2D masks at the superpoint level. Then, we sample a superpoint and get the mask that represents the part at its pivot view by considering the viewpoint. Next, we propagate the mask to other frames and get the masklet that represents the part by SAM2[21] tracking. Finally, we check whether the neighbor superpoints should be merged with the sampled superpoint based on the relationship between these masklets.

video tracks to 3D space. However, these methods are built upon using geometric over-segmentation to get superpoints, which is not appropriate for the part-level instance segmentation.

2.3. 2D segmentation model

SAM[12] popularizes promptable image segmentation: given simple prompts (points, boxes, or coarse masks), it returns high-quality, class-agnostic masks. However, SAM[12] does not explicitly control granularity: a prompt typically yields a valid mask, but not necessarily one aligned to a desired level of detail (e.g., a whole chair vs. its backrest).Semantic-SAM[13] extends this paradigm with controllable granularity, enabling a single model to generate masks that correspond to entire objects or finer parts from the same image, depending on how it is prompted. It can return masks at multiple levels of detail, providing a consistent pool of object- and part-level masks. Semantic-SAM[13] provides the object/part masks we need on individual frames, but its outputs can vary across frames of the same sequence. SAM2[21] generalizes promptable segmentation from single images to image sequences/videos. Given an initial prompt (point/box/mask) on a key frame, SAM2[21] maintains identity-consistent tracks across subsequent frames. We leverage the tracking ability of SAM2[21] to generate a masklet and use it as criteria for superpoint merging.

3. Method

Problem Definition: Given a 3D scene point cloud $\{\mathbf{P}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 6}$ with N points, each point is defined by its spatial coordinates and color values (x, y, z, r, g, b). Additionally, we have T RGB-D frames with the shape of (H, W), each consisting of a color image $\{\mathbf{I}_t\}_{t=1}^{T}$ and a depth image $\{\mathbf{D}_t\}_{t=1}^{T}$, where $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$ and $\mathbf{D}_t \in \mathbb{R}_{+}^{H\times W}$. Every frame t also has the extrinsic matrix $\mathbf{E}_t$ and shares the same intrinsic matrix $\mathbf{K}$ to support the calculation of corresponding pixel projections from the point cloud. We get L 3D object binary masks from Mask3D[22], represented as $\{\mathbf{M}_l^{object}\}_{l=1}^{L}$, where each mask is a binary mask $\mathbf{M}_l^{object} \in \{0,1\}^N$ Our goal is to segment all Q 3D part binary masks from $\{\mathbf{M}_l^{object}\}_{l=1}^{L}$, represented as $\{\mathbf{M}_q^{part}\}_{q=1}^{Q}$, where each mask is a binary mask $\mathbf{M}_q^{part} \in \{0,1\}^N$.

Overview: An overview of PS3 is shown in Fig. 2. Our approach leverages the concept of **3D instance segmentation by 2D masklet**. First, in Sec. 3.1, we aim to generate superpoints that reflect texture information to replace geometry-based superpoints. In Sec. 3.2, these superpoints are then assigned a masklet. Finally, we apply masklet-based merging on these superpoints to form the 3D part binary masks $\{\mathbf{M}_q\}_{q=1}^{Q}$.

3.1. 2D Mask-wise superpoints generation

The goal of this step is to generate superpoints based on the texture in the RGB-D sequence. One main challenge of using geometric over-segmentation like Search3D[24] to generate 3D part masks is the limited ability to identify different parts with similar geometry attributes (e.g., door of a cabinet, geometric segmentator[4] tend to identify multiple doors of a cabinet as a single part). To address this challenge, we leverage the rich texture in RGB images to generate superpoints that represent parts more precisely.

To begin, for each 3D object binary mask $\mathbf{M}_l^{object}$, we get its 2D set of projected points ρ_t^l on frame t by projecting function $\Pi(\cdot)$ as:

$$\rho_t^l = \Pi(\{\mathbf{P}_i\}_{i=1}^N, \mathbf{M}_l^{object}, \mathbf{K}, \mathbf{E}_t, \mathbf{D}_t) \tag{1}$$

Then, for each frame $\mathbf{I}_t$, we repeat the following process. We first get the 2D binary masks $\{\mathbf{m}_j\}_{j=1}^J$ corresponding to $\mathbf{I}_t$ by passing $\mathbf{I}_t$ to Semantic-SAM[13]. For each 2D binary mask, we get the projected points ρ_j^t that are located in the mask by pixel-wise AND, then we back-project ρ_j^t to 3D space and get the generated superpoint. However, some 2D binary masks are usually less meaningful, especially the ones that are located on the boundary of the image. To ensure the generated superpoints are representative, we skip the masks if the projected points ρ_j^t are located on the boundary of the image. We check this by the function $\mathbb{I}_{\text{bdry}}$ as:

$$\mathbb{I}_{\text{bdry}}(\rho_j^t) = \begin{cases} 0, & \text{if } \frac{1}{|\rho_j^t|}\sum_{\rho\in\rho_j^t}\mathbb{B}(\rho) > \tau_{bdry}, \\ 1, & \text{otherwise}, \end{cases} \tag{2}$$

Function $\mathbb{B}(\rho)$ is an indicator function = 1 if pixel ρ lies on the boundary of the image, else 0. The full process of generating superpoints is shown in Algorithm 1

3.2. Masklet-based superpoints merging

Since the superpoints from the previous step are generated based on a single image individually, this means the superpoints may not cover the whole part. Therefore, we conduct merging on these superpoints to form the final 3D binary masks that represent the part. In this step, we build a masklet for each superpoint and merge them based on the relationship between their masklets.

3.2.1. Superpoint masklet initialization.

For each superpoint $\mathbf{S}_r$, we project its 3D points into 2D and get the projected points $\{\rho_t^r\}_{t=1}^T$. For each frame, we get the 2D masks $\{\mathbf{m}_k\}_{k=1}^K$ by passing image $\mathbf{I}_t$ into Semantic-SAM[13] with granularity prompt $\mathcal{G}_{sp}$. Then, we append the mask $\mathbf{m}_{k^*}$ where most projected points are located to the masklet that corresponds to the superpoint.

$$k^* = \arg\max_{k\in\{1,\dots,K\}} \sum_{\rho\in\rho_t^r} \mathbb{I}(\rho \in \mathbf{m}_k) \tag{3}$$

Algorithm 1 Mask-wise superpoints generation.

Input Point cloud $\{\mathbf{P}_i\}_{i=1}^N$, 3D object masks $\mathbf{M}_l^{object}$, color images $\mathbf{I}$, depth $\mathbf{D}$,intrinsic $\mathbf{K}$, extrinsic $\mathbf{E}$
Output 3D superpoints $\mathbf{S} = \{\mathbf{S_r}\}_{r=1}^R$
Initial state $\mathbf{S} \leftarrow \phi$

for t in T **do**
  $\rho_t^l = \Pi(\{\mathbf{P}_i\}_{i=1}^N, \mathbf{M}_l^{object}, \mathbf{K}, \mathbf{E}_t, \mathbf{D}_t)$
  $\{\mathbf{m}_j\}_{j=1}^J$ = Semantic-SAM$\{\mathbf{I}_t\}$
  for $\mathbf{m}_j$ in $\{\mathbf{m}_j\}_{j=1}^J$ **do**
    $\rho_j^t = \rho_t^l \odot \mathbf{m}_j$
    if $\mathbb{I}_{\text{bdry}}(\rho_j^t) = 1$ **then**
      $\mathbf{S} \leftarrow \mathbf{S} \cup \Pi^{-1}(\rho_j^t, \mathbf{K}, \mathbf{E}_t, \mathbf{D}_t)$
    else
      pass
    end if
  end for
end for

Finally, for each superpoint, we get its masklet as $\{\mathbf{m}_t^{S_r}\}_{t=1}^T$

3.2.2. Viewpoint-dependent pivot view selection

Finally, we apply merging on these superpoints based on their masklet by BFS search. To start, we leverage the Farthest Point Sampling (FPS) to sample $\mathbf{Q}_1$ initial superpoints. For each sampled superpoint $\mathbf{S}^{sampled}$, we calculate the pivot view to set up tracking for the corresponding 3D part. Previous methods, like Open3DIS[19], select the pivot view only based on the number of 2D projected points on each frame. However, we found that the tracking results of SAM2[21] are heavily dependent on the mask of the initial image. In other words, we need the pivot view to focus on the part that we want to track. This implies that we require not only a large number of projected points on the frame, but also that these points cover the image's full extent, rather than being concentrated in a limited region. Therefore, we modify the strategy for picking the pivot view from (4) to (5). Fig. 3 shows the results of SAM2[21] tracking and the visualization of the final generated 3D part-level proposals when using different ways of pivot view selection. $\rho_t^{sampled}$ represents the projected points of superpoint $\mathbf{S}^{sampled}$ at frame t. $\mathbf{P}^{sampled}$ represents the corresponding 3D points. $A()$ is a function to get the area of the bounding box covered by the projected points. A_{img} is the area of the full image.

$$t^* = \arg\max_{t\in T} \left|\rho_t^{sampled}\right| \tag{4}$$

$$t^* = \arg\max_{t\in T} \left(\frac{|\rho_t^{sampled}|}{|P^{sampled}|} + \frac{A(\rho_t^{sampled})}{A_{\text{img}}} \right) \tag{5}$$

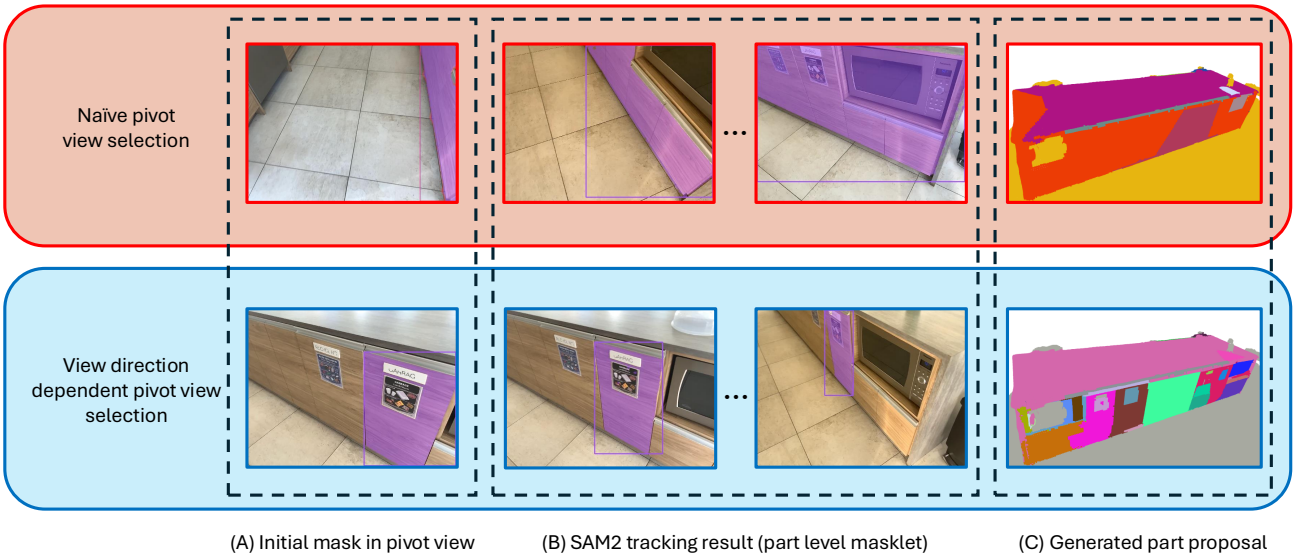


Figure 3. **Pivot view selection.** This figure shows the visualization of 2D tracking results and the generated 3D part-level proposals. When using the naive way of selection, the picked pivot view may not focus on the superpoint we sampled because it only takes into account the number of projected points. This causes wrong tracking results due to SAM2[21] being sensitive to the initial mask prompt. In contrast, our way of selection takes the viewpoint into consideration, which results in better tracking results and 3D part-level proposals.

3.2.3. Part masklet initialization

After picking the pivot view, we start to initialize the masklet that corresponds to the part as a criteria for merging. We check the relationship between the part masklet and the superpoint masklet to decide whether we should merge them as one part. For sampled superpoint $\mathbf{S}^{sampled}$. First, we get the 2D masks $\{\mathbf{m}_z\}_{z=1}^{Z}$ by passing image $\mathbf{I}_{t^*}$ into Semantic-SAM[13] with granularity prompt $\mathcal{G}_{part}$. Then pick $\mathbf{m}_{z^*}$ as the initial mask for SAM2[21] tracking based on the mask in pivot view $\mathbf{m}_{t^*}^{S^{sampled}}$ from the superpoint masklet $\{\mathbf{m}_t^{S^{sampled}}\}_{t=1}^{T}$.

$$z^* = \arg\max_{z\in\{1,\dots,Z\}} \sum_{\rho\in\mathbf{m}_{t^*}^{S^{sampled}}} \mathbb{I}(\rho \in \mathbf{m}_z)\,, \tag{6}$$

Then we conduct forward and backward tracking on the image sequence, and store the tracking result as part masklet $\{\mathbf{m}_t^{part}\}_{t=1}^{T}$.

$$\{\mathbf{m}_t^{part}\}_{t=1}^{T} = SAM2(\{\mathbf{I}_t\}_{t=1}^{T}, \mathbf{m}_{z^*}) \tag{7}$$

3.2.4. Consensus rate calculation

Once we have the part masklet of the sampled superpoint, we can check whether the neighbor superpoints should be merged with the sampled one to form a part by examining the relationship between the part masklet of the sampled one and the superpoint masklet of the neighbor ones. Here, we calculate the consensus rate $\mathbf{s}$ between these two masklets. If the consensus rate $\mathbf{s} > \tau_{\mathbf{merging}}$, we merge the neighbor one into the sampled one. Here, $\mathbf{S}_r$ represents the neighbor superpoint, and $\mathcal{T}$ represents the frames where both masklets exist.

$$\mathcal{T} = \{\, t \mid (\mathbf{m}_t^{part} \neq \varnothing) \wedge (\mathbf{m}_t^{\mathbf{S}_r} \neq \varnothing)\,\}_{t=1}^{T} \tag{8}$$

$$\begin{aligned} s &= \text{Cons}(\{\mathbf{m}_t^{part}\}_{t=1}^{T}, \{\mathbf{m}_t^{\mathbf{S_r}}\}_{t=1}^{T}) \\ &= \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathbb{I}\left(\frac{|\mathbf{m}_t^{part} \cap \mathbf{m}_t^{\mathbf{S}_r}|}{|\mathbf{m}_t^{\mathbf{S}_r}|} \geq 0.7\right) \end{aligned} \tag{9}$$

3.2.5. BFS merging

Now, we have defined the standard to decide whether two superpoints should be merged. We conduct merging by following the Breadth First Search strategy. We created a queue to store the neighbor superpoints to be checked and to check whether the neighbor superpoint should be merged with the sampled one. Once a superpoint has been successfully merged, we add its neighbor to the queue. We repeat this process until there is no superpoint that needs to be checked in the queue and get one 3D part binary mask $\mathbf{M}_q^{part}$. The full process is shown in Algorithm 2

3.2.6. Progressive 3D Part Sampling

This stage first produces $\mathbf{Q}_1$ candidate binary masks $\{\mathbf{M}_q\}_{q=1}^{\mathbf{Q}_1}$ as an initial set for evaluation. Yet, a portion of

superpoints remains unmatched with any mask and can indicate the presence of additional 3D parts. To recover these, the procedure is repeatedly applied until all free superpoints are assigned. The outcome is a complete set of $\mathbf{Q}$ part-level binary masks, given by $\mathbf{Q} = \mathbf{Q}_1 + \mathbf{Q}_2 + \cdots + \mathbf{Q}_n$.

Algorithm 2 Masklet-based superpoints merging.

Input 3D superpoints $\mathbf{S} = \{\mathbf{S_r}\}_{r=1}^{R}$, color images $\mathbf{I}$, depth $\mathbf{D}$,intrinsic $\mathbf{K}$, extrinsic $\mathbf{E}$
Output 3D part binary masks $\mathbf{M}^{part} = \{\mathbf{M}_q^{part}\}_{q=1}^{Q}$
Initial state $\mathbf{M}^{part} \leftarrow \phi$

for $\mathbf{S}_r$ in $\mathbf{S}$ **do** ▷ Masklet Initialization
  $\mathbf{m}^{S_r} \leftarrow \phi$
  for $\mathbf{t}$ in $\mathbf{T}$ **do**
    $\rho_t^r = \Pi(\mathbf{S}_r, \mathbf{K}, \mathbf{E}_t, \mathbf{D}_t)$
    $\{\mathbf{m}_k\}_{k=1}^{K}$ =Semantic-SAM$\{\mathbf{I}_t, \mathcal{G}_{sp}\}$
    $k^* = \arg\max_{k\in\{1,\ldots,K\}} \sum_{\rho\in\rho_t^r} \mathbb{I}(\rho \in \mathbf{m}_k)$
    $\mathbf{m}^{S_r} = \mathbf{m}^{S_r} \cup \mathbf{m}_{k^*}$
  end for
end for

$\mathbf{S}^{fps} = FPS(\mathbf{S})$
for $\mathbf{S}^{sampled}$ in $\mathbf{S}^{fps}$ **do** ▷ BFS merging
  $\rho^{sampled} = \Pi(\mathbf{S}^{sampled}, \mathbf{K}, \mathbf{E}, \mathbf{D})$
  $t^* = \arg\max_{t\in T} \left(\frac{|\rho_t^{sampled}|}{|P^{sampled}|} \cdot \frac{A(\rho_t^{sampled})}{A_{\text{img}}} \right)$
  $\{\mathbf{m}_z\}_{z=1}^{Z}$ =Semantic-SAM$\{\mathbf{I}_{t^*}, \mathcal{G}_{part}\}$
  $z^* = \arg\max_{z\in\{1,\ldots,Z\}} \sum_{\rho\in\mathbf{m}_{t^*}^{S^{sampled}}} \mathbb{I}(\rho \in \mathbf{m}_z)$
  $\{\mathbf{m}_t^{part}\}_{t=1}^{T} = SAM2(\{\mathbf{I}_t\}_{t=1}^{T}, \mathbf{m}_{z^*})$
  $\mathbf{bfs\_queue} \leftarrow \phi$
  ENQUEUE($\mathbf{bfs\_queue}$, $\mathbf{S}^{sampled}$)
  while $\mathbf{bfs\_queue} \neq \emptyset$ **do**
    $\mathbf{S}_{current} \leftarrow$ DEQUEUE($\mathbf{bfs\_queue}$)
    $s = \text{Cons}(\{\mathbf{m}_t^{part}\}_{t=1}^{T}, \{\mathbf{m_t^{S^{current}}}\}_{t=1}^{T})$
    if $s > \tau_{merging}$ **then**
      Merge $\mathbf{S}_{sampled}$ with $\mathbf{S^{current}}$
      ENQUEUE($\mathbf{bfs\_queue}$, NEIGHBOR($\mathbf{S}^{current}$))
    end if
  end while
  $\mathbf{M}^{part} = \mathbf{M}^{part} \cup \mathbf{S}^{sampled}$
end for

4. Experiments

4.1. Experimental Setup

Dataset: To evaluate the effectiveness of our proposed method, we conduct experiments on two datasets, MultiScan[17] and ScanNet++[26], adapted by Search3D[24] for open-vocabulary 3D part segmentation.

For MultiScan[17], Search3D[24] extends the limited original benchmark (17 objects, 5 parts) by regrouping fine-grained annotations into 155 object categories, 15 part categories, and 47 informative object–part pairs, enabling broader evaluation of scene-scale part segmentation.

For ScanNet++[26], Search3D[24] adapts 8 high-resolution laser scan scenes, providing 14 object categories and 20 part categories annotated with SceneFun3D[5], to support fine-grained object–part hierarchy evaluation at the scene level.

Evaluation Metrics: We evaluate our method using the standard AP metric, averaged over IoU thresholds ranging from 50% to 95% in steps of 5%. In addition, we report results at specific IoU thresholds of 50% (AP50) and 25% (AP25).

Implementation Details: We perform RGB-D sampling at an interval of 50 views. For Farthest Point Sampling (FPS), we sample 25% of the remaining superpoints at each round. For SAM2[21] tracking, we use the checkpoint "sam2.1_hiera_large". For open-vocabulary segmentation, we employ the SIGLIP[30] so400m model as describe in Search3D[24] for pointwise feature extraction.

4.2. Results on 3D part segmentation

In this part, we focus on the comparison between Search3D[24] and our method under the same evaluation protocol for open-vocabulary 3D part-level instance segmentation. Both methods use the identical inputs and the same "seg.+hierarchy" aggregation regime at inference time; the key difference is proposal formation: Search3D[24] builds proposals from geometric over-segmentation, whereas PS3 merged texture-informative superpoints based on the results generated by SAM2[21] tracking. Tab. 1 shows the results on multiscan. Compared to Search3D[24], PS3 improves AP from 7.9 to 8.6 (+0.7 abs., +8.9% rel.), AP50 from 14.5 to 20.1 (+5.6 abs., +38.6% rel.), and AP25 from 31.5 to 32.4 (+0.9 abs., +2.9% rel.). Fig. 4 shows the visualizations compared to Search3D. Compared to Search3D[24], our method identifies parts with similar geometry attributes more accurately.

Tab. 3 shows the results on Scannet++[26]. PS3 attains 33.2 AP50 and 41.9 AP25, outperforming Search3D[24] (32.4 / 38.3) at medium and coarse IoU thresholds, while Search3D achieves a higher AP (17.0 vs. 11.5). We found that our method fails to identify tiny parts (e.g., hook of wall hanging rack), which lowers our performance. Also, the test set of ScanNet++[26] only contains 15 labels, which is much less than Multiscan[17]. This means it is less representative than the results on Multiscan[17].

4.3. Ablation study

In this section, we validate the effectiveness of the components used in our pipeline. Tab. 2 shows the effectiveness of our mask-wise superpoints and viewpoint-dependent pivot view selection. Here we examine two components: (i)

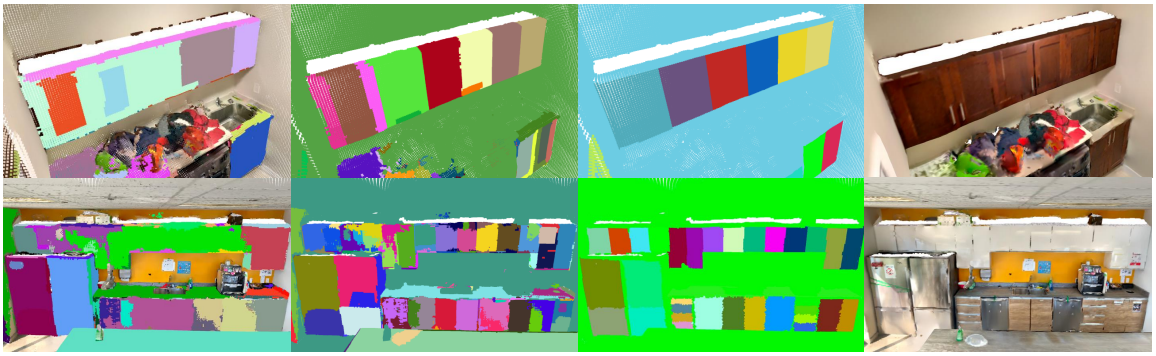


Figure 4. **Qualitative results on Multiscan.** From left to right, we show the results of Search3D[24] and our method, GT segmentations, and the color mesh. In (A) to (C), each color represents one class-agnostic part-level 3D proposal. Our approach achieves more accurate and consistent segmentation than Search3D[24].

Method	Aggregation	AP	AP50	AP25
(1)Openscene [20]	segments	3.2	5.5	13.7
(2)OpenMask3D [23]	objects	3.3	6.1	11.3
(3)OpenMask3D [23]	segments	3.1	6.2	18.2
(4)GarField[11]+Search3D[24]	segments	3.5	8.9	20.5
(5)GarField[11]+Search3D[24]	seg.+hierarchy	3.2	8.4	15.3
(6)Search3D[24]	seg.+hierarchy	7.9	14.5	31.5
(7)PS3 (Ours)	seg.+hierarchy	**8.6**	**20.1**	**32.4**

Table 1. **3D Part Segmentation on MultiScan.** We follow the number proposed in Search3D[24]. (1) uses 2D fused OpenSeg [7] feats., and per-point feats. are aggregated over part segments. (2) uses the orig. object-level masks from OpenMask3D[23]. (3) is a stronger baseline adapted from (2) using segment-level aggregation. (4) and (5) use object and part masks from GARField [11] at scales 0.35 and 0.1, respectively, and employ Search3D [24] for feature computation using these masks.

how we form 3D candidates—either from geometric over-segmentation (using the segmentator provided in [4]) or from mask-wise superpoints based on each 2D mask—and (ii) how we choose the pivot view—either the traditional heuristic that picks the view with the most projected points or our VD-PVS scheme. Using geometric over-segmentation with the Search3D[24] parameters yields 7.0 / 14.1 / 29.0. Making the segments finer with the segmentator's defaults nudges AP and AP50 to 7.6 / 15.6, but AP25 drops to 28.2. Replacing geometric over-segmentation with mask-wise superpoints already lifts performance to 8.1 / 17.5 / 32.3 even under the traditional pivot view selection, showing that texture-aligned superpoints are the main driver of gains. Adding VD-PVS on top of mask-wise superpoints delivers the best results—8.6 / 20.1 / 32.4—primarily by (AP50 +2.6 over the traditional pivot view selection). Overall, the transition from geometry-only to mask-guided superpoints is the dominant factor, and VD-PVS acts as a complementary refinement that further improves; together they yield a net improvement over the geometry-with-Search3D-params setting of +1.6 AP, +6.0 AP50, and +3.4 AP25.

5. Conclusion

In this paper, we present a novel approach to generating part-level 3D proposals without the need to rely on geometric over-segmentation. By using 2D masks, we solve the limitation of traditional geometric over-segmentation in identifying parts that share similar geometry attributes. Also, we leverage SAM2[21] to solve the multi-view inconsistency and propose a method to merge superpoints based on the relationship between masklets. Experiments on Multiscan[17] and ScanNet++[26]

Method	Mask-wise superpoints	VD-PVS	AP	AP50	AP25
(1)Ours		✓	7.0	14.1	29.0
(2)Ours		✓	7.6	15.6	28.2
(3)Ours	✓		8.1	17.5	32.3
(4)Ours	✓	✓	**8.6**	**20.1**	**32.4**

Table 2. **Ablation on MultiScan.** Mask-wise superpoints means whether using our superpoints from 2D masks or superpoints from geometric over-segmentation like Search3D. (1) follows the parameters mentioned in Search3D. (2) uses the default parameters of [4] to generate finer superpoints. VD-PVS means our proposed pivot view selection that takes the viewpoint into consideration.

Method	AP	AP50	AP25
OpenMask3D [23]	5.2	15.0	18.1
Search3d [24]	**17.0**	32.4	38.3
PS3 (Ours)	11.5	**33.2**	**41.9**

Table 3. **3D Part Segmentation on ScanNet++.**

validate the effectiveness of our method compared to the method using geometric over-segmentation.

References

[1] Yash Bhalgat, Iro Laina, Joao F Henriques, Andrew Zisserman, and Andrea Vedaldi. N2f2: Hierarchical scene understanding with nested neural feature fields. In *European Conference on Computer Vision*, pages 197–214. Springer, 2024. 2

[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. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[3] Tianrun Chen, Chunan Yu, Jing Li, Jianqi Zhang, Lanyun Zhu, Deyi Ji, Yong Zhang, Ying Zang, Zejian Li, and Lingyun Sun. Reasoning3d–grounding and reasoning in 3d: Fine-grained zero-shot open-vocabulary 3d reasoning part segmentation via large vision-language models. *arXiv preprint arXiv:2405.19326*, 2024. 2

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 4, 7, 8

[5] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 6

[6] Francis Engelmann, Fabian Manhardt, Michael Niemeyer, Keisuke Tateno, Marc Pollefeys, and Federico Tombari. OpenNeRF: Open Set 3D Neural Scene Segmentation with Pixel-Wise Features and Rendered Novel Views. In *International Conference on Learning Representations*, 2024. 2

[7] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European conference on computer vision*, pages 540–557. Springer, 2022. 7

[8] 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. *European Conference on Computer Vision (ECCV)*, 2024. 2

[9] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso Miguel de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. *Robotics: Science and Systems (RSS)*, 2023. 2

[10] Justin* Kerr, Chung Min* Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *International Conference on Computer Vision (ICCV)*, 2023. 2

[11] Chung Min* Kim, Mingxuan* Wu, Justin* Kerr, Matthew Tancik, Ken Goldberg, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 7

[12] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2, 3

[13] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, Lei Zhang, and Jianfeng Gao. Segment and recognize anything at any granularity. In *European Conference on Computer Vision*, pages 467–484. Springer, 2024. 2, 3, 4, 5

[14] Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. Grounded language-image pre-training. In *CVPR*, 2022. 2

[15] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21736–21746, 2023. 2

[16] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. *arXiv preprint arXiv:2411.13550*, 2024. 2

[17] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel X Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. In *Advances in Neural Information Processing Systems*, 2022. 6, 7

[18] Phuc Nguyen, Minh Luu, Anh Tran, Cuong Pham, and Khoi Nguyen. Any3dis: Class-agnostic 3d instance segmentation by 2d mask tracking. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3636–3645, 2025. 2

[19] Phuc D. A. 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 (CVPR)*, 2024. 2, 4

[20] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 7

[21] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. 2, 3, 4, 5, 6, 7

[22] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. 2023. 3

[23] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 2, 7, 8

[24] Ayca Takmaz, Alexandros Delitzas, Robert W. Sumner, Francis Engelmann, Johanna Wald, and Federico Tombari. Search3D: Hierarchical Open-Vocabulary 3D Segmentation. *IEEE Robotics and Automation Letters (RA-L)*, 2025. 1, 2, 4, 6, 7, 8

[25] 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. 2

[26] 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. 6, 7

[27] 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

[28] Yuanwen Yue, Sabarinath Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation. In *International Conference on Learning Representations (ICLR)*, 2024. 2

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

[30] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023. 6