S²AM3D: Scale-controllable Part Segmentation of 3D Point Clouds

Supplementary Material

This supplementary material presents additional implementation details (Section 1), additional experiments (Section 2), and more visualizations (Section 3).

Figure 1. Representative failure cases of S^{2}AM3D in challenging scenarios.

1. Implementation Details

Decoder. The continuous scale s is encoded with $M = 64$ sinusoidal frequency pairs and passed through a LayerNorm followed by a linear layer to produce the channel-wise FiLM [4] parameters $(\boldsymbol{\gamma}, \boldsymbol{\beta})$. The linear layer is initialized with zero weights and biases, so the FiLM branch starts with an identity transformation. We additionally use a global scaling factor α to modulate the FiLM output, which is initialized to 0.1.

Data pipeline. We train a PointNet-based [5] validator on a manually selected subset of 800 shapes (400 valid and 400 invalid), which is split into 70%/30% training and test sets. To rigorously exclude unqualified samples, we use a confidence threshold of 0.8, achieving a Precision/Recall of 1.00/0.78 on the test set. To mitigate category bias during validator training, we construct the manually labeled dataset via random, class-balanced sampling. The filtering criterion is primarily based on label distributions and is only weakly correlated with geometric structure, reducing the risk of shape-specific bias. We manually inspected approximately 10% of the final 100k-scale dataset, confirming the high quality and broad diversity of the resulting dataset. Overall, the filtering process retains about 60% of the shapes. For connectivity refinement, we apply DBSCAN [1] with radius $\varepsilon = d \times \varepsilon_{\text{factor}}$ using $\varepsilon_{\text{factor}} = 0.15$, where d is the diagonal length of the axis-aligned bounding box of the label, to split spatially disconnected components into distinct parts.

Full segmentation. We perform full segmentation in a simulated interactive setting. For each ground-truth part, we select a prompt point and use the same scale definition as in interactive segmentation. The model generates one candidate mask per part using the same configuration as in the interactive experiments. We then apply lightweight post-processing (see Algorithm 1): overlapping points are assigned to masks based on prediction confidence and distance to the mask centers, followed by k-NN label propagation to cover the remaining points, producing a full segmentation.

Algorithm 1 Post-Processing for Full Segmentation

Input: point cloud $\mathbf{X}$, candidate masks $\{\mathbf{m}_k\}$, point-wise scores $\{p_{ik}\}$, confidence–distance weight α, k-NN size k
Output: final disjoint masks $\{\tilde{\mathbf{m}}_j\}$
▶ **Overlap Resolution**
Compute geometric centers $\boldsymbol{\mu}_k$ for each candidate mask $\mathbf{m}_k$
for each point i **do**
  Let $\mathcal{J}_i = \{k \mid \mathbf{m}_k(i) = 1\}$ be the set of masks covering point i
  if $|\mathcal{J}_i| = 1$ **then**
    Assign point i to that single mask
  else if $|\mathcal{J}_i| > 1$ **then**
    For each $k \in \mathcal{J}_i$, compute

$$d_{ik} = \|\mathbf{X}_i - \boldsymbol{\mu}_k\|_2, \quad s_{ik}^{\text{conf}} = p_{ik}, \quad s_{ik}^{\text{dist}} = \exp(-d_{ik})$$

    Combine them as

$$\text{score}_{ik} = \alpha\, s_{ik}^{\text{conf}} + (1-\alpha)\, s_{ik}^{\text{dist}}$$

    Assign point i to the mask k with the highest score_{ik}
  end if
end for
▶ **k-NN Label Propagation**
Build a k-NN graph over $\mathbf{X}$
for $t = 1$ **to** 5 **do**
  for each point not assigned to any mask **do**
    Collect mask assignments of its k nearest neighbors and update its assignment by majority vote
  end for
end for
Optionally, for each i with $z_i = -1$, set $z_i \leftarrow z_{k^\star}$ where $k^\star = \arg\min_{k:\, z_k \neq -1} \|\mathbf{x}_i - \mathbf{x}_k\|_2$
Return disjoint masks $\{\tilde{\mathbf{m}}_j\}$ with $\tilde{\mathbf{m}}_j(i) = \mathbb{1}[z_i = j]$
return $\{\tilde{\mathbf{m}}_j\}$

2. Additional Experiments

2.1. Density Analysis

Table 1 presents the density analysis, reporting IoU at 10k and 100k input points on PartObjaverse-Tiny [6] and PartNet-E [2]. The results show consistent performance

Method	PartObjaverse-Tiny		PartNet-E	
	10k	100k	10k	100k
Ours	46.47	46.46	62.52	62.41
Ours (+scale)	61.19	60.98	77.51	77.67

Table 1. Effect of point density on S^2AM3D in the interactive setting.

Method	Params (M)	Time (ms)
Point-SAM [7]	311	$\sim$5
P^3-SAM [3]	112	$\sim$3
Ours	120	$\sim$3

Table 2. Complexity analysis of different methods.

at both densities: the impact on PartObjaverse-Tiny [6] is minimal, while higher density brings moderate gains on PartNet-E [2]. These results indicate that S^2AM3D is robust to variations in point density in practical settings.

2.2. Complexity Analysis

Table 2 presents the complexity analysis, reporting the inference time per point prompt in the interactive setting and the number of network parameters, measured on a single NVIDIA H20 GPU.

2.3. Scale Analysis

In the main experiments, the scale prompt s is set to the ground-truth part size for controlled evaluation. In practice, s may instead be provided by a user's rough estimate or by external modules. To assess sensitivity, we perturb the scale by $s' = (1 + \delta)s$ and measure the IoU change relative to $\delta = 0$. As shown in Table 3, on PartObjaverse-Tiny [6], perturbations within $\pm 20\%$ cause negligible performance change ($|\Delta \text{IoU}| < 1.0$). For moderate perturbations ($|\delta| \in 30\%$–50%), IoU decreases slowly, indicating gradual rather than catastrophic degradation. For larger perturbations ($|\delta| \geq 100\%$), the perturbed scale effectively corresponds to a different semantic granularity from the original annotation; in this range (100%–300%), the IoU drop is more pronounced, which is expected when still evaluating against the original ground-truth part.

2.4. Failure Analysis

Due to the frequent absence and low quality of textures in Objaverse, our training and evaluation are based on XYZ input. We visualize failure cases in Fig. 1, which mainly occur on extremely thin/long structures (Fail-1,2) or on geometry-similar parts that require color cues to distinguish (Fail-3, two wristbands merging together).

3. More Visualization Results

Additional qualitative results of S^2AM3D for full segmentation and interactive segmentation are shown in Figure 2. The method produces accurate and fine-grained part predictions across diverse object categories and geometric structures.

References

[1] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, pages 226–231, 1996. 1

[2] 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. 1, 2

[3] Changfeng Ma, Yang Li, Xinhao Yan, Jiachen Xu, Yunhan Yang, Chunshi Wang, Zibo Zhao, Yanwen Guo, Zhuo Chen, and Chunchao Guo. P3-sam: Native 3d part segmentation. *arXiv preprint arXiv:2509.06784*, 2025. 2

[4] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 1

[5] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[6] Yunhan Yang, Yukun Huang, Yuan-Chen Guo, Liangjun Lu, Xiaoyang Wu, Edmund Y Lam, Yan-Pei Cao, and Xihui Liu. Sampart3d: Segment any part in 3d objects. *arXiv preprint arXiv:2411.07184*, 2024. 1, 2, 3

[7] Yuchen Zhou, Jiayuan Gu, Tung Yen Chiang, Fanbo Xiang, and Hao Su. Point-sam: Promptable 3d segmentation model for point clouds. *arXiv preprint arXiv:2406.17741*, 2024. 2

$\lvert\delta\rvert$	5%	10%	20%	30%	50%	100%	200%	300%
ΔIoU ($-\delta$)	$+0.15$	$+0.12$	-0.05	-0.66	-2.99	-7.43	-14.85	-20.19
ΔIoU ($+\delta$)	-0.19	-0.38	-0.78	-1.48	-3.56	-7.47	-14.17	-18.98

Table 3. Effect of relative scale perturbation on mean IoU on PartObjaverse-Tiny [6]. ΔIoU is measured with respect to the $\delta = 0$ setting.

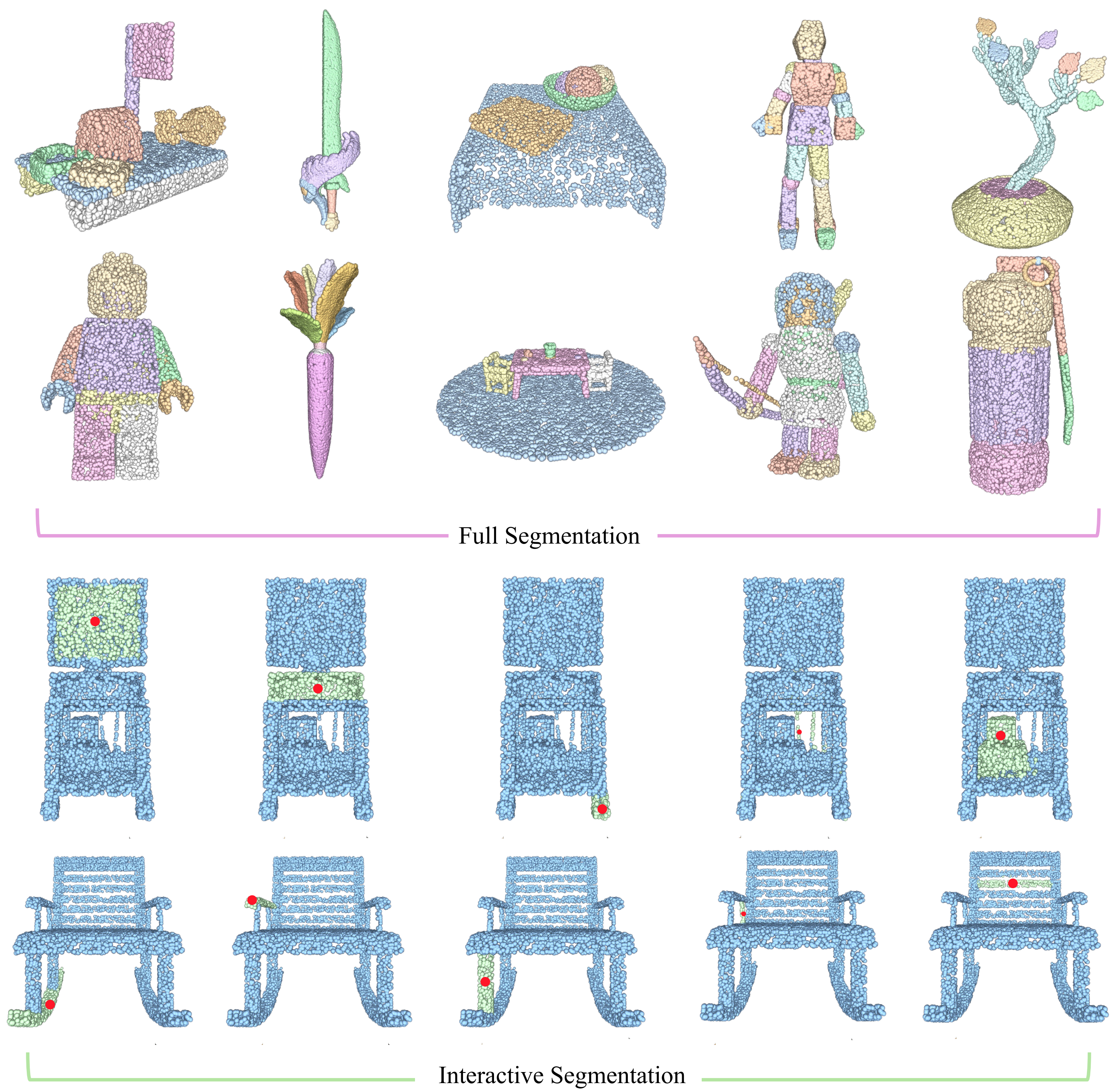

Figure 2. Additional qualitative results of S^{2}AM3D on full segmentation and interactive segmentation on our curated dataset.