PARTITA: MULTI-GRANULARITY PROMPTABLE 3D SEGMENTATION FROM HIERARCHICAL SUPERVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Promptable 3D segmentation involves targets at different granularities, especially when proposing masks from a single click. However, single-granularity annotations limit candidate coverage and target diversity for interactive segmentation. We introduce Partita, a promptable 3D segmentation model that learns multi-granularity prediction and interactive refinement from hierarchical supervision. To provide this supervision at scale, we develop an automatic annotation pipeline that uses a vision-language model (VLM) to organize existing part annotations into nested hierarchies through recursive grouping and correction, yielding PartNeXt-XL, a dataset with 4.20M region masks across about 168K objects. Partita fuses low-resolution prompt-aware features with high-resolution geometric features to resolve finer parts and support efficient interactive refinement. Whereas prior promptable 3D methods supervise a single target mask per prompt, Partita uses multi-granularity matching to jointly supervise predictions across multiple levels. Experiments demonstrate improvements in interactive segmentation and multi-granularity prediction, as well as competitive automatic proposal generation.

1 INTRODUCTION

Promptable segmentation offers a versatile interface for visual understanding by translating prompts into region masks. The Segment Anything Model (SAM) (Kirillov et al., 2023) demonstrated how this interface supports interactive annotation, automatic proposal generation, and other applications. Point-SAM (Zhou et al., 2025) extended promptable segmentation to point clouds, while P^3-SAM (Ma et al., 2025) and PartSAM (Zhu et al., 2026) demonstrated the benefits of scaling 3D part supervision with model designs that can exploit it. These advances make promptable segmentation a promising foundation for both human-guided and automated 3D understanding.

A single click, however, leaves the intended granularity ambiguous: clicking a chair wheel may refer to the wheel or the entire base. Methods such as S^2AM3D (Su et al., 2026) control granularity through a scale input, while SAM and its 3D counterparts output multiple candidate masks. Yet producing high-quality multi-granularity predictions without a scale input remains challenging, limited by both data and supervision. Rich, overlapping annotations in SA-1B support learning different granularities, whereas 3D part annotations typically provide only a single-level partition. Even with multi-granularity annotations, single-target supervision mainly encourages at least one candidate to match a given ground-truth mask, without explicitly promoting joint coverage across granularities; predictions may thus be redundant or low-quality and miss valid targets (Fig. 1). Multi-granularity segmentation also requires small-part and boundary detail, which low-resolution interaction can lose; interaction at the original point resolution instead increases the memory burden of multi-round interactive training.

To provide multi-granularity part annotations at scale, we propose an automatic pipeline for hierarchical 3D part annotation. Existing hierarchical datasets such as PartNet (Mo et al., 2019) rely on manual annotation and category-specific templates, limiting their scalability and category coverage. Our pipeline uses a vision-language model to organize existing part annotations into part trees with explicit containment relations through recursive grouping and automatic correction. Applying the pipeline to HY3D-Bench-Part (Hunyuan3D et al., 2026) yields PartNeXt-XL, a large-scale hierarchical part dataset with approximately **168K objects and 4.20M region masks**, including 1.10M newly generated coarser masks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

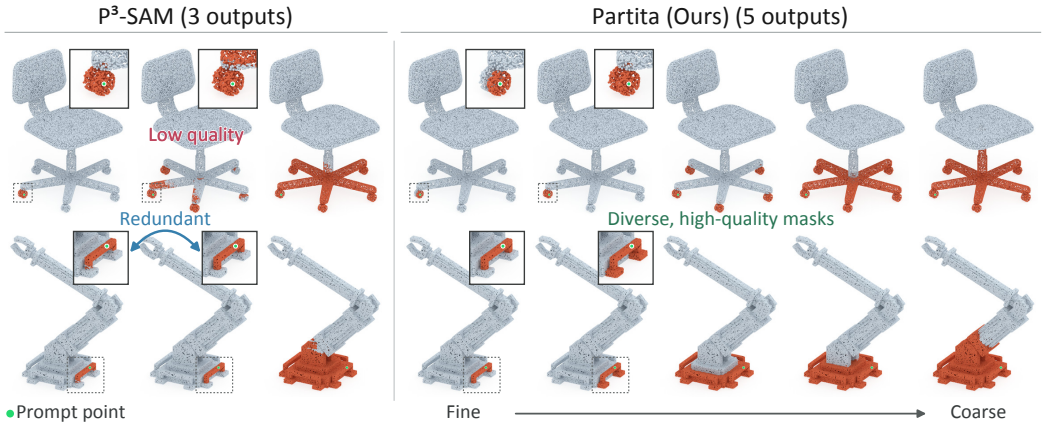

Figure 1: Prior 3D interactive segmentation models, such as P^3-SAM, may produce redundant or low-quality candidates and miss valid targets, limiting their usefulness as proposals. Partita instead predicts a set of five high-quality masks spanning fine to coarse granularities from a single click.

Building on these annotations, we introduce **Partita**, a promptable 3D segmentation model that jointly learns multi-granularity prediction and interactive refinement. During training, we match the first-click predictions to the set of annotated masks containing the click, jointly supervising candidate masks to encourage coverage of different granularities; subsequent interactions refine the mask for the selected target. To balance segmentation detail with the memory demands of multi-round interactive training, the decoder performs prompt interaction only on low-resolution features and fuses independently extracted high-resolution geometric features for mask prediction. The prompt-independent geometric features are reused across clicks. Our decoder achieves comparable overall IoU to full-resolution interaction with approximately 45% less training memory.

We evaluate Partita on multi-granularity segmentation, interactive segmentation, and automatic proposal generation. In addition to PartObjaverse-Tiny (Yang et al., 2024) and PartNetE (Liu et al., 2023), we introduce two multi-granularity evaluation sets: PartNetM, derived from PartNet, and Partita200, our curated set of 200 objects. We use Match mIoU@1 to measure the quality and coverage of first-click predictions across granularities via one-to-one matching. On PartNetM and Partita200, Partita achieves **74.25% and 77.39%** Match mIoU@1, respectively, exceeding the best baseline scores of 64.51% and 62.91%. For interactive segmentation, Partita achieves **91.56%** IoU after seven clicks on PartObjaverse-Tiny, compared with PartSAM's 86.58%. Ablations further validate the benefits of multi-granularity annotations and multi-target supervision for multi-granularity prediction, and demonstrate that our decoder achieves detailed segmentation with lower memory usage. Our contributions are threefold:

- **PartNeXt-XL and a hierarchical annotation pipeline.** We organize existing part annotations into nested hierarchies through VLM-guided recursive grouping and correction, enabling multi-granularity data construction at scale.
- **Partita and a hierarchy-aware training recipe.** We combine low-resolution prompt interaction with a prompt-independent high-resolution geometric branch and jointly supervise candidate masks to cover different granularities.
- **Protocols and datasets for multi-granularity evaluation.** We introduce PartNetM and Partita200, with protocols that assess candidate masks across hierarchy levels and the quality and coverage of automatic proposals.

2 Related Work

3D Interactive Segmentation. Interactive segmentation has been extensively studied in 2D (Kirillov et al., 2023; Ravi et al., 2025; Sofiiuk et al., 2022; Xu et al., 2016). InterObject3D (Kontogianni et al., 2023) and AGILE3D (Yue et al., 2024) demonstrated click-based point cloud seg-

mentation, while Point-SAM (Zhou et al., 2025) first introduced the SAM (Kirillov et al., 2023) paradigm to interactive 3D segmentation. Subsequent methods PartSAM (Zhu et al., 2026) and P^3-SAM (Ma et al., 2025) use large-scale native 3D part data to improve object part segmentation, while SNAP (Gupta et al., 2026) targets cross-domain interactive scene segmentation. These methods make different trade-offs in interaction resolution: low-resolution interaction (Point-SAM and PartSAM) may lose boundary detail, whereas high-resolution interaction (P^3-SAM and SNAP) faces the memory demands of dense point clouds. Although 2D methods such as HQ-SAM (Ke et al., 2023) and PointRend (Kirillov et al., 2020) seek to recover detail with limited additional overhead, balancing small-part and boundary quality, low interaction cost, and multi-round refinement for dense 3D point clouds remains an open challenge.

Multi-Granularity Modeling in 3D Segmentation. A single prompt can correspond to valid segmentations at different granularities. SAM (Kirillov et al., 2023) addresses this ambiguity with multiple candidate masks. The dense, overlapping annotations in SA-1B provide a basis for its prediction heads to learn different granularities. In contrast, the lack of large-scale multi-granularity annotations in 3D limits the ability of interactive segmentation methods such as PartSAM and P^3-SAM to generate high-quality multi-granularity candidate masks. Early work such as PartNet (Yu et al., 2019) and Learning to Group (Luo et al., 2020) explored top-down recursive part decomposition and bottom-up part grouping, respectively. GARField (Kim et al., 2024), SAGA (Cen et al., 2025), and SAMPart3D (Yang et al., 2024) transfer rich 2D segmentation priors into 3D features and control segmentation granularity through scale conditioning. OmniSeg3D (Ying et al., 2024), Ultrametric Feature Fields (He et al., 2024), and PartField (Liu et al., 2025) obtain segmentations at different levels through hierarchical feature learning and clustering. S^{2}AM3D (Su et al., 2026) combines a click with a scale parameter for part segmentation at a user-specified granularity. However, directly obtaining high-quality multi-granularity mask sets from a single click without manually specifying a scale remains an important research challenge.

3D Part Data. Datasets such as ShapeNetPart (Yi et al., 2016), PartVerse (Dong et al., 2025), and PartVerse-XL (Ding et al., 2026) provide ready-made part annotations and structural information. Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2023a) provide abundant assets for constructing 3D part data at scale. Cap3D (Luo et al., 2023) generates object-level text descriptions for these assets to facilitate data filtering. Subsequent work further mines part information from these assets: P^3-SAM (Ma et al., 2025) generates annotations through mesh connectivity analysis, while PartCrafter (Lin et al., 2026) uses part-level metadata embedded in mesh assets to construct training data. Recently, HY3D-Bench-Part (Hunyuan3D et al., 2026) has released part annotations for approximately 240k objects. However, these datasets primarily provide single-level part partitions and lack cross-granularity containment relations, making it difficult to provide hierarchically consistent multi-granularity supervision. PartNet (Mo et al., 2019) and PartNeXt (Wang et al., 2026) provide explicit hierarchical part annotations, but their limited scale and category coverage, together with their reliance on manual annotation and predefined category hierarchies, constrain the construction of large-scale multi-granularity data for open categories.

3 Nested Multi-Granularity 3D Part Data

We construct PartNeXt-XL, a 3D part dataset with nested hierarchies that capture part composition at multiple granularities. Starting from the original part annotations of 3D assets, our pipeline uses recursive grouping and correction guided by a vision-language model to organize parts into a tree with explicit containment relations (Figure 2).

3.1 Automatic Data Annotation Pipeline

Visual prompting with SoM. We use GPT-5.6 Sol (OpenAI, 2026) as our vision-language model (VLM) and adopt Set-of-Mark (SoM) prompting (Yang et al., 2023) to label parts with numeric identifiers in renderings from eight viewpoints and an exploded view. Each round also receives the spatial adjacency relationships among all current masks, allowing the VLM to group parts by appearance, function, and geometry and express each group as a set of identifiers. Each round starts from the original part masks within the current region; when there are too many, we prioritize merging the smallest mask into a spatially adjacent mask to simplify the input.

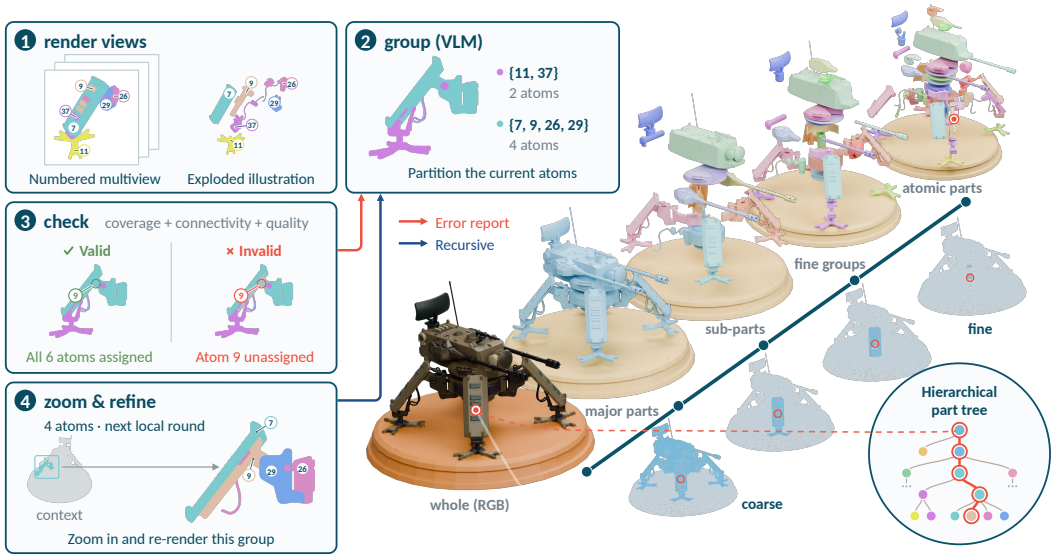

Figure 2: Overview of our automatic annotation pipeline. Left: a VLM recursively groups the original parts using numbered multi-view renderings, an exploded view, and spatial adjacency. Groupings are checked and corrected throughout recursion. Right: the resulting part tree and nested masks at different granularities.

Recursive grouping and correction. The VLM first groups the object's original parts into major functional regions. The pipeline then recurses into each region, using close-up views and its location within the object to inspect the parent's grouping. If a child round identifies an inappropriate mask merge, an incorrect annotation, or spatially disconnected parts, it returns specific revision suggestions to the parent round, which regroups the parts. Once the grouping passes inspection, the child round partitions the region into finer subparts and continues recursively until no meaningful further subdivision remains. Inspection and correction thus proceed throughout recursion, completing the grouping checks as the hierarchy is built.

Hierarchical annotations. The resulting hierarchy has original parts as leaves, with each internal node representing the union of its constituent original parts. Each partition is required to consist of non-overlapping child regions that together cover the parent, establishing consistent containment across granularities.

3.2 Dataset Statistics

We process HY3D-Bench-Part (Hunyuan3D et al., 2026) objects with original part annotations using our automatic annotation pipeline, obtaining multi-granularity annotations for approximately 168k objects. The dataset contains about 4.20M masks, including 1.10M coarser masks generated by our pipeline in addition to the original annotations. The mean hierarchy depth is 2.96, with 99.6% of objects having 2–5 levels. Further implementation and dataset details are provided in Appendix A.

4 Method

Given a point cloud $\mathbf{P} \in \mathbb{R}^{N\times 9}$ (coordinates, colors, and normals) and positive and negative point prompts $\mathbf{U} \in \mathbb{R}^{M\times 3}$, Partita predicts a set of category-agnostic multi-granularity segmentation masks. The model follows an encoder–decoder architecture: a point cloud encoder extracts features, a prompt-conditioned decoder generates masks, and a high-resolution geometric branch provides local detail (§4.1). During training, multi-granularity supervision guides multiple outputs to jointly learn part segmentation at different granularities (§4.2). Figure 3 provides an overview of our framework.

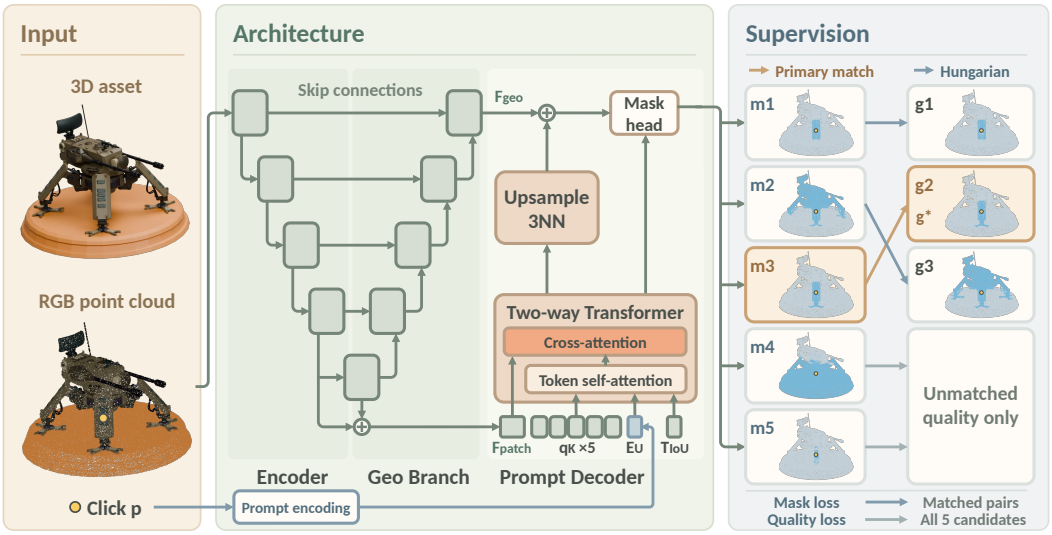

Figure 3: Overview of Partita. (a) Architecture: High-resolution mask prediction through low-resolution prompt interaction and geometric feature fusion. (b) Supervision: Same-click multi-granularity supervision with primary-target assignment followed by Hungarian matching.

4.1 Architecture

Encoder and patch construction. We use a Point Transformer v3 encoder (Wu et al., 2024), initialized with P^3-SAM weights (Ma et al., 2025), to extract multiscale features $\mathcal{F} = \{F_0, \ldots, F_L\}$. Each non-empty voxel at the penultimate stage defines a patch. We align the deepest-stage features to this stage through the pooling indices, concatenate them with the corresponding features, and project the result:

$$F_{\text{patch}} = \phi_p\big([F_{L-1} \| \, \text{Up}_{L \to L-1}(F_L)]\big)\,, \tag{1}$$

where ϕ_p is an MLP and $\|$ denotes feature concatenation. Patch counts vary with the spatial distribution of the input point cloud. During training, we randomly sample patch features when their number exceeds a threshold; at inference, we retain all patches. This sampling limits decoder memory usage during training and can also be viewed as a form of patch-level dropout (Srivastava et al., 2014).

Prompt-conditioned decoder. We combine each point prompt's positional encoding, nearest patch feature, and positive/negative state embedding into a prompt token, denoting their collection by $\mathbf{E}_U$. Guided by the hierarchy-depth distribution of the multi-granularity training set, we use $K = 5$ learnable *granularity queries*, $\mathbf{Q} = (q_1, \ldots, q_K)$, for initial candidate prediction (Table 6). These queries and prompt tokens enter a two-way Transformer (Kirillov et al., 2023). Each decoder block first exchanges information among queries and between queries and prompts through self-attention, then jointly updates the queries and patch features through bidirectional cross-attention:

$$(\mathbf{Q}', F'_{\text{patch}}) = \text{TwoWayDecoder}(\mathbf{Q}, \mathbf{E}_U, F_{\text{patch}}). \tag{2}$$

Through multi-granularity supervision, the queries learn to predict candidate masks at different granularities for the same prompts; subsequent interactions refine the selected mask.

High-resolution geometric branch and mask prediction. To recover small-part and boundary details that the low-resolution features F'_{patch} struggle to represent, we introduce a prompt-independent high-resolution geometric branch. Prompt interaction operates on low-resolution patches, and the two branches fuse only at prediction. This design aims to approach the segmentation quality of interaction at the original point resolution with a lower memory footprint, supporting multi-round interactive training (Table 4). The geometric branch progressively upsamples along the encoder's pooling mappings and fuses corresponding encoder features through skip connections,

yielding the highest-resolution geometric features F_{geo} aligned with the input points. We restore F'_{patch} to the original point resolution by three-nearest-neighbor interpolation and fuse it with the geometric features:

$$F_{\text{fused}} = \phi_f\big([F_{\text{geo}} \| \operatorname{Interp}_{3\text{NN}}(F'_{\text{patch}})]\big), \tag{3}$$

where ϕ_f is an MLP. Projected granularity queries produce candidate masks via pointwise dot products with the fused features, and a quality prediction head outputs their quality scores. Encoder and geometric features are reused across interaction rounds.

4.2 Multi-granularity Training

Same-click multi-granularity targets. Multi-mask methods using single-target, minimum-loss supervision (Kirillov et al., 2023) supervise only the lowest-loss candidate against one ground-truth mask, without an explicit constraint to jointly cover different granularities. We construct same-click multi-granularity target sets to jointly supervise the granularity queries. During training, we uniformly sample a primary target $g^\star$ from the annotation pool across all granularity levels, sample a positive point p uniformly within it, and collect all annotated masks containing p. For hierarchical annotations, these targets form a strictly nested chain from fine to coarse:

$$\mathcal{G}_p = \{g_1 \subsetneq g_2 \subsetneq \cdots \subsetneq g_{L_p}\}. \tag{4}$$

Target assignment and supervision losses. The first round predicts K candidate masks $\{\hat{m}_k\}_{k=1}^K$. We first assign the candidate with the lowest mask loss to $g^\star$, remove $g^\star$ from $\mathcal{G}_p$, and perform one-to-one Hungarian matching between the remaining candidates and targets using negative soft Dice similarity as the cost. Let $\mathcal{A}_p$ denote all matched pairs. The first-round loss for a single prompt is

$$\mathcal{L}_p^{(1)} = \frac{1}{|\mathcal{A}_p|} \sum_{(k,g)\in\mathcal{A}_p} \ell_{\text{mask}}(\hat{m}_k, g) + \frac{1}{K}\sum_{k=1}^{K}(s_k - y_k)^2, \tag{5}$$

where $\ell_{\text{mask}} = \ell_{\text{focal}} + \alpha\,\ell_{\text{Dice}}$ combines focal loss (Lin et al., 2017) and Dice loss (Milletari et al., 2016), and s_k is the predicted quality score. For matched candidates, y_k is the actual IoU with the assigned target; unmatched candidates use the maximum IoU over the complete same-click target set. Mask losses are averaged over actual matched pairs within each prompt and then over prompts, preventing longer target chains from receiving greater weight through loss accumulation.

Interactive training. Our multi-granularity training set comprises approximately 190k objects from PartNeXt-XL and PartNeXt (Wang et al., 2026). We simulate $T = 7$ interaction rounds. The first round uses multi-granularity supervision. Later rounds combine mask prompts with positive and negative corrective clicks sampled from prediction error regions to refine a single mask, supervised against the original primary target $g^\star$. Each round uses the same mask loss and quality regression, and losses are averaged across rounds.

5 Experiments

5.1 Setup and Protocol

We evaluate interactive segmentation and refinement, multi-granularity prediction, and automatic proposal generation, assessing the model's ability to respond to user prompts and progressively refine masks, predict targets at different granularities from the same prompt, and automatically generate and rank valid segmentation proposals.

We compare Partita with strong baselines: Point-SAM (Zhou et al., 2025), P^3-SAM (Ma et al., 2025), PartSAM (Zhu et al., 2026), and S^{2}AM3D (Su et al., 2026). The first three produce multiple candidate masks from the initial click. Point-SAM and PartSAM support iterative click-based refinement, whereas P^3-SAM supports only single-point prompts. S^{2}AM3D produces a single mask from a click and a scale parameter. We use the official implementations and released weights for all baselines and adopt a 100,000-point evaluation setting. Baseline-specific adaptations and settings are provided in Appendix C.3.1.

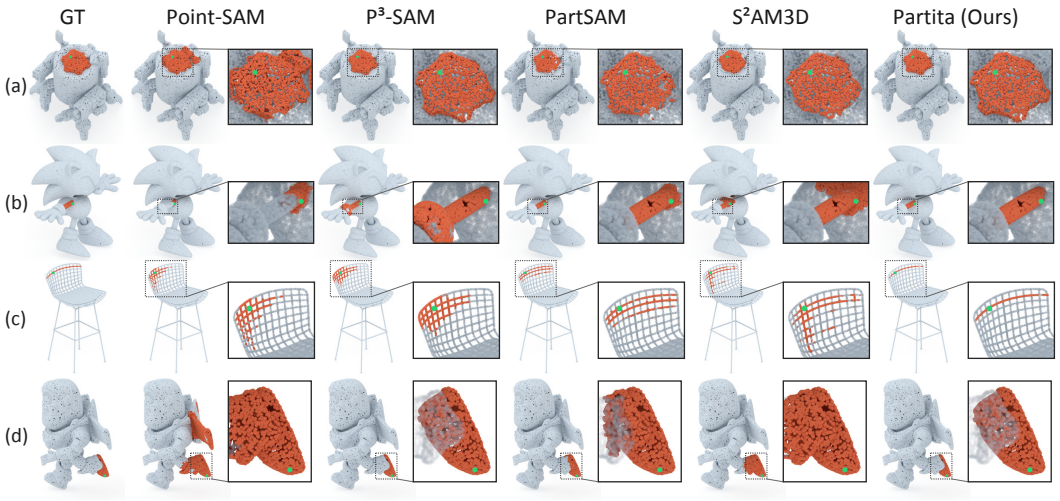

Figure 4: Single-click segmentation comparison across methods. Partita achieves better segmentation quality on small parts and along part boundaries. In some cases, its results are comparable to those of P^3-SAM, which uses full-resolution interaction. For each object, all methods use the same prompt point (green). **Zoom in for a better view.**

Table 1: Interactive part segmentation (IoU@k, %; higher is better), where k is the number of clicks. Best results within each dataset are in bold; dashes denote unsupported click counts. For S^{2}AM3D, we use scales $\{0.01, 0.03, 0.09, 0.27, 0.81\}$ to generate five candidate masks for the same click.

Method	PartNetE				PartObjaverse-Tiny			
	IoU@1	IoU@3	IoU@5	IoU@7	IoU@1	IoU@3	IoU@5	IoU@7
Point-SAM	60.14	79.79	84.83	86.47	46.54	63.01	70.77	73.81
P^3-SAM	68.31	—	—	—	65.50	—	—	—
PartSAM	72.06	86.93	91.40	93.03	60.91	78.30	84.56	86.58
S^{2}AM3D	63.75	—	—	—	52.02	—	—	—
Ours	**80.67**	**88.03**	**91.76**	**93.05**	**74.35**	**83.72**	**89.81**	**91.56**

5.2 Interactive Segmentation and Refinement

We evaluate interactive part segmentation on PartNetE (Liu et al., 2023) and PartObjaverse-Tiny (Yang et al., 2024) using the intersection over union (IoU) between predicted and ground-truth (GT) masks. To assess interactive refinement, we report the IoU after k clicks, denoted as IoU@k. The first click is randomly sampled within the GT mask, and subsequent clicks are placed in the deepest prediction error regions. Since the same click can correspond to valid targets at different granularities, we select the first-round candidate with the highest GT IoU rather than the highest predicted quality score, reducing the influence of candidate ranking and focusing on segmentation and interactive refinement.

The quantitative results in Table 1 show that Partita achieves the best segmentation and refinement results at each click count, with particularly strong advantages when few clicks are available, demonstrating higher interaction efficiency. The qualitative results in Fig. 4 further demonstrate Partita's accurate delineation of small parts and mask boundaries. Compared with Point-SAM and PartSAM, our predictions preserve more complete fine structures and clearer part boundaries, matching the detail quality of P^3-SAM's native high-resolution predictions in the illustrated regions. This is enabled by the high-resolution geometric branch, which supplies local details that interpolation struggles to recover.

5.3 Multi-Granularity Prediction and Proposal Generation

Single-granularity part segmentation benchmarks evaluate only one of a model's candidate masks, leaving the multi-granularity quality of the full prediction set unmeasured. We therefore construct

Table 2: Multi-granularity prediction and automatic proposal generation (%; higher is better). Best results within each dataset are in bold. S^{2}AM3D uses five scales as in Table 1; its AP/AR are unreported because it provides no native mask-quality scores.

Method	PartNetM			Partita200		
	Match mIoU@1	AP	AR@300	Match mIoU@1	AP	AR@300
Point-SAM	52.20	15.63	31.89	41.72	8.84	23.94
P^3-SAM	60.07	19.25	35.55	62.91	38.40	52.37
PartSAM	64.51	**30.92**	45.84	51.48	22.62	36.57
S^{2}AM3D	62.12	—	—	54.19	—	—
Ours	**74.25**	28.17	**45.96**	**77.39**	**40.73**	**55.99**

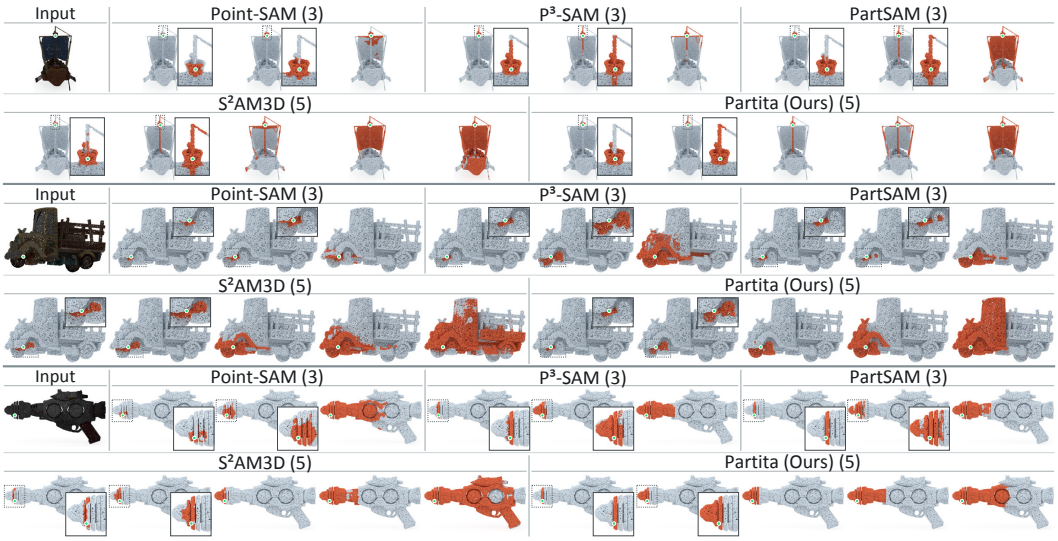

Figure 5: Single-click multi-granularity segmentation comparison. For each object, all candidate masks are shown using the same prompt point (green). Partita predicts diverse, high-quality masks from fine parts to larger components. **Zoom in for a better view.**

two new multi-granularity test sets. **PartNetM** comprises 2,000 objects with multi-granularity annotations selected from 20 categories in PartNet (Mo et al., 2019). Since PartNet primarily covers household objects, we additionally construct **Partita200**, comprising 200 objects across a broader range of object categories. Dataset details are provided in Appendix C.1. We use Match mIoU@1 to measure the quality of the multi-granularity mask set from a single click: Hungarian matching assigns predictions one-to-one to all GT masks containing the click, and we compute the average IoU. For automatic proposals, we use COCO-style AP (Lin et al., 2014) to assess proposal quality and ranking, and AR@300 to measure target coverage. Results are shown in Table 2. Notably, PartSAM uses PartNet for training, whereas we do not. Additional results are provided in Appendix C.3.3.

Fig. 5 shows Partita's single-click multi-granularity predictions: the same click yields accurate masks from fine-grained parts to larger components. In the illustrated examples, baseline methods produce duplicate or lower-quality candidates that do not fully cover different target granularities. Higher Match mIoU@1 (Table 2) further supports this advantage.

5.4 Ablation Study

Multi-granularity annotations. We construct three training sets from the same pool of point clouds, selecting each smaller object set from the larger one. Under a fixed budget of 1M masks, we set the proportion of data with multi-granularity annotations, measured by object count, to 0%, 50%, and 100%. Table 3(a) shows that the 100% setting improves Match mIoU@1 and AP by 14.29 and 8.16 points, respectively, compared with the 0% setting, despite using fewer training objects. This demonstrates that our annotations more effectively support multi-granularity mask-set predic-

Table 3: Multi-granularity annotation ablations on Partita200: (a) varying the annotation ratio under a fixed mask budget; (b) with or without multi-granularity annotations on a fixed object set.

(a) Fixed mask budget

Multi-gran. (%)	Objects (k)	Match	AP	AR@300
0	54.65	62.45	31.22	52.74
50	46.47	73.22	35.82	53.56
100	40.47	**76.74**	**39.38**	**54.05**

(b) Fixed object set

Multi-gran. annotations	Masks (M)	Match	AP	AR@300
×	0.74	60.90	32.33	**54.11**
✓	1.00	**76.74**	**39.38**	54.05

Table 4: Segmentation quality and training memory on PartObjaverse-Tiny using 100k-update checkpoints. Small denotes masks containing less than 5% of the input points.

Decoder design	IoU@1 (%) ↑		Boundary IoU@1 (%) ↑		Memory ↓ (GiB)
	Small	Overall	Small	Overall	
No geometric branch	65.15	68.17	47.84	42.78	**8.83**
Full-resolution interaction	72.62 (+7.47)	**74.23** (+6.06)	**61.60** (+13.76)	**56.82** (+14.03)	20.25
Ours (100k)	**73.29** (+8.14)	73.46 (+5.28)	61.30 (+13.46)	54.15 (+11.36)	11.18 **(−44.8%)**

tion and proposal generation under a fixed annotation budget. We further compare training with and without multi-granularity annotations on the same 40,470 objects (Table 3(b)) to assess the benefit of adding multi-granularity annotations to a fixed object set.

Multi-target supervision. On the same multi-granularity training data, we compare two matching strategies: selecting one predicted mask to match a single GT, and matching the predicted mask set to the multi-granularity GT set for the same click (ours). Table 11 in Appendix C.3.4 shows that set matching improves Match mIoU@1 and AP by 7.77 and 6.10 points, respectively, while maintaining comparable AR@300. This indicates that, even with identical multi-granularity training data, explicit set matching helps the model make better use of the annotations by guiding its candidate outputs toward different granularities.

High-resolution geometric branch. We compare the segmentation quality of three decoder designs on PartObjaverse-Tiny: low-resolution prompt interaction without geometric features, direct interaction at the original point resolution, and our low-resolution interaction with high-resolution geometric fusion. Table 4 shows that our design substantially improves small-part IoU and Boundary IoU (Cheng et al., 2021) over the version without geometric features. It achieves overall IoU comparable to full-resolution interaction while using approximately 45% less training memory, combining fine spatial detail with efficient prompt interaction.

6 Conclusion

We introduced Partita for single-click multi-granularity 3D segmentation and interactive refinement. Our VLM-guided pipeline constructs PartNeXt-XL by organizing existing part annotations into nested hierarchies through recursive grouping and correction, without category-specific templates. Same-click multi-target matching encourages candidates to cover different granularities. The decoder combines low-resolution prompt interaction with a prompt-independent high-resolution geometric branch, preserving fine details with less training memory than full-resolution interaction. Partita outperforms prior methods in multi-granularity and interactive segmentation and remains competitive in automatic proposal generation. These results suggest that granularity can be learned as an output from scalable hierarchical supervision without an explicit scale input.

AI USE STATEMENT

We used generative AI in two ways: as a component of our annotation pipeline and as an assistant for research and writing. In our method, the annotation pipeline uses GPT-5.6 Sol (OpenAI, 2026) as its VLM to recursively group existing part annotations into part trees (§3). This pipeline produces the hierarchical annotations of PartNeXt-XL and the initial annotations of Partita200. Appendix A lists all prompts and settings used with this model. In addition, we used coding and writing tools based on large language models (OpenAI Codex and Claude Code) to translate text, to implement data processing, training, and evaluation code, and to build and organize datasets. We did not use generative AI to formulate research questions, design methods or experiments, or interpret results; the paper contains no mathematical proofs, and the remaining tasks that require disclosure do not apply to this work. We also used these tools to draft and polish parts of the text and to write rendering scripts for figures. We have reviewed all AI-assisted work: every number reported in the paper was checked against the experimental result files, and every reference was checked against its original source. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no human subjects and collects no personal data; the main considerations are the provenance and licensing of the released data. PartNeXt-XL is built on HY3D-Bench-Part (Hunyuan3D et al., 2026), whose assets are curated from Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2023a). Both datasets are licensed as a whole under ODC-By 1.0, and each object carries its own license; all Objaverse objects are released under Creative Commons licenses. The release of PartNeXt-XL will follow the format of HY3D-Bench and strictly adhere to its license terms. The manual review of Partita200 was carried out by the authors.

REPRODUCIBILITY STATEMENT

Since the contributions of this paper lie mainly in data and supervision, we provide complete details of data construction, training, and evaluation. For data, §3 describes the design of the annotation pipeline, and Appendix A gives the complete per-object algorithm (Algorithm 1), all pipeline settings (Table 5), the full prompts of the root and child rounds (Figure 6), and the validation checks that every round's output and the final hierarchies must pass. Because the pipeline relies on a proprietary VLM, rerunning it may not reproduce exactly the same groupings; we will therefore release the PartNeXt-XL dataset generated by the pipeline, so that training and evaluation on it require no further VLM queries. For training, Appendix B specifies the model architecture, the loss and matching rules, the composition of the training data and its deduplication against the evaluation sets, the optimization and scheduling settings, and click sampling for interactive training. For evaluation, §5.1 and Appendices C.1–C.3 describe the evaluation sets; the point clouds, GT masks, and click indices shared by all methods; the metric definitions; and the baseline settings. Section 5.4 specifies the controlled settings of the ablations, including the fixed mask budget and nested object sets. The code and model weights will also be publicly released.

REFERENCES

Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 1971–1979, 2025.

Bowen Cheng, Ross Girshick, Piotr Dollár, Alexander C Berg, and Alexander Kirillov. Boundary iou: Improving object-centric image segmentation evaluation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15329–15337. IEEE, 2021.

Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023a.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of anno-

tated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13142–13153, 2023b.

Lihe Ding, Shaocong Dong, Yaokun Li, Chenjian Gao, Xiao Chen, Rui Han, Yihao Kuang, Hong Zhang, Bo Huang, Zhanpeng Huang, et al. Fullpart: Generating each 3d part at full resolution. In *International Conference on Learning Representations*, volume 2026, pp. 45856–45876, 2026.

Shaocong Dong, Lihe Ding, Xiao Chen, Yaokun Li, Yuxin Wang, Yucheng Wang, Qi Wang, Jaehyeok Kim, Chenjian Gao, Zhanpeng Huang, et al. From one to more: Contextual part latents for 3d generation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8230–8240. IEEE, 2025.

Aniket Gupta, Hanhui Wang, Charles Saunders, Aruni RoyChowdhury, Hanumant Singh, and Huaizu Jiang. Snap: Towards segmenting anything in any point cloud. In *2026 International Conference on 3D Vision (3DV)*, pp. 1660–1669. IEEE, 2026.

Haodi He, Colton Stearns, Adam W Harley, and Leonidas J Guibas. View-consistent hierarchical 3d segmentation using ultrametric feature fields. In *European Conference on Computer Vision*, pp. 268–286. Springer, 2024.

Team Hunyuan3D, Bowen Zhang, Chunchao Guo, Dongyuan Guo, Haolin Liu, Hongyu Yan, Huiwen Shi, Jiaao Yu, Jiachen Xu, Jingwei Huang, et al. Hy3d-bench: Generation of 3d assets. *arXiv preprint arXiv:2602.03907*, 2026.

Lei Ke, Mingqiao Ye, Martin Danelljan, Yu-Wing Tai, Chi-Keung Tang, Fisher Yu, et al. Segment anything in high quality. *Advances in Neural Information Processing Systems*, 36:29914–29934, 2023.

Chung Min Kim, Mingxuan Wu, Justin Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with radiance fields. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21530–21539. IEEE, 2024.

Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 9796–9805. IEEE, 2020.

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 *2023 IEEE/CVF international conference on computer vision (ICCV)*, pp. 3992–4003. IEEE, 2023.

Theodora Kontogianni, Ekin Celikkan, Siyu Tang, and Konrad Schindler. Interactive object segmentation in 3d point clouds. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2891–2897. IEEE, 2023.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pp. 2980–2988, 2017.

Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Feng Yiqiang, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *Advances in Neural Information Processing Systems*, 38:35387–35415, 2026.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21736–21746. IEEE, 2023.

Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9704–9715. IEEE, 2025.

Tiange Luo, Kaichun Mo, Zhiao Huang, Jiarui Xu, Siyu Hu, Liwei Wang, and Hao Su. Learning to group: A bottom-up framework for 3d part discovery in unseen categories. *arXiv preprint arXiv:2002.06478*, 2020.

Tiange Luo, Chris Rockwell, Honglak Lee, and Justin Johnson. Scalable 3d captioning with pretrained models. *Advances in Neural Information Processing Systems*, 36:75307–75337, 2023.

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.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, pp. 565–571. Ieee, 2016.

Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 909–918. IEEE, 2019.

OpenAI. GPT-5.6 Sol model documentation, 2026. URL `https://developers.openai.com/api/docs/models/gpt-5.6-sol`.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. In *International Conference on Learning Representations*, volume 2025, pp. 28085–28128, 2025.

Konstantin Sofiiuk, Ilya A Petrov, and Anton Konushin. Reviving iterative training with mask guidance for interactive segmentation. In *2022 IEEE international conference on image processing (ICIP)*, pp. 3141–3145. IEEE, 2022.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Han Su, Tianyu Huang, Zichen Wan, Xiaohe Wu, and Wangmeng Zuo. S^2am3d: Scale-controllable part segmentation of 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14357–14366, 2026.

Penghao Wang, Yiyang He, Xin Lv, Yukai Zhou, Lan Xu, Jingyi Yu, and Jiayuan Gu. Partnext: A next-generation dataset for fine-grained and hierarchical 3d part understanding. *Advances in Neural Information Processing Systems*, 38, 2026.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4840–4851. IEEE, 2024.

Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-supervised learning of reliable point representations. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22193–22204. IEEE, 2025.

Ning Xu, Brian Price, Scott Cohen, Jimei Yang, and Thomas S Huang. Deep interactive object selection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 373–381, 2016.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

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.

Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.

Haiyang Ying, Yixuan Yin, Jinzhi Zhang, Fan Wang, Tao Yu, Ruqi Huang, and Lu Fang. Omniseg3d: Omniversal 3d segmentation via hierarchical contrastive learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 20612–20622. IEEE, 2024.

Fenggen Yu, Kun Liu, Yan Zhang, Chenyang Zhu, and Kai Xu. Partnet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9483–9492. IEEE, 2019.

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*, volume 2024, pp. 11391–11411, 2024.

Yuchen Zhou, Jiayuan Gu, Tung Chiang, Fanbo Xiang, and Hao Su. Point-sam: Promptable 3d segmentation model for point clouds. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 56850–56869, 2025.

Zhe Zhu, Le Wan, Rui Xu, Yiheng Zhang, Honghua Chen, Zhiyang Dou, Cheng Lin, Yuan Liu, and Mingqiang Wei. Partsam: A scalable promptable part segmentation model trained on native 3d data. In *International Conference on Learning Representations*, volume 2026, pp. 132469–132493, 2026.

A PartNeXt-XL Dataset

A.1 Annotation Pipeline and Verification

Data sources. PartNeXt-XL is built on HY3D-Bench-Part (Hunyuan3D et al., 2026), the part-level subset of HY3D-Bench, which contains 240,524 assets curated from Objaverse (Deitke et al., 2023b) and Objaverse-XL (Deitke et al., 2023a). The original part annotations of this subset are obtained automatically: each asset is split into connected components, and components with very small surface areas are merged into adjacent larger components. We use these components as the original parts of our hierarchies, and PartNeXt-XL contains 167,782 of these assets. The annotation data for PartNeXt-XL will be publicly released.

Pipeline pseudocode. Algorithm 1 summarizes recursive grouping and correction for one object. Each round merges masks until at most N remain, queries the VLM, and validates the response. A child round either returns revision suggestions, which send its parent region back for regrouping, or partitions its own region into subparts.

Release-level validation. We validate all 167,782 hierarchies of PartNeXt-XL at two levels. At the annotation level, every hierarchy passes structural checks: every partition covers each original part of its parent exactly once, every child region is contained in its parent, sibling regions are disjoint, every region at each level is spatially connected without floating components, and every leaf is a single original part. At the point-cloud level, we materialize the hierarchies on both mesh versions described in Appendix B.3 and remove empty, whole-object, and duplicate masks after projection. All non-watertight versions retain every original part. In about 10% of the watertight versions, some original parts receive no sampled points, and the affected assets consist of the remaining visible parts.

A.2 VLM Prompts and Implementation Settings

Table 5 lists the settings of the annotation pipeline, and Figure 6 shows the prompts of the root and child rounds. Each prompt is followed by a JSON payload that lists the mask identifiers, the area fraction of each mask, and the spatial adjacency between masks. When a response fails validation, the error messages are appended to the prompt and the VLM is queried again.

Algorithm 1 Recursive grouping and correction for one object.

Require: original part masks $\mathcal{A}$, spatial adjacency $\mathcal{E}$ over $\mathcal{A}$, mask limit N, leaf size M, depth limit D, attempt limit T
Ensure: part tree $\mathcal{T}$ with original parts as leaves

function ROUND(R, f) ▷ region R, feedback f
  $\mathcal{U} \leftarrow$ original part masks within R
  while $|\mathcal{U}| > N$ **do**
    merge the smallest mask in $\mathcal{U}$ into a spatially adjacent mask
  end while
  $\mathcal{I} \leftarrow$ SoM renderings of $\mathcal{U}$ ▷ multi-view and exploded views; locator for child rounds
  for $t = 1, \ldots, T$ **do**
    $y \leftarrow \mathrm{VLM}(\mathcal{I}, \mathcal{E}|_{\mathcal{U}}, f)$
    $e \leftarrow$ VALIDATE$(y, \mathcal{U}, \mathcal{E})$ ▷ coverage, quality, connectivity
    if $e = \varnothing$ **then**
      return y
    end if
    $f \leftarrow f \cup e$ ▷ re-query with the errors
  end for
  return FAIL ▷ the object is later rerun from this round
end function
$\mathcal{T} \leftarrow$ tree containing only the whole object O; $Q \leftarrow [(O, \varnothing)]$
while $Q \neq \varnothing$ **do**
  $(R, f) \leftarrow$ POP(Q); $y \leftarrow$ ROUND(R, f)
  if y contains revision suggestions r **then** ▷ child rejects its parent's grouping
    $P \leftarrow$ PARENT(R); remove the subtree below P from $\mathcal{T}$ and its pending rounds from Q
    PUSH$(Q, (P, r))$ ▷ parent regroups with the suggestions
  else if y is flat **then**
    attach the original parts in R to R as leaves
  else
    for each group C in y **do**
      add C to $\mathcal{T}$ as a child of R
      if C contains more than M original parts and DEPTH$(C) \leq D$ **then**
        PUSH$(Q, (C, \varnothing))$
      else
        attach the original parts in C to C as leaves
      end if
    end for
  end if
end while
check that every level of $\mathcal{T}$ covers each original part exactly once
return $\mathcal{T}$

Table 5: Settings of the automatic annotation pipeline.

Setting	Value
VLM	GPT-5.6 Sol
Reasoning effort	xhigh
Maximum completion tokens	32,000
Views per round	8 numbered views and 1 exploded view
Additional input of child rounds	1 locator image of the region
Image resolution	900×900 (locator 640×640)
Rendering	flat shading, one color and one number per mask
Adjacency threshold	0.5% of the bounding-box diagonal
Mask limit per round N	40
Regions in the root round	2–6
Leaf size M / depth limit D	2 original parts / 6
Attempts per round T	3

Common instructions. You are annotating the part hierarchy of a 3D object. The images show the object from eight viewpoints and in an exploded view. Each colored region is one mask marked with a numeric ID (Set-of-Mark). The payload lists the area fraction of every mask and the spatial adjacency between masks; rely on the adjacency list for contacts that are occluded in the images. Rules: (1) Use only the mask IDs in the payload, and assign each ID to exactly one group. (2) Give each group a short lowercase English name, e.g., `head`, `left_arm`, `seat_cushion`. (3) Colors only identify masks; group masks by appearance, function, and geometry. (4) Every group must be spatially connected under the adjacency list. Do not pack spatially separated repeated parts, such as the four legs of a chair, into one group; keep them separate or include the part that connects them. (5) Output only JSON in the required format.

Root round. Group the whole object into 2–6 major functional regions, e.g., `head`, `body`, `base`, `accessories`. Do not split too finely; each region will be refined in later rounds. Prefer groupings whose regions contain further substructure. Output: `{"groups": [{"name": ..., "ids": [...]}, ...]}`

Child round (region `<name>`). The first image locates this region, shown in red, within the whole object; the other images are close-up views and an exploded view of this region only. (1) Check the quality of the grouping that produced this region, i.e., whether the region forms a coherent and meaningful part. If masks were merged inappropriately, a mask is annotated incorrectly, or the region contains spatially disconnected parts, return revision suggestions for the parent round, naming the mask IDs and the proposed change. (2) Otherwise, partition the region into meaningful, nameable subparts. If the masks are parallel repeated parts without a nameable intermediate level, such as a row of identical bolts, set `"flat": true`. Do not invent structure; the finest level always falls back to the original parts. Output: `{"revisions": [{"ids": [...], "issue": ..., "suggestion": ...}], "flat": false, "groups": [...]}`

Figure 6: Prompts of the root and child rounds. The JSON payload of each round is appended to its prompt.

Table 6: Annotation statistics of PartNeXt-XL. Chain lengths are weighted by original parts; percentages are rounded independently.

Hierarchy depth	2	3	4	5		≥ 6
Objects (%)	33.433	41.756	20.746	3.676		0.384
Chain length L_a	1	2	3	4	5	≥ 6
Parts (%)	3.023	50.675	36.346	8.899	0.979	0.079

Statistic per object	Mean	Median
Original parts	18.51	16
Coarser masks	6.53	5
Total masks	25.05	21
Hierarchy depth	2.96	3

A.3 Hierarchy and Region Statistics

Table 6 reports statistics of PartNeXt-XL, computed from the original part annotations of its 167,782 objects. These objects contain 3,106,302 original part masks and 1,095,802 coarser masks, 4,202,104 masks in total. A region is represented by the set of original parts it contains. Identical regions within each object are counted once, and the whole-object region is excluded. Coarser masks are the intermediate and coarse regions generated by our pipeline beyond the original part annotations.

The hierarchy depth is the maximum length of a strict containment chain, counting the original part but excluding the whole object. Equivalently, all splittable branches are refined simultaneously, so independent branch splits do not introduce extra levels. The table also reports per-object summaries and the distribution of chain lengths L_a, where L_a counts the distinct regions containing original part a.

Table 7: Comparison with existing 3D part datasets.

Dataset	Objects	Parts / regions	Annotation organization
PartVerse (Dong et al., 2025)	12K	91K	Single-level parts
PartVerse-XL (Ding et al., 2026)	40K	320K	Single-level parts
PartNet (Mo et al., 2019)	26,671	573,585	Hierarchical
PartNeXt (Wang et al., 2026)	23,519	350,187	Hierarchical
PartNeXt-XL (Ours)	167,782	4,202,104	Hierarchical



Figure 7: PartNeXt-XL objects at multiple granularities. Each row is one object: textured input, hierarchy levels 1–3, and the original parts (leaves).

A.4 Comparison with Existing Part Datasets

Table 7 compares PartNeXt-XL with existing 3D part datasets in scale and annotation organization. PartVerse (Dong et al., 2025) and PartVerse-XL (Ding et al., 2026) provide single-level part annotations, obtained by automatic pre-segmentation followed by manual refinement. PartNet (Mo et al., 2019) and PartNeXt (Wang et al., 2026) provide hierarchical annotations, which are manually labeled following category-specific hierarchy templates and cover 24 and 50 categories, respectively. In contrast, our pipeline builds object-specific hierarchies automatically without predefined category templates. This allows PartNeXt-XL to scale to 167,782 objects, about six to seven times as many as PartNet and PartNeXt and more than four times as many as PartVerse-XL, with 4.20M region masks across all granularities.

A.5 Dataset Visualizations

Figure 7 shows twelve PartNeXt-XL objects across granularities. For each object, the first column is the textured input, the next three columns show the first three levels of the nested hierarchy, and the last column shows the original parts, i.e., the leaves of the tree.

B Model and Training Details

This section provides additional details on the model architecture, supervision, training data, and implementation.

B.1 Architecture Details

Patch sampling and feature projections. The number of patches varies with the spatial distribution of the input points. In practice, the vast majority of training point clouds contain 2,000–4,000 patches. We cap the number of patches at 6,144 per training point cloud, sampling uniformly at random when this limit is exceeded. Validation and testing retain all patches. The concatenated encoder features are normalized with LayerNorm and projected by an MLP ϕ_p with dimensions $896 \to 512 \to 512$.

Prompt encoding and decoder. For a click at u_j, we retrieve its nearest retained patch feature f_j and combine it with a random Fourier positional encoding and a learned state embedding e_{s_j} for a positive or negative click:

$$e_j = \mathrm{MLP}([f_j \| \, \mathrm{PE}(u_j)]) + e_{s_j}, \qquad \mathbf{E}_U = (e_1, \ldots, e_M). \tag{6}$$

Previous-round mask prompts are concatenated with patch-relative point coordinates, encoded by lightweight convolutions, and scatter-max pooled into patches to provide dense conditioning. A learned no-mask embedding is used when no mask prompt is available. The two-way Transformer uses four blocks (512 channels, eight heads, FFN width 2048). Its sparse tokens include five granularity queries, one refinement query, one quality token, and click tokens. It predicts five masks initially and refines one in later rounds.

Geometric features and mask prediction. The geometric decoder uses four unpooling stages with encoder skip connections to recover point-level features. These are fused with prompt-conditioned patch features upsampled by three-nearest-neighbor interpolation. Projected queries predict masks from the fused features, while the quality token predicts their scores. Encoder and geometric features are computed once per point cloud and reused across interaction rounds.

B.2 Supervision Details

Mask loss. We combine point-averaged sigmoid focal loss (Lin et al., 2017) and Dice loss (Milletari et al., 2016) with weights 1 and 2, respectively. The Dice smoothing constant is 10^{-3}.

Assignment and quality targets. For each initial click, we assign the candidate with the lowest mask loss to the primary target and exclude the matched target from the auxiliary pool. The remaining candidates and targets are paired by Hungarian matching, using negative soft-Dice similarity with a numerical stability constant of 10^{-6} as the cost. Only matched pairs receive mask supervision; unmatched candidates receive quality supervision only.

We regress quality scores to mask IoU using mean squared error, binarizing predictions at a probability threshold of 0.5. Matched candidates use the IoU with their assigned targets; unmatched candidates use the maximum IoU over all ground-truth masks containing the click. These IoU targets are detached from gradient computation. Within each prompt, mask loss is averaged over the actual matched pairs and quality loss over all candidates. We then average over prompts and interaction rounds.

B.3 Training Data

We train with multi-granularity annotations from PartNeXt-XL and PartNeXt (Wang et al., 2026), about 190K objects in total. Objects that overlap with PartNetE, PartObjaverse-Tiny, PartNetM, or Partita200 are removed by matching original asset identifiers.

Building on the original part annotations of HY3D-Bench-Part (Hunyuan3D et al., 2026), PartNeXt-XL retains both watertight and non-watertight mesh versions. We include the watertight version to support segmentation of generated 3D assets and scan reconstructions. For objects with both

Table 8: Category distribution of PartNetM.

Category	Objects	Category	Objects
Bag	2	Knife	65
Bed	27	Lamp	243
Chair	243	Laptop	108
Clock	38	Microwave	45
Dishwasher	70	Refrigerator	52
Display	183	Scissors	30
Door	47	Storage furniture	243
Earphone	33	Table	243
Faucet	187	Trash can	55
Keyboard	11	Vase	75

versions, we sample one per epoch, choosing the watertight version with probability 66% and the non-watertight version with probability 34%.

B.4 Training Implementation

Training setup. We train Partita for 200,000 steps on 16 NVIDIA H20 GPUs using BF16 mixed precision, with a batch size of 2 per GPU and 32 globally. Each point cloud provides 100,000 sampled points and four primary-mask tasks. Before augmentation, point clouds are centered and normalized to a unit sphere; PTv3 uses a voxel size of 0.005. We use the AdamW optimizer with a weight decay of 0.1. The learning rate is linearly warmed up to 5×10^{-5} over 1,500 steps, then decayed by a factor of 0.6 every 50k steps. Training takes approximately 8 days.

Data augmentation. We apply random scaling in $[0.8, 1.0]$, independent x/z flips with probability 0.5 each, and random y-axis rotation in $[-\pi, \pi]$. Additional Gaussian rotation perturbations about all axes use a standard deviation of 0.06 rad and are clipped to ± 0.18 rad. We also apply color auto-contrast, translation, and Gaussian jitter. Color and normal features are independently dropped with probability 0.2 per cloud. During training, we perturb click coordinates with Gaussian noise of standard deviation 0.001.

Interaction rollout. The first positive click is sampled uniformly inside the primary mask. Subsequent corrective clicks are sampled from prediction errors: false-negative regions provide positive clicks, and false-positive regions provide negative clicks. Sampling probabilities are obtained by applying a softmax with temperature 0.25 to normalized distances from points to the corresponding error-region boundary. The logits of the first-round candidate with the highest predicted quality serve as the next mask prompt. During training, one randomly chosen later round adds no new click and refines the mask using the accumulated clicks and previous mask prompt.

C Evaluation Protocols and Additional Results

C.1 Evaluation Datasets

Multi-granularity evaluation sets. PartNetM contains 2,000 objects from 20 PartNet (Mo et al., 2019) categories, with 32,412 part masks. Table 8 reports the number of objects in each category. We construct Partita200 by annotating 200 additional objects with our multi-granularity pipeline. The objects are screened for overlap with the training set, and the automatically generated annotations undergo manual review. The resulting benchmark contains 6,441 masks. All methods use the same 100,000-point clouds, GT masks, and prompt indices.

Table 9: Additional baseline configurations for single-click segmentation (IoU@1, %). GT scale uses target-derived information.

Method	Setting	PartNetE	PartObjaverse-Tiny
P^3-SAM	Voxel 0.005	53.09	65.50
P^3-SAM	Voxel 0.01	68.31	59.64
S^{2}AM3D	Five fixed scales	63.75	52.02
S^{2}AM3D	GT scale	70.23	53.23

C.2 Evaluation Protocols

C.2.1 Interactive Segmentation

All methods share one uniformly sampled initial click per GT mask. Corrective clicks are placed in the deepest false-negative or false-positive regions of each method's prediction. IoU is averaged over masks within each object and then equally over objects.

C.2.2 Single-Click Multi-Granularity Evaluation

We sample a click uniformly from each finest annotated part and collect all ground-truth masks containing it. Hungarian matching maximizes total IoU between these targets and the full first-click candidate set, with unmatched targets assigned zero IoU. Match mIoU@1 averages over targets per click, then over clicks within each object and finally over objects.

C.2.3 Automatic Proposals and Optional Refinement

Proposal generation and evaluation. We select 100 independent positive prompts by deterministic farthest point sampling, starting from the point farthest from the normalized cloud's centroid. Each cloud is encoded once. Native outputs yield 300 raw proposals for Point-SAM, P^3-SAM, and PartSAM, and 500 for Partita.

Masks are binarized at zero logit and ranked by native quality scores, with ties resolved by candidate order. Greedy NMS suppresses candidates with point-mask IoU greater than 0.7 with a retained mask; at most 300 proposals per object are kept, without additional confidence or mask-area filtering. Class-agnostic AP uses 101 recall levels. AP and AR@300 are averaged over IoU thresholds $0.50{:}0.05{:}0.95$, with one-to-one matching between detections and GT masks at each threshold. Both metrics are computed over each complete dataset.

C.3 Baseline Settings and Additional Comparisons

C.3.1 Baseline Evaluation Settings

Candidate and baseline settings. Point-SAM, P^3-SAM, and PartSAM retain three native first-click candidates, while Partita produces five. For interactive evaluation, we use the better P^3-SAM voxel setting on each dataset (Table 9). S^{2}AM3D produces five scale-conditioned predictions using fixed scales. P^3-SAM and S^{2}AM3D are evaluated only at the first click.

Multi-granularity baseline settings. We evaluate the released baseline weights without retraining. For the multi-granularity comparison, P^3-SAM uses voxel size 0.005 on both datasets. S^{2}AM3D uses the five fixed scales $\{0.01, 0.03, 0.09, 0.27, 0.81\}$ for Match mIoU@1; its AP/AR are unreported because it has no native mask-quality score.

C.3.2 Alternative Baseline Configurations

Table 9 reports both P^3-SAM voxel settings and S^{2}AM3D scale settings. Both P^3-SAM rows are single-click results. The S^{2}AM3D GT-scale setting supplies the foreground-point fraction of the target mask, so it is a GT-assisted reference.

Table 10: Multi-granularity prediction and proposal generation with three or five candidates (%; higher is better). Five-candidate baselines combine two independent prompts; all other rows use one. Best values are in bold.

Method	Candidates	PartNetM			Partita200		
		Match mIoU	AP	AR@300	Match mIoU	AP	AR@300
Point-SAM	3	52.20	15.63	31.89	41.72	8.84	23.94
	5	59.15	15.19	32.77	49.69	8.65	24.84
P^3-SAM	3	60.07	19.25	35.55	62.91	38.40	52.37
	5	63.73	19.42	36.60	68.67	39.82	55.47
PartSAM	3	64.51	30.92	45.84	51.48	22.62	36.57
	5	70.71	**31.52**	**48.27**	59.46	23.16	38.70
Ours	5	**74.25**	28.17	45.96	**77.39**	**40.73**	**55.99**

Table 11: Multi-target supervision on Partita200: single-target versus set-to-set matching.

Matching	Match	AP	AR@300
Single-target	68.97	33.28	**54.65**
Set-to-set (Ours)	**76.74**	**39.38**	54.05

C.3.3 Three- and Five-Candidate Baseline Comparisons

We compare the native three-candidate outputs of the released baselines with five-candidate sets formed without retraining. The latter combine three masks from the original prompt with the two highest-scoring masks from a second independent prompt. For Match mIoU, the second point is randomly sampled within the same finest region with identical containing-GT membership, preserving the original target set. For AP/AR, we add 100 GT-independent random points to the original FPS100 and retain 500 proposals before the unchanged NMS. Partita uses its five native candidates from each single prompt.

Table 10 shows that Partita achieves the highest Match mIoU on both datasets and the best AP/AR on Partita200, while PartSAM achieves the best AP/AR on PartNetM. Notably, PartSAM uses PartNet, the source dataset of PartNetM, for training, whereas our training does not include PartNet.

C.3.4 Multi-Target Supervision

Table 11 reports the multi-target supervision ablation discussed in §5.4.

C.3.5 Encoder Initialization

We compare initialization from P^3-SAM with initialization from Sonata (Wu et al., 2025) without loading P^3-SAM weights. The latter retains pretrained encoder features and does not use random initialization. Both encoders are trainable. The Sonata-only configuration is trained for 100k updates with the same recipe as the decoder variants in Table 4; the P^3-SAM reference is our 100k-update model from that table. We evaluate both checkpoints on PartObjaverse-Tiny with the same uniformly sampled first click per GT mask, selecting the candidate with the highest GT IoU. Results are reported in Table 12.

Table 12: Encoder-initialization comparison on PartObjaverse-Tiny at 100k updates. The P^3-SAM reference follows Table 4.

Encoder initialization	IoU@1 (%)
P^3-SAM	**73.46**
Sonata-only	71.83

D Additional Visualizations and Failure Cases

D.1 Qualitative Comparisons

Figure 8 shows additional multi-granularity predictions from a single click, alongside the available ground-truth masks. Figure 9 compares single-click segmentation across methods, highlighting small parts and local boundaries.

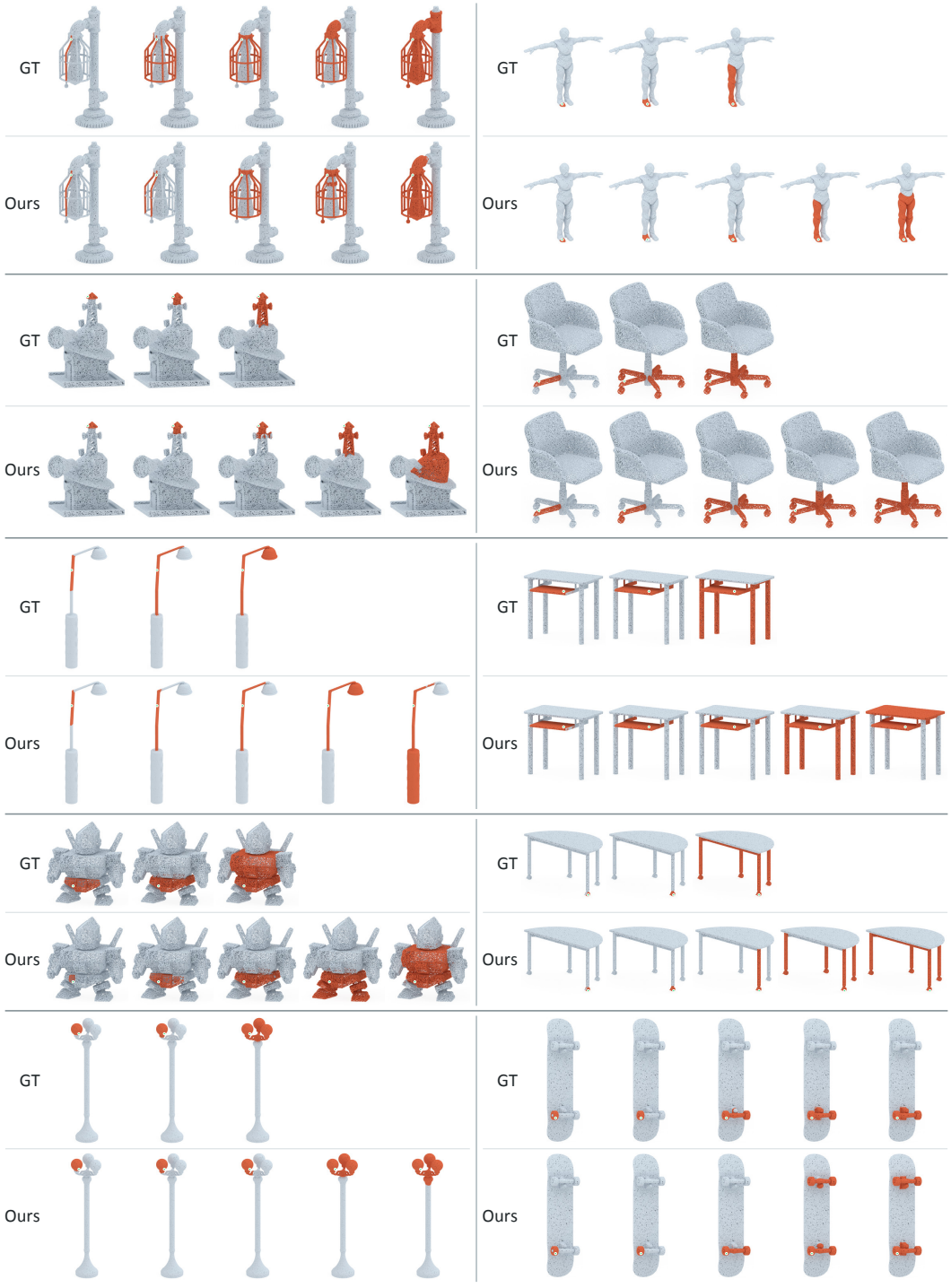

Figure 8: Additional single-click multi-granularity predictions. Ground-truth masks (top) and Partita predictions (bottom) are independently sorted by mask size. **Zoom in for a better view.**

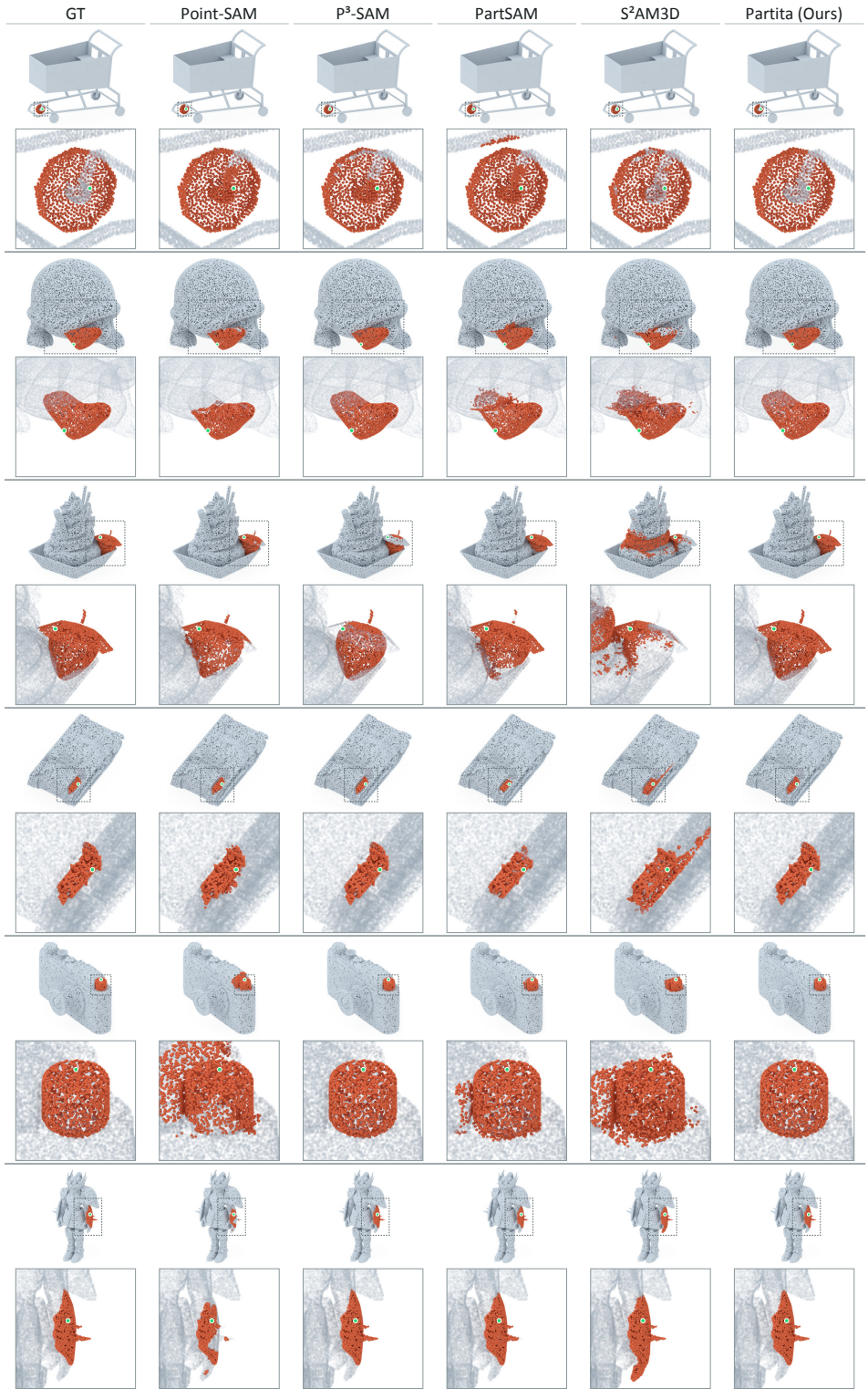

Figure 9: Additional single-click segmentation comparisons with local close-ups. For each method, we show the candidate with the highest ground-truth IoU. **Zoom in for a better view.**

D.2 Failure Cases

When an object has only a limited number of meaningful granularity levels, different prediction heads may still produce duplicate masks (Figure 10).

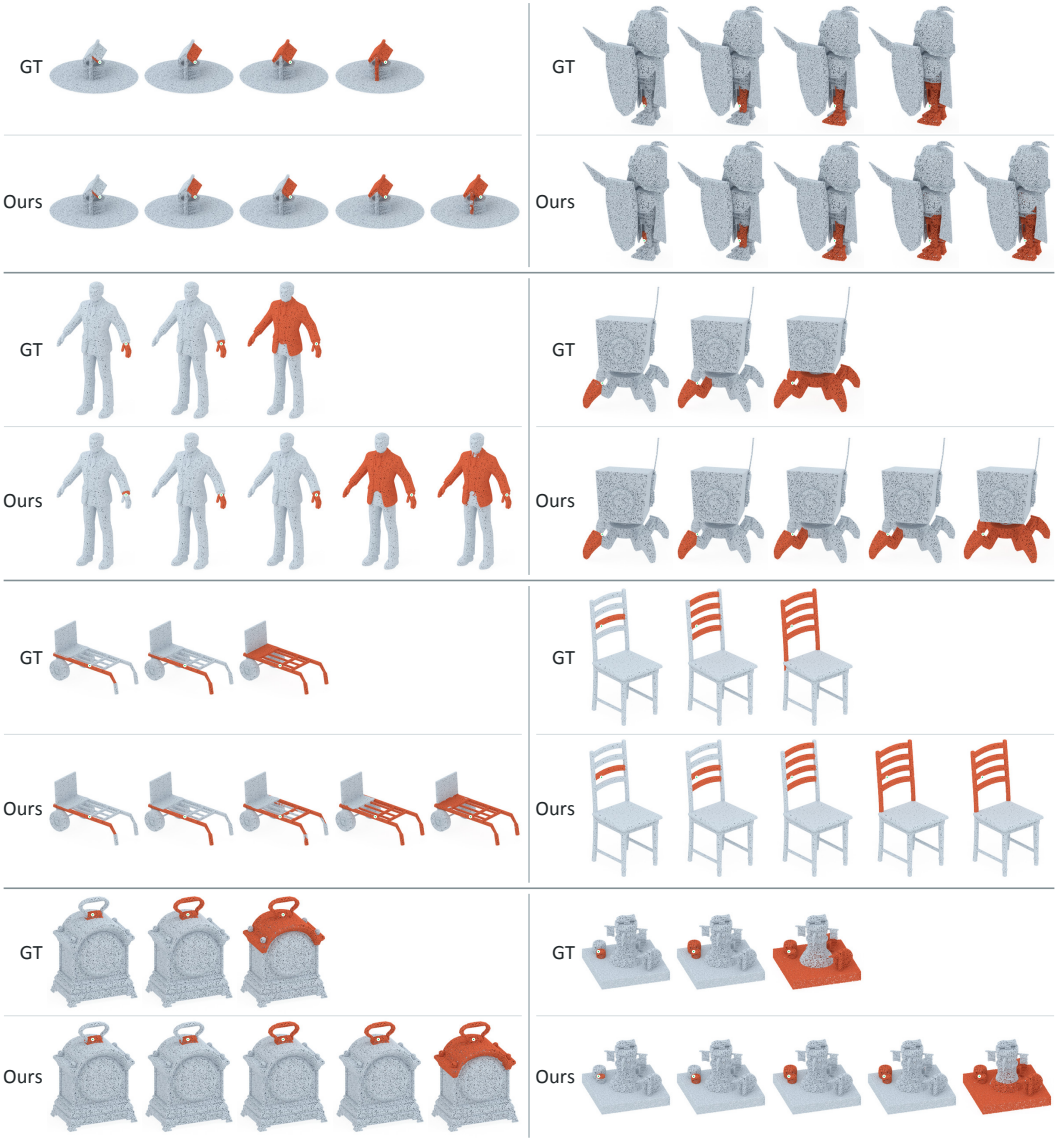

Figure 10: Failure cases of single-click multi-granularity segmentation. Ground-truth masks (top) and Partita predictions (bottom) are independently sorted by mask size. **Zoom in for a better view.**

E Discussion and Limitations

Our results show that granularity in promptable 3D segmentation can be learned as an output from supervision rather than specified by the user as an input. The key is that the organization of the data matches the training objective. Our data provide a complete nested chain from fine to coarse for each click, and training correspondingly matches the candidate set to this entire chain. Both are necessary: on the same multi-granularity data, matching only one candidate to a single target lowers Match mIoU@1 by 7.77 points (Table 11), whereas under a fixed mask budget, replacing single-level annotations with multi-granularity annotations raises Match mIoU@1 by 14.29 points despite fewer training objects (Table 3). Multi-granularity data are fully effective only when paired with the corresponding set-level supervision.

Our approach also has limitations. First, the model outputs only $K = 5$ candidates per click, which bounds the length of the chain that a single click can return. Chains longer than five account for only 0.08% of the original parts in PartNeXt-XL (Table 6), so the current impact is small, but a single click cannot return the complete chain for objects with deeper hierarchies. Second, the finest level of each hierarchy is determined by the original parts. Because these parts come from connected-component analysis, the pipeline can only group them into coarser regions and cannot split them further, so errors in the original partition persist across all levels. In addition, structural checks only guarantee the structural consistency of each hierarchy, and quality verification depends on the capability of the VLM itself, so the semantic soundness of every automatically generated grouping cannot be strictly guaranteed.

Looking ahead, a promising direction is to bring the model itself into the annotation loop. The current pipeline relies on the existing part partitions of the assets and therefore cannot readily handle data without usable parts, such as scan reconstructions and single-piece meshes. A trained Partita could predict multi-granularity parts for such assets as the starting point of the pipeline, the VLM could then organize them into hierarchical annotations, and the new data could in turn train a stronger model. As the model and data improve each other in this loop, hierarchical part annotation could extend beyond assets with existing parts to far larger object collections and even scene-level 3D data.