This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

S^{2}AM3D: Scale-controllable Part Segmentation of 3D Point Clouds

Han Su[1] Tianyu Huang[1] Zichen Wan[1] XiaoHe Wu[1] Wangmeng Zuo[1] *
[1]Harbin Institute of Technology

Abstract

Part-level point cloud segmentation has recently attracted significant attention in 3D computer vision. Nevertheless, existing research is constrained by two major challenges: native 3D models lack generalization due to data scarcity, while introducing 2D pre-trained knowledge often leads to inconsistent segmentation results across different views. To address these challenges, we propose S^{2}AM3D, which incorporates 2D segmentation priors with 3D consistent supervision. We design a point-consistent part encoder that aggregates multi-view 2D features through native 3D contrastive learning, producing globally consistent point features. A scale-aware prompt decoder is then proposed to enable real-time adjustment of segmentation granularity via continuous scale signals. Simultaneously, we introduce a large-scale, high-quality part-level point cloud dataset with more than 100k samples, providing ample supervision signals for model training. Extensive experiments demonstrate that S^{2}AM3D achieves leading performance across multiple evaluation settings, exhibiting exceptional robustness and controllability when handling complex structures and parts with significant size variations. The project page is available at https://sumuru789.github.io/S2AM3D-website/.

1. Introduction

Part-level point cloud segmentation plays a pivotal role in bridging fine-grained geometric details with high-level semantic understanding, supporting significant applications in 3D content creation, robotic manipulation, and reverse engineering [3, 9, 11, 17, 20, 37], *etc.*Unlike instance detection, which only provides holistic semantics, part-level segmentation enables flexible granularity adjustment and seamless switching between local and global regions. This capability directly influences the feasibility and efficiency of downstream tasks, including part generation, replacement, assembly, and parametric editing. However, high-quality 3D part-level data remains scarce. Compared with the 2D domain, the cost of point cloud annotation is more prohibitive; consequently, existing datasets suffer from limited scale and category diversity, which severely hinders the generalization performance of 3D models [15, 23, 27, 28, 32].

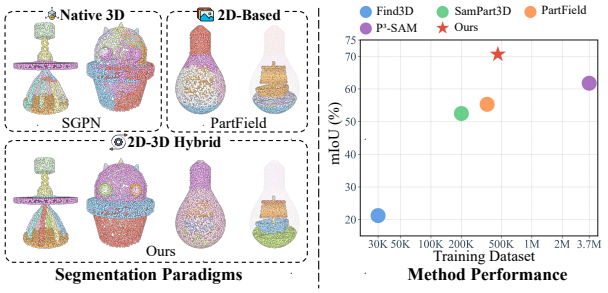


Figure 1. Paradigm comparison (left): Native 3D methods present limited generalization, and 2D-based methods fail in complex cases like occlusions. Our hybrid solution solves these issues. Performance Comparison (right): Our method reaches large-scale training performance with much less data and significantly outperforms previous methods at similar data scales.

Recent works [8, 18, 19, 30, 35, 39, 40] leverage 2D pre-trained knowledge to enhance the generalization of 3D models, *i.e.*, adopting 2D segmentation on 3D renderings. Segmented multi-view contents are then aggregated via mechanisms like multi-view lifting [8, 18, 30, 39] or distillation [19, 35, 40] to obtain the final 3D segmentation results. However, cross-view inconsistencies caused by occlusions, slender structures, and complex topology, can lead to accumulated errors that compromise global 3D consistency. To alleviate the inconsistency, some methods [19, 40] further incorporate 2D priors and provide supervision in the 3D space. Although native 3D data is used to enhance multi-view consistency, these methods still rely too heavily on 2D segmentation results. For example, PartField [19] primarily leverages part proposals from SAM [13], supplemented with limited 3D supervision to learn multi-view dependencies. Consequently, when 2D segmentation is affected by occlusions and other spatial factors, 3D representations struggle to correct these issues (see Figure 1).

In contrast to the 2D domain, point cloud segmentation is more challenging, considering its requirements of integrating both global and local context. The absence of global information can compromise multi-view consistency,

while an excessive focus on global features may cause distortions in the correlation between local parts. To better combine the spatial features with 2D priors, we introduce **S^{2}AM3D**, a multi-modal joint-supervised framework for scale-aware point cloud part segmentation. We propose a point-consistent part encoder to aggregate global information from multi-view 2D features with contrastive supervision of point cloud data. Additionally, given the inherent weakness of global representations in encoding the relationships of local parts, we propose a scale-aware prompt decoder to explicitly model cross-granularity relations. To this end, the input scale is mapped to a learnable sinusoidal embedding and then fed to bi-directional cross-attention modules together with encoded global point features. Using the prompt point as a query, the attention layers fuse multi-granularity context to finally determine the per-point probabilities in one pass.

Currently, existing part-level point cloud datasets generally suffer from limited scale and insufficient annotation fidelity. To support our point-consistent 3D supervision, we introduce an automatic data pipeline for scalable part-level point cloud data curation, including part annotation, quality filtering, and connectivity refinement. Filtering and refinement strictly enforce 3D geometric constraints on the raw annotations, ensuring the quality of processed labels. Using this pipeline, we collect a dataset of more than 100,000 point cloud instances across 400 categories, annotated with about 1.2 million fine-grained part labels.

Extensive experiments are conducted to demonstrate the effectiveness of S^{2}AM3D on part-level point cloud segmentation. As shown in Figure 1, our 2D-3D hybrid training recipe allows S^{2}AM3D to achieve comparable performance to P^3-SAM [23] with far less training data, and significantly outperforms SAMPart3D and PartField. Benefitting from the proposed scale-aware prompt decoder, S^{2}AM3D exhibits remarkable flexibility by naturally extending its capability to point-prompted segmentation, making it a unified 3D part segmentation framework.

Our contributions can be summarized as:

- We propose a 2D-3D training recipe for part segmentation: it reuses 2D pre-trained knowledge and conducts native 3D supervision to yield globally point-consistent part features.
- We propose a scale-aware prompt decoder with a scale modulator and bi-directional cross-attention, enabling flexible 3D part segmentation.
- We introduce a scalable data pipeline for part-level point cloud data curation, collecting over 100,000 labeled point cloud instances for training.

2. Related Work

Native 3D Part Segmentation. Early point cloud/mesh segmentation methods were typically trained under closed-label settings on ShapeNet-Part and PartNet, leveraging pointwise or neighborhood aggregation networks such as PointNet/PointNet++ and the Point Transformer family [25, 27, 28, 36, 38]. While strong within restricted categories and fixed granularities, these models often struggle under open-domain shapes, long-tailed parts, and cross-source shifts: (i) limited category/hierarchy coverage hampers generalization to unseen objects and rare parts; (ii) geometric-detail and noise discrepancies across sources obscure unified supervision; and (iii) the absence of explicit hierarchical consistency and granularity control constrains editing and interactive scenarios [2, 4, 25]. These limitations motivate scalable part understanding beyond closed-world assumptions.

2D Transferred Part Segmentation. To mitigate the scarcity of native 3D annotations, many works transfer 2D foundation priors (e.g., SAM, CLIP/GLIP) to 3D [13, 14, 29]. Two mainstream paradigms emerge: (i) *multi-view lifting* renders shapes to images, applies 2D segmentation or vision–language priors per view, and fuses/back-projects/distills view-level cues into 3D supervision [13, 18, 33, 34]; and (ii) *2D-to-3D distillation* directly trains native 3D networks from image-domain pseudo labels [10, 19, 35]. Despite reduced labeling cost and open-vocabulary semantics, both paradigms are prone to cross-view inconsistencies on thin/occluded structures, boundary ambiguity that undermines global 3D coherence, and costly post-processing pipelines; moreover, surface-only coverage limits supervision of internal structures [10, 18, 33, 34]. These observations point to the value of native 3D signals for globally consistent part supervision.

Granularity Controllable Part Segmentation. Existing methods still face significant limitations in achieving flexible control over segmentation granularity. Feature clustering-based approaches (e.g., PartField [19] and SAMPart3D [35]) typically rely on post-processing clustering to determine segmentation granularity, resulting in non-continuous and non-intuitive control that hinders real-time fine-grained adjustment. Point prompt-based methods (e.g., Point-SAM [40] and P^3-SAM [23]), while supporting interactive point prompting, lack explicit granularity control mechanisms and cannot precisely regulate segmentation scope through continuous parameters. Furthermore, the segmentation granularity in both types of methods remains fundamentally constrained by their reliance on 2D priors, making it challenging to ensure global consistency in 3D space. These limitations motivate us to explore a new solution that introduces an explicit continuous scale signal coupled with native 3D supervision, ultimately achieving truly reliable and controllable 3D segmentation granularity.

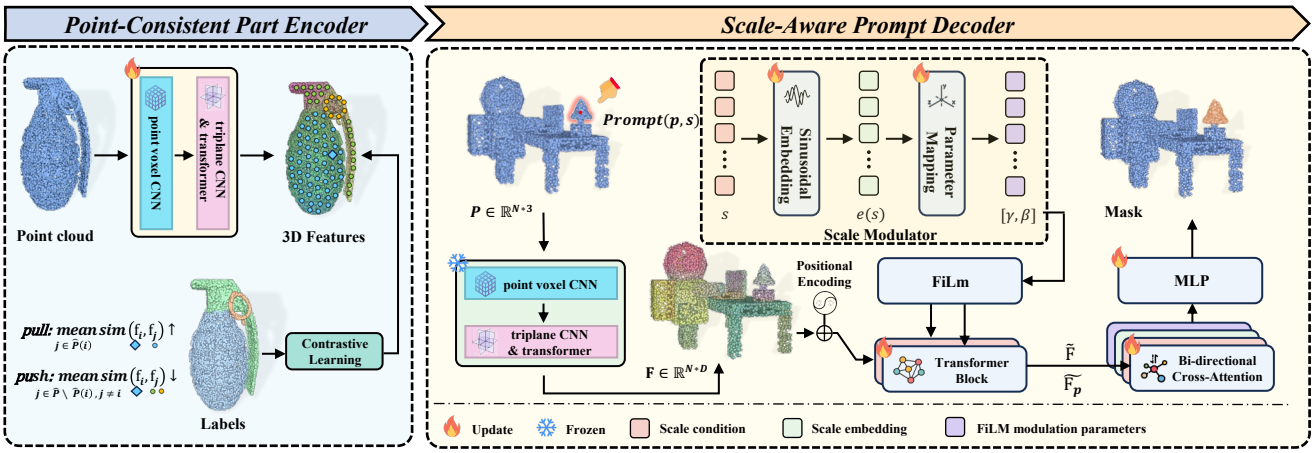


Figure 2. S^{2}AM3D pipeline. Left: under 3D supervision with contrastive learning, the input point cloud $\mathbf{P} \in \mathbb{R}^{N\times 3}$ is encoded into per-point features $\mathbf{F} \in \mathbb{R}^{N\times D}$. Right: given a prompt (p, s), s is mapped by a sinusoidal embedding $\mathbf{e}(s)$ to FiLM parameters $[\gamma, \beta]$, which perform channel-wise modulation to obtain a scale-enhanced representation $\tilde{\mathbf{F}}$; the prompt vector $\tilde{\mathbf{F}}_p$ is then indexed and interacts with the global features via bi-directional cross-attention, after which an MLP and a Sigmoid produce a probability mask.

3. Method

3.1. Overview

We propose **S^{2}AM3D**, a *scale-controllable*, *point-prompted* framework for part segmentation of 3D point clouds. Given the input point cloud $P \in \mathbb{R}^{N\times 3}$, we first extract point-consistent features by incorporating 2D segmentation priors with 3D contrastive supervision (Sec. 3.2). Conditioned on a point prompt index p with an optional scale prompt $s \in [0, 1]$, our Scale-Aware Prompt Decoder enables flexible part segmentation via scale modulator and bi-directional cross-attention (Sec. 3.3). We adopt a decoupled training scheme, *i.e.*, first stabilizing the encoder representation, then training the scale-aware decoder on top of the frozen encoder (Sec. 3.4). The overall framework is shown in Figure 2.

3.2. Point-Consistent Part Encoder

To aggregate point cloud features with general segmentation priors, the common practice is to train a point cloud encoder by distilling the pre-training knowledge from 2D segmentation models like SAM [13]. Following PartField [19], we exploit a voxel-based encoder PVCNN [21] to extract the point latent and then convert it to the tri-plane [1] (xy, yz, and zx) representation with orthogonal projection. The Tri-plane features $\mathbf{T} \in \mathbb{R}^{3\times D\times H\times W}$ are then aggregated by a series of Transformer blocks, where D is the feature channel. During training, tri-plane features are rendered from random viewpoints to formulate a 2D latent, which can be supervised with the distillation of SAM. However, relying solely on multi-view 2D distillation tends to introduce multi-view inconsistencies, leading to segmentation problems like boundary artifacts and cross-view conflicts.

Therefore, we additionally perform a *native 3D contrastive supervision* under labeled point cloud data to enhance global consistency. To operationalize this objective in the construction of contrastive pairs, we restrict contrasts to *intra-instance*, *i.e.*, each mini-batch contains a single object. This mechanism ensures that positive and negative points are drawn from the same instance to avoid cross-instance semantic mismatches. We randomly subsample points from a single instance and denote their indices by $\hat{P}$ (with $|\hat{P}| = \hat{N}$). For any anchor $i \in \hat{P}$ with label y_i, the positive set is:

$$\hat{P}(i) = \{\, j \in \hat{P} \setminus \{i\} \mid y_j = y_i \,\}.$$

We use cosine similarity with temperature τ between ℓ_2-normalized features $\mathbf{f}_i$ and $\mathbf{f}_j$: $s_{ij} = \mathbf{f}_i^\top \mathbf{f}_j / \tau$. The contrastive objective is

$$\mathcal{L}_{\text{contr}} = \frac{1}{|\hat{P}|} \sum_{i\in\hat{P}} -\log \frac{\sum_{j\in\hat{P}(i)} e^{s_{ij}}}{\sum_{j\in\hat{P}\setminus\{i\}} e^{s_{ij}}}.$$

To obtain point features $\mathbf{F}$ given the 3D coordinate (x, y, z) from the tri-plane representation [1], we unproject the point to the three feature planes of $\mathbf{T}$. The sampled features are summed across planes and stacked over all points as:

$$\mathbf{F} = \Big[\, \mathbf{T}_{xy}(x_n, y_n) + \mathbf{T}_{yz}(y_n, z_n) + \mathbf{T}_{zx}(z_n, x_n) \,\Big]_{n=1}^{N} \tag{1}$$

This objective compacts intra-part clusters and enlarges inter-part margins, yielding globally coherent per-point embeddings and sharper boundaries for the subsequent scale-aware prompt decoding.

3.3. Scale-Aware Prompt Decoder

Given point features $\mathbf{F} \in \mathbb{R}^{N\times D}$ and 3D coordinates $\mathbf{P} \in \mathbb{R}^{N\times 3}$, we compute a 3D sinusoidal positional encoding

$\mathrm{PE}(\mathbf{P})$ and obtain the base representation

$$\mathbf{X}^{(0)} = \mathbf{F} + \mathrm{PE}(\mathbf{P}). \tag{2}$$

However, such global representations often struggle to capture fine-grained intra-class multi-scale variations and local relations among parts, whereas interactive applications require continuously controllable granularity from fine to coarse. Therefore, beyond the point prompt, we further introduce the *scale* prompting to guide feature decoding and adopt bi-directional cross-attention to enhance the point prompting.

Scale Modulator. The scale s is defined by the part's relative size, which is the ratio of its points to the total number of points. Scale information should influence the global representation in a *cross-layer transferable* manner. For a continuous scale $s \in [0, 1]$, we construct a learnable sinusoidal embedding

$$\mathbf{e}(s) = \big[\sin(\omega_k s + \phi_k),\ \cos(\omega_k s + \phi_k)\big]_{k=1}^{M}, \tag{3}$$

where $\{\omega_k, \phi_k\}$ are learnable and M denotes the number of sinusoidal frequency pairs. $\mathbf{e}(s)$ is then fed to a feature-wise linear modulation [26] (FiLM) in the channel dimension, which modulates the global feature map with a learnable gate α:

$$\begin{aligned} [\boldsymbol{\gamma}, \boldsymbol{\beta}] &= \mathrm{Linear}\big(\mathrm{LN}(\mathbf{e}(s))\big), \\ \mathrm{FiLM}(\mathbf{X}; s) &= \mathbf{X} \odot (1 + \alpha\boldsymbol{\gamma}) + \alpha\boldsymbol{\beta}. \end{aligned} \tag{4}$$

where Linear() is a linear layer applied after Layer Normalization on $\mathbf{e}(s)$, $\boldsymbol{\gamma}, \boldsymbol{\beta} \in \mathbb{R}^D$ are channel-wise FiLM parameters [26], and $\alpha \in \mathbb{R}$ is a learnable scalar gate. To maintain a consistent scale context across semantic levels, we interleave FiLM with Transformer blocks T:

$$\mathbf{X}^{(\ell+1)} = T_\ell\big(\mathrm{FiLM}(\mathbf{X}^{(\ell)}; s)\big), \qquad \ell = 0, \ldots, L_m - 1. \tag{5}$$

The resulting scale-conditioned enhanced representation is

$$\tilde{\mathbf{F}} = \mathbf{X}^{(L_m)}. \tag{6}$$

To support inference without scale input, we randomly drop out the scale as $\mathbf{e}(s) = \mathbf{0}$ during training. FiLM naturally degenerates to the identity, i.e., $\mathrm{FiLM}(\mathbf{X}; 0) = \mathbf{X}$.

Bi-directional Cross-Attention. Unidirectional cross-attention struggles to perform both context aggregation and fine-grained refinement in one pass. To address this, we adopt *bi-directional* cross-attention for joint localization and refinement. We have scale-aware features $\tilde{\mathbf{F}} \in \mathbb{R}^{N\times D}$ and the corresponding feature of the point prompt $\tilde{\mathbf{F}}_p \in \mathbb{R}^{1\times D}$. Let the input to the ℓ-th layer be $(\mathbf{Y}^{(\ell)}, \mathbf{q}^{(\ell)})$, with $\mathbf{Y}^{(0)} = \tilde{\mathbf{F}}$ and $\mathbf{q}^{(0)} = \tilde{\mathbf{F}}_p$. Each layer comprises two cross-attention steps with explicit Q/K/V roles, followed by an FFN:

$$\begin{aligned} \mathrm{CAttn}(A; B) &= \mathrm{MHA}(Q{=}A,\ K{=}B,\ V{=}B), \\ \mathbf{q}^{(\ell+1)} &= \mathbf{q}^{(\ell)} + \mathrm{CAttn}\big(\mathbf{q}^{(\ell)};\ \mathbf{Y}^{(\ell)}\big), \\ \mathbf{Y}^{(\ell+1)} &= \mathrm{FFN}\Big(\mathbf{Y}^{(\ell)} + \mathrm{CAttn}\big(\mathbf{Y}^{(\ell)};\ \mathbf{q}^{(\ell+1)}\big)\Big), \end{aligned} \tag{7}$$

where MHA denotes *MultiHeadAttention*, and $\ell = 0, \ldots, L_d - 1$. After stacking L_d layers, we obtain interaction-enhanced features $\mathbf{H} = \mathbf{Y}^{(L_d)}$, and a lightweight segmentation head outputs pointwise probabilities:

$$\mathbf{o} = \mathrm{MLP}(\mathbf{H}), \qquad \hat{\mathbf{m}} = \sigma(\mathbf{o}) \in [0, 1]^N. \tag{8}$$

3.4. Training Objective

We adopt a decoupled training scheme: our point-consistent part encoder is trained with the contrastive objective [12] in Sec. 3.2, while the scale-aware prompt decoder is optimized subsequently with the encoder frozen.

In the second stage, we randomly sample one target part per object, which is selected following the strategy of our experimental setup (Sec. 5.2).

However, each sample annotates only one part as positive, resulting in a systematically small within-sample positive ratio π in this setting. Using BCE alone tends to bias the optimization toward negatives/background and can admit a degenerate all-negative solution. To mitigate this bias while balancing set-level overlap and pointwise calibration, we employ a hybrid objective that combines Dice with a dynamically reweighted BCE. Given the predicted probability mask $\hat{\mathbf{m}}$ and ground truth $\mathbf{m} \in \{0, 1\}^N$, the segmentation loss is

$$\mathcal{L}_{\text{seg}} = \lambda_{\text{bce}}\, \mathrm{BCE}_{\text{dyn}}(\hat{\mathbf{m}}, \mathbf{m}) \ + \ \lambda_{\text{dice}}\Big(1 - \tfrac{2\,\hat{\mathbf{m}}^\top \mathbf{m}}{\|\hat{\mathbf{m}}\|_1 + \|\mathbf{m}\|_1}\Big). \tag{9}$$

This design simultaneously optimizes pointwise probabilities and set-level overlap: $\mathrm{BCE}_{\text{dyn}}$ calibrates probabilities, whereas the Dice term directly improves set-level overlap and is more robust to class imbalance, thereby improving recall and maintaining gradient stability for small parts, long-tailed cases, and sparse masks. The dynamically reweighted BCE is defined as

$$\begin{aligned} \mathrm{BCE}_{\text{dyn}} &= -\frac{1}{N}\Big(\beta \sum_{j\in J_+} \log \hat{m}_j + \sum_{j\in J_-} \log(1 - \hat{m}_j)\Big), \\ \beta &= \frac{1-\pi}{\pi + \varepsilon}, \quad \pi = \frac{1}{N}\sum_{j=1}^{N} m_j, \\ J_+ &= \{\, j \mid m_j = 1 \,\}, \quad J_- = \{\, j \mid m_j = 0 \,\}, \end{aligned} \tag{10}$$

where β is adaptively computed from the per-sample positive ratio π to mitigate class imbalance, and $\varepsilon > 0$ is a numerical stabilizer.

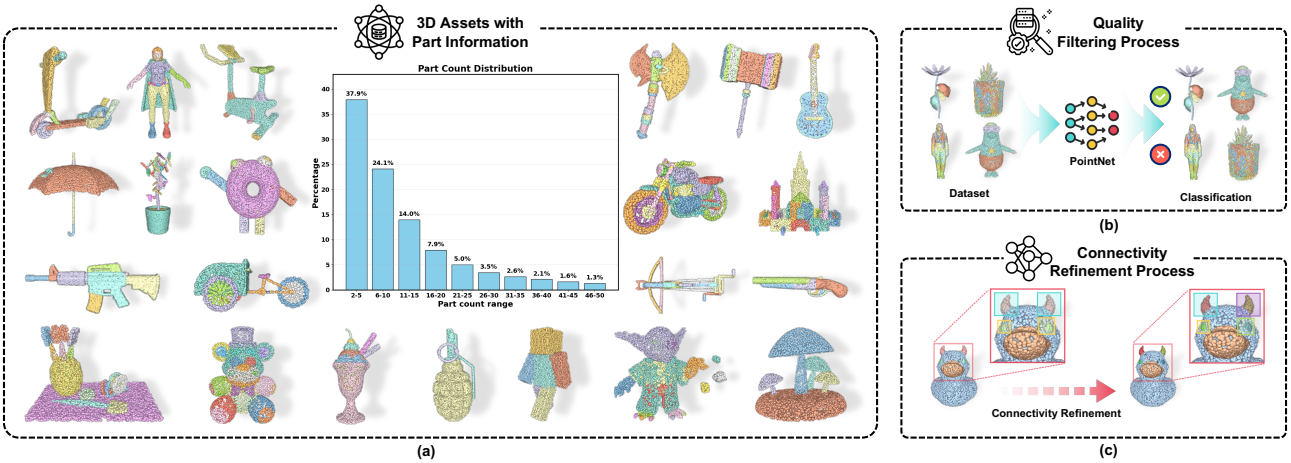


Figure 3. Dataset overview: covering diverse categories and providing high-quality part-level annotations; the histogram shows the long-tailed distribution of part counts.

4. Dataset Curation

To support 3D-supervised training, we introduce a scalable data pipeline for point cloud part segmentation labeling and curate a large-scale, high-quality labeled point cloud part segmentation dataset. The pipeline comprises part annotation, quality filtering, and connectivity refinement to ensure robustness and diversity. We collect over 100,000 point cloud instances with around 1.2 million part labels, which, to the best of our knowledge, is one of the largest publicly available 3D part segmentation datasets.

4.1. Scalable Data Pipeline

The data pipeline consists of three steps: part annotation, quality filtering, and connectivity refinement. Given a 3D object, we adopt surface-area–proportional sampling and assign the corresponding part label to every sampled point, which follows the part-annotation mining strategy of PartCrafter [16]. However, such an automatic strategy inevitably introduces erroneous annotation artifacts, which may mislead the model training. To filter out possible wrong labels, we further propose quality filtering and connectivity refinement.

In the quality filtering stage (see Figure 3(b)), we first manually curate a small-scale dataset containing examples of both reasonable and unreasonable part annotations. We then train a binary PointNet [27] validator on this dataset. Each point cloud is represented by N points, each defined by a 4D vector (x, y, z, l) encompassing normalized coordinates and its part-label index. Once trained, this validator is applied across the entire dataset to automatically screen and filter out samples with unreasonable annotations, ensuring that only high-quality labeled data proceed to subsequent stages.

We then perform connectivity refinement (see Figure 3(c)): artist annotations may group spatially disconnected yet semantically related regions under the same label, which degrades prompt-segmentation precision and hampers isolating local structures. For the points of each label, we compute the axis-aligned bounding-box diagonal length d and set the DBSCAN [7] radius to $\epsilon = d \times \epsilon_{\text{factor}}$. If multiple spatial clusters are detected within a single label, we split it into distinct new labels. Finally, we filter and retain objects containing 2 to 50 parts to ensure dataset balance and suitability for part segmentation tasks.

4.2. Dataset Overview

We mainly collect 3D assets from Objaverse [5, 6], which is one of the largest 3D object datasets. We filter more than 100,000 point cloud instances from this dataset, spanning over 400 categories. This category advantage lies in covering open-world shapes from simple objects to complex machinery, supporting broader geometric and semantic diversity. In total, we annotate around 1.2 million part labels, with an average of 11 parts per object. As shown in Figure 3(a), the part count distribution spans 2-50, exhibiting a long-tailed pattern to ensure balanced coverage of fine- and coarse-grained shapes.

5. Experiments

5.1. Implementation Details

We adopt a decoupled supervision strategy to separately train our proposed encoder and decoder. During training, we fix the tri-plane feature dimensions D, H, W to $448, 512, 512$, respectively. The number of sampling points N is set to 10,000.

Encoder training. The encoder is initialized with PartField [19] pre-trained parameters and then further optimized with contrastive learning on native point cloud annotations.

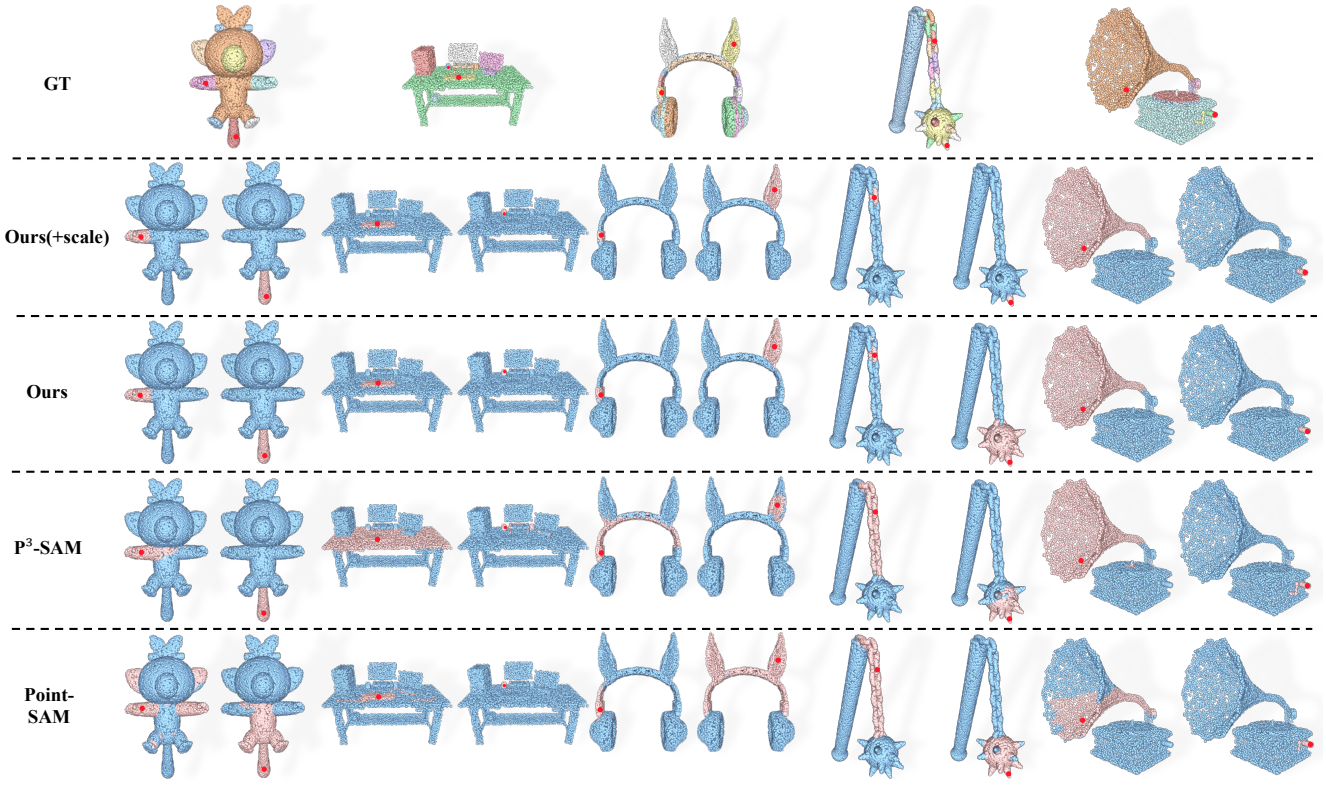


Figure 4. Qualitative comparison on our curated dataset (see Sec. 4.2). With a point prompt, S^{2}AM3D responds more accurately to the target, producing masks with cleaner boundaries and more complete topology.

For each mini-batch, we uniformly sample $\hat{N} = 1000$ points to form $\hat{P}$ and compute the contrastive loss on it with temperature $\tau = 0.07$. We use AdamW [22] with a learning rate of 1×10^{-5}. The encoder is trained for 15 epochs, which takes about one day on 8 NVIDIA A6000 GPUs.

Decoder training. In this phase, we freeze the parameters of the encoder and fine-tune the decoder only. We set the layer number of scale modulation L_m to 2 and the layer number of bi-directional cross-attention blocks L_d to 4. The segmentation loss is a weighted sum of dynamic BCE and Dice, with $\lambda_{\text{bce}} = 0.7$ and $\lambda_{\text{dice}} = 0.3$. During training, we apply the scale dropout with a probability of 0.1, and at inference, we threshold the probability at $\theta = 0.7$. We use AdamW [22] with a learning rate of 1×10^{-4}. The training batch size is 40. The decoder is trained for 110 epochs with 4 NVIDIA A6000 GPUs.

Training data. We train both the encoder and decoder on our curated dataset (Sec. 4.2) and PartNet, removing PartNet instances whose IDs appear in PartNet-E to prevent data leakage.

5.2. Comparison

We compare S^{2}AM3D with state-of-the-art methods on two tasks: *(i) Interactive Segmentation* and *(ii) Full Segmentation*. Interactive segmentation focuses on generating a single segment corresponding to the prompt point, while full segmentation requires predicting part labels for all the points. These two tasks demand a robust perception of both local point coherence and global semantic modeling.

Table 1. Quantitative comparison of interactive part-level segmentation. We report IoU (%) on PartObjaverse-Tiny and PartNet-E. We also provide a scale-aware version `(+scale)`, which has an additional scale prompt. The best score in each column is highlighted in **bold**, and underline denotes the second best.

Method	PartObjaverse-Tiny	PartNet-E	*Avg.*
Point-SAM [40]	31.46	50.23	40.85
P^3-SAM [23]	35.05	39.98	37.52
Ours	46.47	62.52	54.50
Ours (+scale)	**61.19**	**77.51**	**69.35**

Evaluation Datasets. We choose PartObjaverse-Tiny [35] and PartNet-E [18] for experimental evaluation. PartObjaverse-Tiny [35] is a compact subset of Objaverse [5, 6], containing 200 samples across 8 categories with manual part annotations. PartNet-E [18] provides 1,906 point cloud shapes across 45 categories, which can be used to assess cross-category generalization.

Baselines. For interactive segmentation, we compare two point-prompt baselines: Point-SAM [40] and P^3-SAM [23].

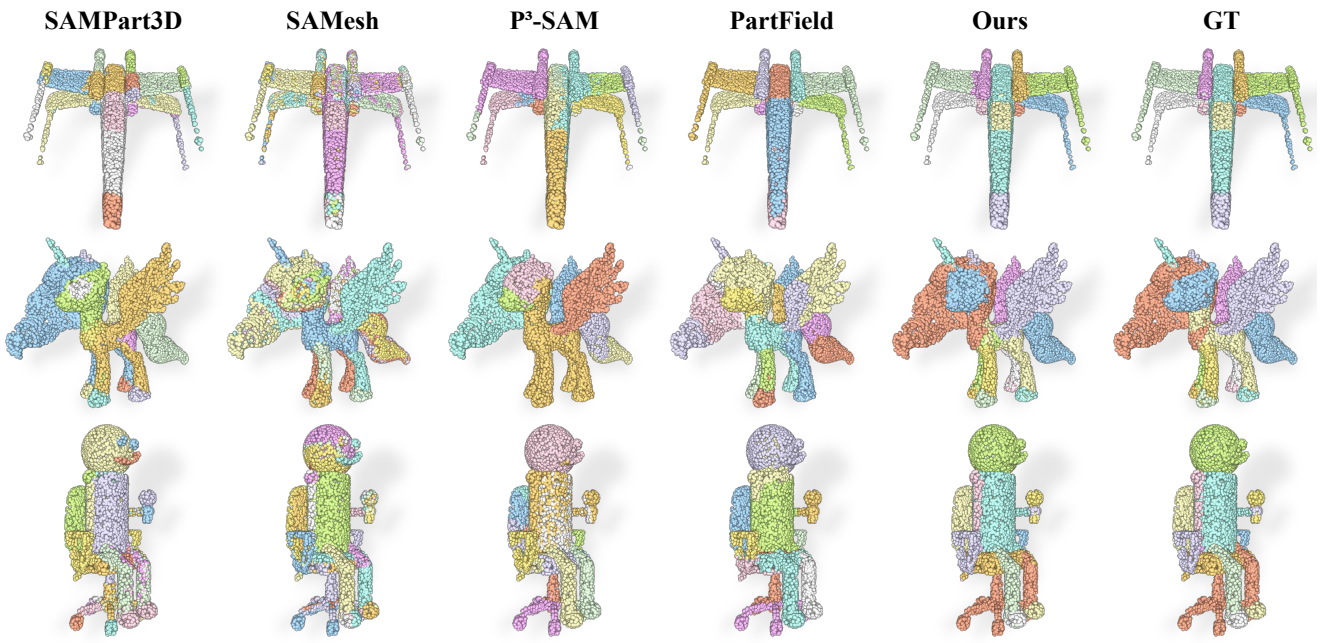


Figure 5. Qualitative comparison of full segmentation (PartObjaverse-Tiny [35]). For ease of comparison with our point cloud method, mesh-level outputs are presented as point clouds by uniformly sampling the segmented meshes.

Table 2. Quantitative comparison of full segmentation.

Method	PartObjaverse-Tiny	PartNet-E	*Avg.*
Find3D [24]	20.76	21.69	21.23
SAMPart3D [35]	48.79	56.17	52.48
SAMesh [31]	-	26.66	-
PartField [19]	51.54	59.10	55.32
P^3-SAM [23]	58.10	65.39	61.75
Ours	**63.29**	**77.98**	**70.64**

For fair comparison, all methods are evaluated with their default input configurations: Point-SAM [40] employs both coordinates and color information, while P^3-SAM [23] utilizes coordinates and surface normals. In contrast, our approach requires only XYZ coordinates. For full segmentation, we select current state-of-the-art methods, including Find3D [24], SAMPart3D [35], SAMesh [31], PartField [19], and P^3-SAM [23].

Metrics. We use Intersection over Union (IoU) as the evaluation metric. The metric is computed across all parts of a single object to obtain the per-sample score, which is subsequently averaged over the entire dataset. For interactive segmentation, we follow the strategy in Point-SAM [40], which selects the prompt point for each part as the interior point farthest from its boundary. All methods are evaluated on 10,000 points normalized to the unit sphere. For full segmentation, we follow the experimental protocol in P^3-SAM [23]. For our method, a point prompt is provided to the model for each ground-truth part to generate its mask. These masks are then consolidated to yield a comprehensive segmentation for the entire object. Please refer to Suppl. for more information.

Results. We report quantitative results of interactive segmentation in Table 1. Although our method can optionally incorporate scale prompts to boost performance, we also provide a non-scale version to ensure a fair comparison with the other two methods [23, 40]. Table 1 shows that S^{2}AM3D achieves the best results in both datasets [18, 35]. With the input of scale prompts, our method further achieves a performance gain of 14.72% and 14.99%, demonstrating the effectiveness of our scale-aware decoding. The qualitative results in Figure 4 are basically consistent with the quantitative results. Our model produces segmentations largely consistent with the GT without scale condition; supplying scale guidance corrects the few cases of granularity mismatch by calibrating the level of detail.

For full segmentation, Table 2 shows that S^{2}AM3D achieves 63.29 mIoU on PartObjaverse-Tiny and 77.98 mIoU on PartNet-E [18]. We also visualize the segmentation results in Figure 5. Constrained by 2D priors, methods like PartField [19] and SAMPart3D [35] produce segmentations with poor 3D consistency. On the other hand, 3D native methods like SAMesh [31] and P^3-SAM [23] tend to fail in some long-tailed cases. In contrast, S^{2}AM3D effectively circumvents these issues, achieving segmentation with higher completeness and better consistency on complex structures.

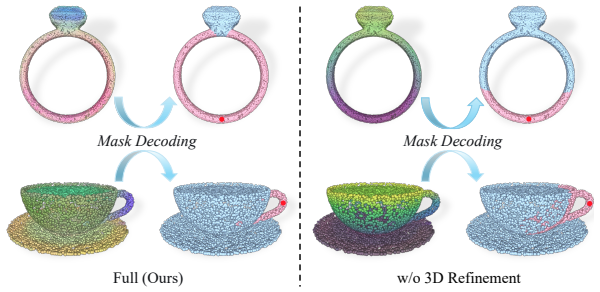


Figure 6. Visualization of the ablation study on encoder feature quality. This side-by-side comparison displays extracted features and their resulting segmentations for our full model (left) and the ablated model without 3D refinement (right).

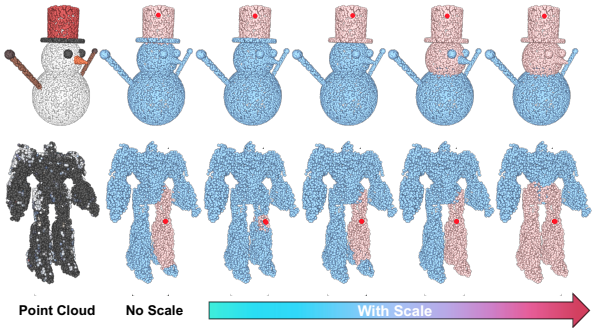


Figure 7. Visualization of continuous scale controllability. With the same prompt point, as the scale s increases from 0 to 1, the segmentation transitions smoothly from fine to coarse; a `(No scale)` counterpart is also provided for reference.

5.3. Ablation Studies

To demonstrate the effectiveness of our proposed modules and data, we conduct ablation studies in Table 3.

3D Contrastive Supervision. Among all ablation variants, replacing an encoder without 3D supervision yields the most significant performance degradation. This finding unequivocally establishes that encoder-side 3D feature refinement is the principal contributor to the observed gains. The rationale behind this substantial performance gap is vividly illustrated in Figure 6: features extracted by the ablated model exhibit noticeable blurry boundaries and internal inconsistencies compared to our pre-trained encoder. This disparity in feature quality directly translates to contrasting decoding outcomes: the full model generates coherent and accurate segmentation masks, whereas the ablated model produces fragmented and inconsistent results. This analysis clearly demonstrates that achieving improved feature consistency and geometric fidelity is paramount for high-quality point cloud segmentation.

Our Curated Data. To verify the impact of our own data, we train our model with PartNet [25] dataset, which results in performance drops compared to the model trained with our data. This finding strongly indicates that our curated data, having undergone systematic filtering and annotation governance, provides a crucial distributional complement to existing benchmarks. Specifically, it introduces richer geometric variation and diverse part composition, which directly translates into stable cross-dataset generalization gains for our model.

Setting	PartObjaverse-Tiny	PartNet-E	Avg.
+scale			
Full	**61.19**	**77.51**	**69.35**
w/o 3D supervision	53.94	64.11	59.03
w/o our data	53.12	66.12	59.62
No scale			
Full	**46.47**	**62.52**	**54.50**
w/o 3D supervision	41.14	55.39	48.27
w/o our data	42.12	58.56	50.34
w/o scale embedding	42.31	58.28	50.30

Table 3. Ablation studies (mIoU, %). Groups indicate whether scale information is provided at test time: **+scale** means a scale condition is given, while **No scale** means it is not.

Scale conditioning. We visualize the effectiveness of scale prompting in Figure 7. The results show that our method can provide reasonable segmentation without scale information, while scale conditioning allows more flexible and controllable segmentation for users. To further quantify the contribution of explicit scale information, we train a model without our scale embeddings. As shown in the last row of Table 3, even though the scale condition is not provided, our model still outperforms the one without scale embeddings. This comparison indicates that scale embeddings enhance the robustness of feature decoding, benefiting the training of our proposed decoder.

6. Conclusion

We present **S^{2}AM3D**, a scale-controllable framework for part segmentation of 3D point clouds. Through the design of a *point-consistent encoder* and a *scale-aware prompt decoder*, our method addresses the challenge of cross-view inconsistency, enhances global feature consistency, and enables real-time adjustment of segmentation granularity via continuous scale signals. The construction of a large-scale, high-quality part-level point cloud dataset provides crucial support for model training. Extensive experiments demonstrate that S^{2}AM3D achieves leading performance on multiple benchmarks, exhibiting robustness and controllability when handling complex structures and multi-scale parts, thereby providing a reliable solution for fine-grained 3D scene understanding and content editing. While the current framework primarily relies on point prompts and scale signals for interaction, future work will explore incorporating richer prompt modalities such as text instructions to support more intuitive semantic interaction.

Acknowledgements

This work was supported by the National Key R&D Program of China under Grant No. 2022YFA1004100 and the National Natural Science Foundation of China (NSFC) under Grant No. 62476067.

References

[1] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 3

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[3] Minghao Chen, Roman Shapovalov, Iro Laina, Tom Monnier, Jianyuan Wang, David Novotny, and Andrea Vedaldi. Partgen: Part-level 3d generation and reconstruction with multi-view diffusion models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5881–5892, 2025. 1

[4] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. *CVPR*, 2022. 2

[5] 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 annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 5, 6

[6] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 5, 6

[7] 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. 5

[8] Marco Garosi, Riccardo Tedoldi, Davide Boscaini, Massimiliano Mancini, Nicu Sebe, and Fabio Poiesi. 3d part segmentation via geometric aggregation of 2d visual features. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3257–3267. IEEE, 2025. 1

[9] Haoran Geng, Helin Xu, Chengyang Zhao, Chao Xu, Li Yi, Siyuan Huang, and He Wang. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7081–7091, 2023. 1

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

[11] Tianyu Huang, Yihan Zeng, Zhilu Zhang, Wan Xu, Hang Xu, Songcen Xu, Rynson WH Lau, and Wangmeng Zuo. Dreamcontrol: Control-based text-to-3d generation with 3d self-prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5364–5373, 2024. 1

[12] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020. 4

[13] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023. 1, 2, 3

[14] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10965–10975, 2022. 2

[15] Yangyan Li, Ren Bu, Mingchao Sun, Wei Wu, Wei Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *NeurIPS*, 2018. 1

[16] Yuchen Lin, Chenguo Lin, Panwang Pan, Honglei Yan, Yiqiang Feng, Yadong Mu, and Katerina Fragkiadaki. Partcrafter: Structured 3d mesh generation via compositional latent diffusion transformers. *arXiv preprint arXiv:2506.05573*, 2025. 5

[17] Jian Liu, Xiaoshui Huang, Tianyu Huang, Lu Chen, Yuenan Hou, Shixiang Tang, Ziwei Liu, Wanli Ouyang, Wangmeng Zuo, Junjun Jiang, et al. A comprehensive survey on 3d content generation. *arXiv preprint arXiv:2402.01166*, 2024. 1

[18] 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, 6, 7

[19] 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. *arXiv preprint arXiv:2504.11451*, 2025. 1, 2, 3, 5, 7

[20] Yujia Liu, Anton Obukhov, Jan Dirk Wegner, and Konrad Schindler. Point2cad: Reverse engineering cad models from 3d point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3763–3772, 2024. 1

[21] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. *Advances in neural information processing systems*, 32, 2019. 3

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[23] 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. 1, 2, 6, 7

[24] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7818–7827, 2025. 7

[25] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 2, 8

[26] 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. 4

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2, 5

[28] 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, 2

[29] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 2

[30] Haowen Sun, Yueqi Duan, Juncheng Yan, Yifan Liu, and Jiwen Lu. Mirageroom: 3d scene segmentation with 2d pre-trained models by mirage projection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20237–20246, 2024. 1

[31] George Tang, William Zhao, Logan Ford, David Benhaim, and Paul Zhang. Segment any mesh: Zero-shot mesh part segmentation via lifting segment anything 2 to 3d. *arXiv e-prints*, pages arXiv–2408, 2024. 7

[32] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *CVPR*, 2018. 1

[33] Mutian Xu, Xingyilang Yin, Lingteng Qiu, Yang Liu, Xin Tong, and Xiaoguang Han. Sampro3d: Locating sam prompts in 3d for zero-shot scene segmentation. *arXiv preprint arXiv:2311.17707*, 2023. 2

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

[35] 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, 6, 7

[36] 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. 2

[37] Haoze Zhang, Tianyu Huang, Zichen Wan, Xiaowei Jin, Hongzhi Zhang, Hui Li, and Wangmeng Zuo. Physchoreo: Physics-controllable video generation with part-aware semantic grounding, 2025. 1

[38] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

[39] Yuchen Zhou, Jiayuan Gu, Xuanlin Li, Minghua Liu, Yunhao Fang, and Hao Su. Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. *arXiv preprint arXiv:2312.03015*, 2023. 1

[40] 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. 1, 2, 6, 7