ZeroPS: High-quality Cross-modal Knowledge Transfer for Zero-Shot 3D Part Segmentation

Yuheng Xue[1] Nenglun Chen[1] Jun Liu[2] Wenyun Sun[1◇]
[1]Nanjing University of Information Science and Technology, China
[2]Lancaster University, UK
{yuhengxue,chennenglun,wenyunsun}@nuist.edu.cn {j.liu81}@lancaster.ac.uk

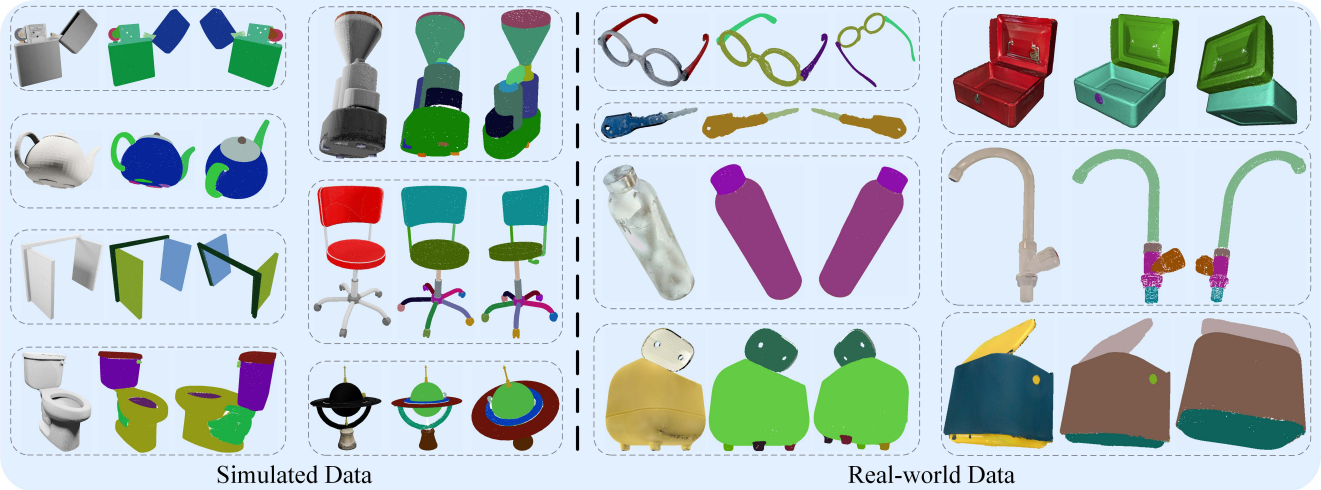

Figure 1. We propose ZeroPS, a novel zero-shot 3D part segmentation pipeline by leveraging pretrained foundation models, SAM [18] and GLIP [22], without training or fine-tuning. The figure shows our unlabeled segmentation results. Left: PartNetE's simulated data. Right: AKBSeg's real-world data. In each example, the input 3D object and output two visualizations are shown in turn. Please zoom in to see accurate 3D segmentation boundaries. ZeroPS also supports instance segmentation. Please refer to Fig. 5 for the qualitative comparison.

Abstract

Zero-shot 3D part segmentation is a challenging and fundamental task. In this work, we propose a novel pipeline, ZeroPS, which achieves high-quality knowledge transfer from 2D pretrained foundation models (FMs), SAM and GLIP, to 3D object point clouds. We aim to explore the natural relationship between multi-view correspondence and the FMs' prompt mechanism and build bridges on it. In ZeroPS, the relationship manifests as follows: 1) lifting 2D to 3D by leveraging co-viewed regions and SAM's prompt mechanism, 2) relating 1D classes to 3D parts by leveraging 2D-3D view projection and GLIP's prompt mechanism, and 3) enhancing prediction performance by leveraging multi-view observations. Extensive evaluations on the PartNetE and AKBSeg benchmarks demonstrate that ZeroPS significantly outperforms the SOTA method across zero-shot unlabeled and instance segmentation tasks. ZeroPS does not require additional training or fine-tuning for the FMs. ZeroPS applies to both simulated and real-world data. It is hardly affected by domain shift. The project page is available at https://luis2088.github.io/ZeroPS_page/.

◇: Corresponding author

1. Introduction

3D part segmentation is a crucial task in computer vision and computer graphics, leading to various applications such as robotics, shape editing, and AR/VR [2, 23, 27, 50, 65]. Due to the scarcity of 3D training data, recent research efforts focus on leveraging knowledge from foundation models (FMs) in other modalities (*e.g.*, text or images) to design a zero-shot manner, i.e., zero-shot 3D part segmentation. During inference, zero-shot 3D part segmentation requires not only making predictions on unseen data and classes but also ensuring accurate 3D segmentation. Though challenging, this task aligns with practical scenarios, like accurately segmenting a 3D object in an unfamiliar environment.

In this work, we propose a novel pipeline, ZeroPS,

which achieves high-quality knowledge transfer from 2D pretrained FMs, SAM [18] and GLIP [22], to 3D object point clouds. We aim to explore the natural relationship between multi-view correspondence and the FMs' prompt mechanism and build bridges on it. The following two subsections describe the manifestations of the relationship.

Intuitively, for a 3D object, we can obtain 2D groups by leveraging SAM to segment 2D images from different viewpoints. By back-projecting these groups into 3D and merging them, we can obtain 3D unlabeled parts. However, an important insight is that there exists a natural relationship between co-viewed regions and SAM's prompt mechanism. For any group in 3D, the visible portion of the group in adjacent viewpoints can be used as SAM's prompt to further extend it. By leveraging other viewpoints to continuously extend the 2D segmentation results, they will gradually become more complete in 3D. Therefore, as shown in Figs. 2 and 3, we design a component self-extension, which obtains 2D groups and extends each group from 2D to 3D. **Self-extension leverages the natural relationship between co-viewed regions and SAM's prompt mechanism to lift 2D to 3D in a training-free manner**.

To assign an instance label to each 3D unlabeled part, we integrate the GLIP model. As shown in Fig. 4, given a text prompt containing part classes, GLIP predicts many 2D bounding boxes. Since 2D boxes and 3D parts are not in the same space, we propose a two-dimensional checking mechanism (TDCM) to vote each 2D box to the best-matched 3D part, yielding a Vote Matrix. We select the highest vote in each column (part), thereby assigning an instance label to each 3D part. **TDCM leverages the natural relationship between 2D-3D view projection and GLIP's prompt mechanism (1D-2D) to relate 1D to 3D in a training-free manner.** To enhance the accuracy of label assignment for 3D parts, we propose a Class Non-highest Vote Penalty (CNVP) function to refine the Vote Matrix. Since GLIP inevitably produces incorrect predictions, the Vote Matrix exhibits certain unfairness (See Sec. 3.5 for details). An insight is that in each row (class) of the Vote Matrix, the highest vote represents GLIP's prediction for that class across as many views as possible, indicating most likely to be the correct prediction. CNVP penalizes other votes by using the highest vote in each row (class), yielding a refined version of the Vote Matrix. **CNVP leverages multi-view observations, which allows it to enhance prediction performance while retaining GLIP's zero-shot generalization in a training-free manner.**

In the experiment, we conduct extensive evaluations on the public PartNetE [28] benchmark. Our method outperforms the SOTA method by a large margin across unlabeled and instance segmentation tasks. Our method further narrows the gap between zero-shot and 3D fully supervised counterparts. Ablation studies demonstrate that both self-extension and CNVP contribute significantly to performance improvements. To better evaluate the generalization of all zero-shot methods concerning *unseen data, unseen classes, and hyperparameters*, we propose an AKBSeg benchmark from the existing AKB-48 [26] dataset. Retaining the default configurations from the evaluation on PartNetE, all zero-shot methods are re-evaluated on the AKBSeg benchmark. Our method continues to outperform the SOTA method by a large margin. Overall, the main contributions of our paper include:

- A novel zero-shot 3D part segmentation pipeline by leveraging pretrained foundation models, SAM and GLIP, without training or fine-tuning, is based only on multi-view correspondence and the foundation models' prompt mechanism, allowing it to demonstrate superior segmentation performance while preserving the zero-shot generalization from the pretrained foundation models.
- Three training-free manners: 1) self-extension lifts 2D to 3D by leveraging co-viewed regions and SAM's prompt mechanism; 2) TDCM relates 1D classes to 3D parts by leveraging 2D-3D view projection and GLIP's prompt mechanism; and 3) CNVP enhances prediction performance by leveraging multi-view observations.
- Our method achieves better zero-shot generalization and segmentation performance than the SOTA method.

2. Related Work

2.1. Supervised 3D Segmentation

Most methods [36–38, 49, 53, 58, 75] are fully supervised training on 3D datasets. These works focus on the design of network architectures to learn better 3D representations. The classical PointNet [36] considers the data structure of 3D points. Subsequent works [5, 20, 25, 30, 31, 56, 59, 60, 69, 71, 73] introduce ideas from the common deep learning field, such as transformer [52], unet [43], graph CNN [9, 17], rpn [41], etc. However, the 3D datasets [10, 33, 68] are several orders of magnitude smaller than the image datasets [44], but the complexity of 3D data is higher than images. Therefore, many works make up for the defects of insufficient 3D data through different training strategies, such as weak supervision [8, 19, 55, 64], self-supervision [11, 24, 35, 72] or few-shot learning [3, 14, 16, 45, 46, 54, 77]. In this work, we contrast zero-shot methods and 3D fully supervised counterparts on quantitative metrics.

2.2. 2D Foundation Models (FMs)

Recently, 2D FMs trained on large-scale datasets have demonstrated impressive zero-shot generalization. Another notable characteristic is the rich prompt mechanisms that enable these 2D FMs to establish connections across different modalities. Using free-form text prompts, CLIP [39] generates pixel-level predictions for a given image. A series of zero-shot 2D detectors, such as GLIP [22], GDINO

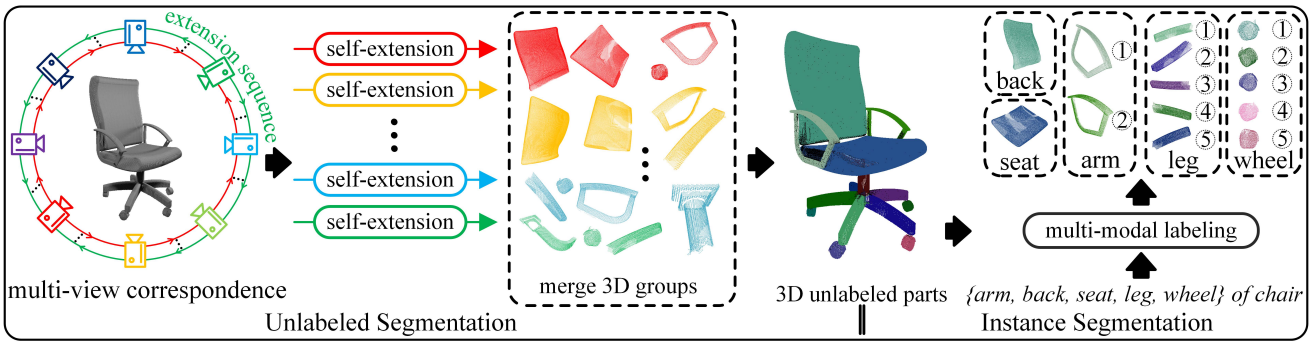

Figure 2. Overview of the proposed pipeline ZeroPS. First, in the unlabeled segmentation phase, the input 3D object is segmented into unlabeled parts. The self-extension (See Fig. 3) component can extend 2D segmentation from a single viewpoint to 3D segmentation (3D groups), by using a predefined extension sequence starting from that viewpoint. For example, the red cue on the left side of the figure illustrates this process. Second, in the instance segmentation phase, given a text prompt, the multi-modal labeling (See Fig. 4) component assigns an instance label to each 3D unlabeled part.

[29, 42], and Yolo-world [7], can output prediction bounding boxes for target objects based on given the text template prompt (e.g., 'arm, back, seat, wheel, leg of chair'). Segment Anything (SAM) [18] is an FM for zero-shot 2D segmentation. Given an image, SAM output instance masks at three different granularities (whole, part, and subpart) based on point or box prompts. Due to SAM's groundbreaking impact in 2D segmentation, many SAM-like models enhance various aspects of SAM, such as quality [6, 15] and efficiency [62, 74, 76]. For more works about FMs, please refer to this review [4]. This work leverages 1) the 2D instance-level segmentation capability and point prompt mechanism of a pretrained SAM model and 2) the 2D instance-level detection capability and text template prompt mechanism of a pretrained GLIP model.

2.3. Zero-shot 3D Part Segmentation

Except for: 1) the lack of large-scale 3D training data and 2) the robust zero-shot generalization of 2D FMs, another factor is 3) the 2D-3D mapping can be established through view projection and back-projection. These three factors drive recent research into exploring how to leverage 2D FMs to perform zero-shot 3D part segmentation.

Most existing works directly transfer knowledge from 2D FMs through multi-view 2D-3D mapping. PointCLIP V2 [78] proposes a realistic projection technique to enhance CLIP's visual encoder. It enables PointCLIP V2 to segment 3D sparse object point clouds. GeoZe [32] considers the intrinsic geometric information of 3D objects and proposes a training-free geometry-driven aggregation strategy. PartSLIP [28] uses the predicted bounding boxes from GLIP, to determine each initial superpoint's semantics. A well-designed grouping module partitions all semantic superpoints into 3D instance parts. Satr [1] also utilizes GLIP, but while PartSLIP focuses on point cloud segmentation, Satr is oriented toward mesh segmentation. Unlike these methods, PartDistill [51] introduces a bi-directional distillation framework, distilling 2D knowledge from CLIP or GLIP into a 3D student network, fully leveraging unlabeled 3D data for end-to-end training. However, limited by predefined output classes, the 3D network can predict unseen data but not unseen classes, with CLIP or GLIP's capability in this regard not retained in 3D. Our work explores new ideas for directly transferring knowledge.

The existing work mostly focuses on semantic segmentation [1, 32, 48, 51, 78], with only PartSLIP [28] capable of performing instance segmentation. To address the gap in instance segmentation methods, this work continues to explore new ideas for zero-shot instance segmentation.

Another parallel direction investigates zero-shot methods for scene segmentation [12, 34, 47, 63, 66, 67, 70].

3. Proposed Method: ZeroPS

3.1. Overview

Given a 3D object point cloud, this work aims to utilize SAM and GLIP to perform two types of segmentation: unlabeled and instance segmentation. The overall pipeline (See Fig. 2) is divided into two phases. In the unlabeled part segmentation phase, we first define the following operators by multi-view correspondence (See Sec. 3.2): 1) obtaining the extension sequence S_i starting from any viewpoint V_i; 2) calculating the forward- and back-projection between any viewpoint V_i and 3D space. Then, each self-extension (See Sec. 3.3) component inputs an extension sequence S_i and outputs 3D groups. Next, we merge all 3D groups (See Sec. 3.4) by a merging algorithm and get 3D unlabeled parts. In the instance segmentation phase, the multi-model labeling (See Sec. 3.5) component assigns an instance label to each 3D unlabeled part based on a text prompt.

3.2. Multi-view Correspondence

In 3D space, given an object point cloud $Q^{3D} \in \mathbb{R}^{N\times 6}$ as input, where N represents the number of points, each point includes a position $\{x, y, z\}$ and color $\{r, g, b\}$. We arrange K viewpoints relatively uniformly around Q^{3D}. It can be

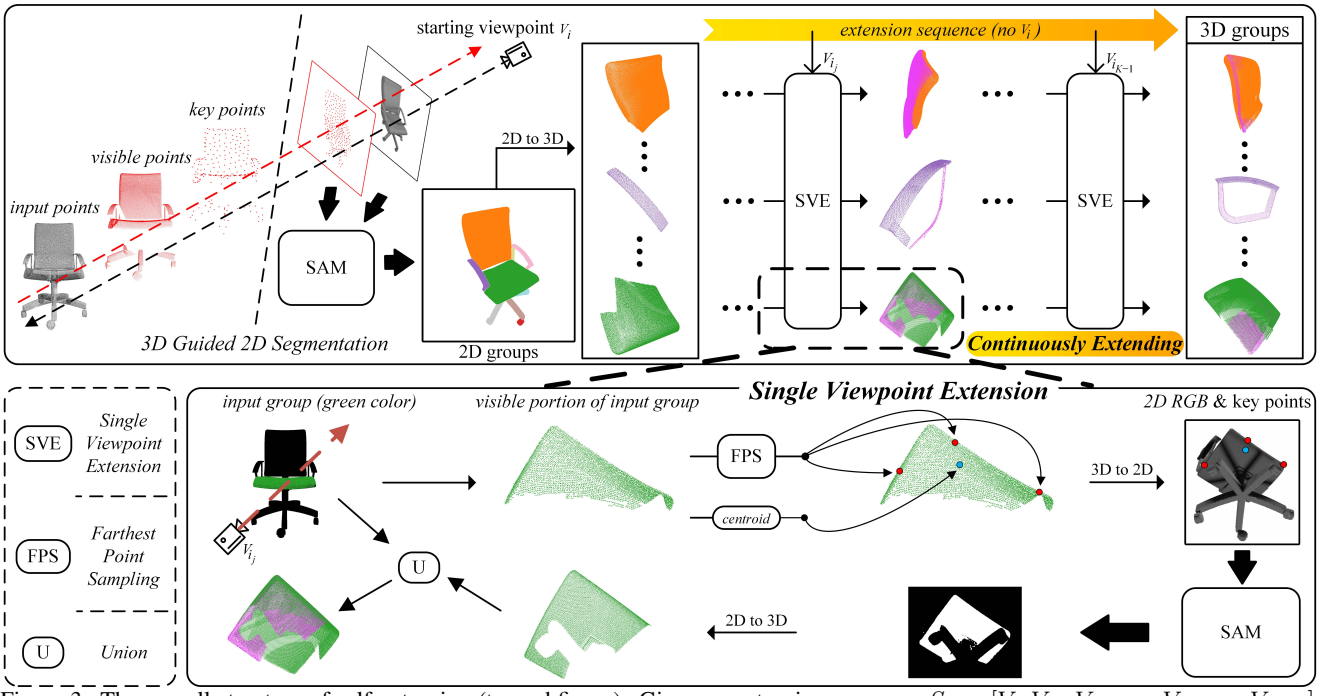

Figure 3. The overall structure of self-extension (top subfigure). Given an extension sequence $S_i = [V_i, V_{i_1}, V_{i_2}, \ldots, V_{i_j}, \ldots, V_{i_{k-1}}]$, self-extension aims to obtain 2D groups from starting viewpoint V_i and extends each group from 2D to 3D by the remaining viewpoints. Specifically, for the starting viewpoint V_i, self-extension utilizes 3D key points to guide SAM to segment the 2D image. As the segmented 2D groups originate from 2D segmentation results, self-extension continuously extends these groups to 3D segmentation results (3D groups) by SVE (Single Viewpoint Extension). During continuously extending, the remaining viewpoints in S_i, $[V_{i_1}, V_{i_2}, \ldots, V_{i_j}, \ldots, V_{i_{k-1}}]$, are iterated. At each iteration, inputting the current viewpoint and each group, SVE extends each input group. As an example, a detailed process of how SVE extends a single group is provided in the bottom right subfigure.

referred to Table S1 in Supplementary. We use the notation V_i $(i = 1, 2, \ldots, K)$ to name each viewpoint V. Then, we perform point cloud rendering of Q^{3D} from each viewpoint V_i. The output of each V_i consists of a 2D RGB image denoted as I_i with shape $(H \times W \times 3)$ and a point cloud index matrix denoted as P_i with shape $(H \times W \times 1)$, where each element at matching locations in I_i and P_i respectively represent the 3D position and color of the same point. Now, for any point of Q^{3D} in 3D space, we can easily find its position in the pixel coordinate system of V_i, or vice versa.

Extension Sequence. We construct an undirected, unweighted graph using all K viewpoints surrounding the 3D object Q^{3D}. In this graph, each node represents a viewpoint, and edges are established between adjacent viewpoints based on their spatial arrangement. Starting from any viewpoint V_i, we can perform a breadth-first search algorithm on the graph, resulting in a sequence referred to as the extension sequence, denoted as $S_i = [V_i, V_{i_1}, V_{i_2}, \ldots, V_{i_j}, \ldots, V_{i_{K-1}}]$. The extension sequence ensures that at any iteration, there is a co-viewed region between the current viewpoint and the previously iterated viewpoints. This allows, during continuously extending, Single Viewpoint Extension (SVE) to obtain the visible portion of the input group from the input viewpoint.

Bi-directional Projection (BiP). To facilitate the mapping of a subset of Q^{3D} between 2D (any V_i) and 3D space, we denote forward- and back-projection simply as BiP as follows:

$$X^{3D} = BiP(X^{2D}, P_i), \tag{1}$$

$$X^{2D} = BiP(X^{3D}, P_i), \tag{2}$$

where X^{3D} indicates a subset of Q^{3D} and X^{2D} indicates the subset of 2D coordinates of the pixel coordinate system of V_i. More generally, BiP can also in parallel process multiple subsets in the same viewpoint.

3.3. Self-extension

Self-extension aims to obtain 2D groups and extends each group from 2D to 3D. The overall structure of self-extension is illustrated in Fig. 3. Given an extension sequence $S_i = [V_i, V_{i_1}, V_{i_2}, \ldots, V_{i_j}, \ldots, V_{i_{k-1}}]$, for the starting viewpoint V_i, self-extension utilizes 3D key points to guide SAM to segment the 2D image. As the segmented 2D groups originate from 2D segmentation results, self-extension continuously extends these groups to 3D segmentation by SVE.

3D Guided 2D Segmentation. As shown in the top subfigure of Fig. 3, to obtain 2D groups, all key points and 2D RGB image I_i in V_i are fed into SAM. An automatic segmentation setting is then performed. The overall process

can be formulated as follows:

$$KPoints_{V_i}^{2D} = BiP(FPS(Q_{V_i}^{3D}), P_i), \quad (3)$$

$$\{G_1^{2D}, \ldots, G_n^{2D}\} = SAM(I_i, KPoints_{V_i}^{2D}), \quad (4)$$

where $Q_{V_i}^{3D}$ indicates the visible points of Q^{3D} in V_i, $KPoints^{2D}$ indicates key points, $\{G_1^{2D}, \ldots, G_n^{2D}\}$ indicates n 2D groups, and FPS indicates Farthest Point Sampling. These 2D groups are back-projected into 3D space:

$$\{G_1^{3D}, \ldots, G_n^{3D}\} = BiP(\{G_1^{2D}, \ldots, G_n^{2D}\}, P_i). \quad (5)$$

Single Viewpoint Extension (SVE). Before continuously extending, a Single Viewpoint Extension (SVE) operator needs to be defined. Given a viewpoint, SVE can extend the input group. For a group in 3D, we observe a natural relationship between the co-viewed region and SAM's prompt mechanism. As shown in the bottom right subfigure of Fig. 3, from V_{i_j}, 'we observe' a portion of the input group (green color), as there is a co-viewed region between V_{i_j} and $\{V_{i_1}, V_{i_2}, \ldots, V_{i_{j-1}}\}$ (See Sec. 3.2, 'Extension Sequence'). SVE obtains the visible portion of the input group from V_{i_j}. Then, SVE feeds both I_{i_j} as 2D RGB and the key points as prompt into SAM and performs inference. Finally, SVE obtains the union of the mask and the input group. The input group is extended to more points with the same semantics. The overall process can be formulated as:

$$KPoints_{V_{i_j}}^{2D} = BiP(FPS(G_{V_{i_j}}^{3D}) \cup CC(G_{V_{i_j}}^{3D}), P_{i_j}), \quad (6)$$

$$Mask = SAM(I_{i_j}, KPoints_{V_{i_j}}^{2D}), \quad (7)$$

$$G^{3D} \leftarrow G^{3D} \cup BiP(Mask, P_{i_j}), \quad (8)$$

where $G_{V_{i_j}}^{3D}$ indicates the visible portion of the input group from V_{i_j}, $KPoints_{V_{i_j}}^{2D}$ indicates key points in V_{i_j}, $\leftarrow$ indicates set extension and CC indicates the calculation of the point closest to the centroid. We propose the Single Viewpoint Extension (SVE) with input a G^{3D} and V:

$$G^{3D} \leftarrow SVE(G^{3D}, V), \quad (9)$$

where SVE is utilized to extend the G^{3D} from V. For SVE's input, G^{3D} needs to be within the visual range of viewpoint V. Otherwise, it will not be extended (remaining unchanged). More generally, SVE can in parallel extend a set of groups in the same viewpoint.

Continuously Extending. To continuously extend each group of Eq. (5), we iterate over the remaining viewpoints of S_i, $[V_{i_1}, V_{i_2}, \ldots, V_{i_j}, \ldots, V_{i_{K-1}}]$, by SVE:

$$\begin{aligned}
&\{G_1^{3D}, \ldots, G_n^{3D}\} \leftarrow SVE(\{G_1^{3D}, \ldots, G_n^{3D}\}, V_{i_1}) \\
&\{G_1^{3D}, \ldots, G_n^{3D}\} \leftarrow SVE(\{G_1^{3D}, \ldots, G_n^{3D}\}, V_{i_2}) \\
&\qquad\qquad \vdots \\
&\{G_1^{3D}, \ldots, G_n^{3D}\} \leftarrow SVE(\{G_1^{3D}, \ldots, G_n^{3D}\}, V_{i_{K-1}}).
\end{aligned} \quad (10)$$

Finally, each group in $\{G_1^{3D}, \ldots, G_n^{3D}\}$ is extended from 2D (the starting viewpoint V_i) to 3D (all viewpoints). In summary, the self-extension can be represented as:

$$\{G_1^{3D}, \ldots, G_n^{3D}\} = SE(S_i), \quad (11)$$

where S_i indicates an extension sequence starting from V_i, SE indicates self-extension component and $\{G_1^{3D}, \ldots, G_n^{3D}\}$ indicates a set of 3D groups resulting from $SE(S_i)$ starting from V_i.

3.4. Merging 3D Groups

To get 3D unlabeled parts, a merging algorithm is employed to merge 3D groups, which are the output of all self-extensions (See Fig. 2). The algorithm depends on a merging threshold T. The pseudocode and detailed explanation are in the supplementary materials.

3.5. Multi-model Labeling

Multi-model labeling aims to assign an instance label to each 3D unlabeled part. The main idea is shown in Fig. 4. To get lots of 2D bounding boxes with instance labels, a text prompt containing part classes and K images (from all viewpoints) are fed into GLIP. Then, we vote each box to the best-matched 3D part and obtain a Vote Matrix that relates 1D classes (rows) to 3D parts (columns). Intuitively, we simply get the highest vote per column (part) and assign its class as a label to that part. However, we must face two problems: 1) How to vote each 2D bounding box to the best-matched 3D part, given that they are not in the same space; 2) How to enhance the accuracy of label assignment for 3D parts, since GLIP inevitably produces incorrect predictions.

Two-dimensional Checking Mechanism (TDCM). To vote each 2D predicted bounding box to the best-matched 3D part, we design a two-dimensional checking mechanism. Meanwhile, some unqualified boxes are discarded.

In detail, for any 2D predicted bounding box BB, we perform the Intersection over Union (IoU) between the F^{3D} and each 3D part P^{3D} in $C = \{P_1^{3D}, P_2^{3D}, \ldots, P_{m_2}^{3D}\}$. Further, we let P_s^{3D}, with the Maximum IoU, be the best-matched 3D part in 3D space:

$$P_s^{3D} = \arg\max_{P^{3D} \in C} \frac{|F^{3D} \cap P^{3D}|}{|F^{3D} \cup P^{3D}|}, \quad (12)$$

where the F^{3D} indicates the 3D visible points inside the BB, and the C indicates m_2 unlabeled parts P^{3D}. Meanwhile, we perform the IoU between the BB and each P^{box} in $C' = \{P_1^{box}, P_2^{box}, \ldots, P_{m_2}^{box}\}$. Then we let P_t^{box}, with the Maximum IoU, be the best-matched 3D part in 2D space:

$$P_t^{box} = \arg\max_{P^{box} \in C'} \frac{|BB \cap P^{box}|}{|BB \cup P^{box}|}, \quad (13)$$

where the C' indicates m_2 2D bounding box P^{box} of all 3D parts in the viewpoint where the BB is located. Note that

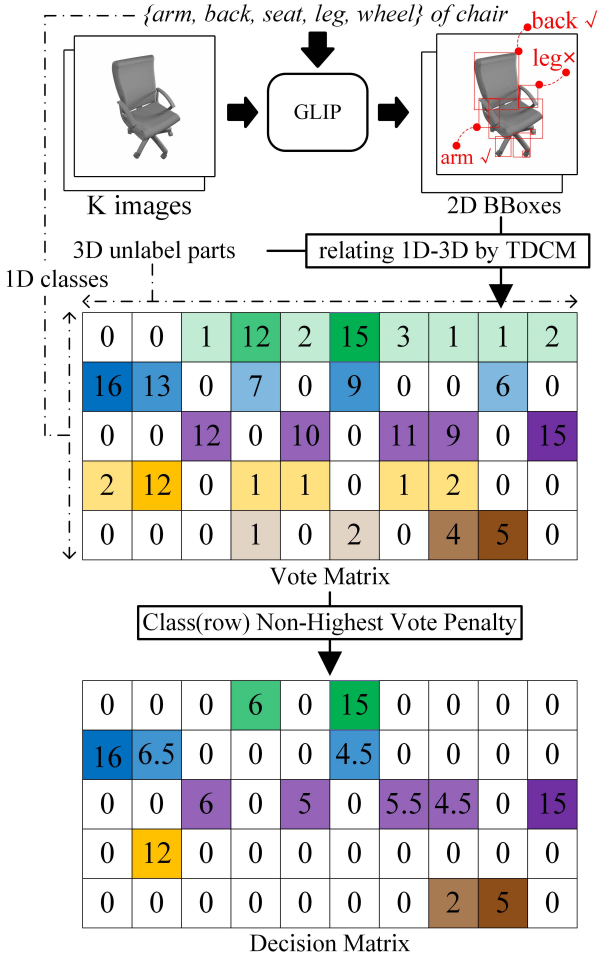

Figure 4. The overall structure of multi-modal labeling.

P_u in the C and C' denotes two states of the same 3D part, 3D point set and 2D bounding box, respectively. Finally, if $s = t$, the BB is voted to P_s^{3D}. Otherwise, the BB is discarded. In other words, it must guarantee that the best-matched part of the predicted bounding box in both 2D and 3D space is the same 3D part. Overall, TDCM leverages the natural relationship between 2D-3D view projection and GLIP's prompt mechanism (1D-2D) to relate 1D classes to 3D parts into a Vote Matrix.

Class Non-highest Vote Penalty (CNVP). To enhance the accuracy of label assignment for 3D parts, we propose a Class Non-highest Vote Penalty function.

In fact, the 2D predicted bounding boxes produced by GLIP inevitably have incorrect labels (See top right of Fig. 4). When the highest vote per column (part) is directly obtained and its class assigned as a label to that 3D part, this leads to two kinds of unfairness: 1) For a specific column (part), the highest vote 'wins' by only a narrow margin compared to other votes. For example, in the second column of the Vote Matrix in Fig. 4, '13' wins over '12' by just one vote; 2) For different columns (parts), the gap between the highest votes is too large when their highest votes are in the same row (class). For example, compared to the highest vote '16', in the first column (part) of the Vote Matrix, the election of '6' in the penultimate column (part) is unreasonable. In this case, in the penultimate column (part), '5' is more trustworthy than '6', because '5' possesses the highest vote in the final row (class), while '6' does not even reach half of the highest vote in the second row (class).

The unfairness in the Vote Matrix mentioned above needs improvement. In each row (class), the highest vote represents GLIP's prediction for that class across as many views as possible. This indicates that, compared to other votes, the highest vote is more likely to be the correct prediction, making it a reliable pivot in the Vote Matrix. Therefore, we use the highest vote per row (class) to penalize other votes through CNVP:

$$\begin{cases} \alpha, & \text{if } \alpha/\alpha_{\text{rm}} = 1 \\ \alpha/2, & \text{if } 0.5 \leq \alpha/\alpha_{\text{rm}} < 1 \\ 0, & \text{if } 0 \leq \alpha/\alpha_{\text{rm}} < 0.5 \end{cases}, \tag{14}$$

where α indicates each element of the Vote Matrix, α_{rm} indicates the maximum value within the same row where α is located. CNVP results in a Decision Matrix, a refined version of the Vote Matrix. It mitigates the incorrect predictions generated by GLIP. Overall, CNVP leverages multi-view observations, which allows it to enhance prediction performance while retaining GLIP's zero-shot generalization.

4. Experiments

4.1. Benchmark and Metric

PartNetE. For PartNetE [28], the training data are 28,367 3D objects from PartNet [33] and 45×8 3D objects from PartNet-Mobility [61], and the testing data are 1906 3D objects from PartNet-Mobility covering 45 object categories. PartNetE encompasses both common coarse-grained (e.g., chair seat) and fine-grained (e.g., knob) parts. This diversity of levels of granularity presents a significant challenge for the evaluated method.

AKBSeg. To better evaluate the generalization of all zero-shot baselines, we propose an AKBSeg benchmark. It collects 508 3D objects from the AKB-48 [26] dataset covering 16 object categories for testing data. Based on the original semantic annotations, we provide additional instance labels. Building upon PartNetE's simulated data, incorporating AKBSeg's real-world data into the experiment further enhances the zero-shot baseline's evaluative credibility. This also benefits the evaluation of future work in zero-shot 3D part segmentation.

Metric. We follow [57] to utilize the Average IoU as the unlabeled segmentation metric. We use the mask of instance label as ground truth for evaluating the unlabeled segmentation. We follow [28] to utilize the category mAP

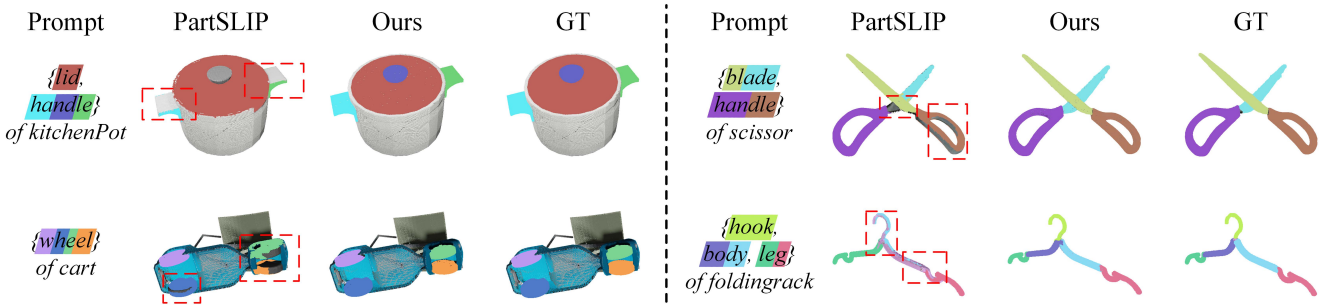

Figure 5. Qualitative comparison on zero-shot instance segmentation (zoom in for details). Left: PartNetE's simulated data. Right: AKBSeg's real-world data. The red dashed boxes indicate that our method produces more accurate 3D segmentation boundaries compared to the SOTA method, PartSLIP.

Table 1. Zero-shot unlabeled segmentation results on the PartNetE benchmark. Object category Average IoUs(%) are shown.

Method	Overall (45)	Bottle	Door	Lamp	Scissors	Table	Box	Kettle	KitchenPot	Lighter	Pliers	Stapler	Toilet
PartSLIP [28]	36.4	78.0	27.9	47.6	47.2	46.9	38.8	66.2	60.5	53.2	3.5	27.3	45.4
Ours (w/o Extending)	45.6	55.7	23.9	42.8	39.9	47.9	51.6	57.8	70.7	47.3	40.1	39.7	56.0
Ours	**56.0**	**80.4**	**37.8**	**72.9**	**51.1**	**53.3**	**63.1**	**85.5**	**80.3**	**64.4**	**61.3**	**80.7**	**58.2**

Please refer to the supplementary material for the full table. This also applies to Tabs. 2 to 4.

Table 2. Instance segmentation results on the PartNetE benchmark. Object category mAP50s(%) are shown.

Method	Overall (45)	Bottle	Door	Lamp	Scissors	Table	Box	Kettle	KitchenPot	Lighter	Pliers	Stapler	Toilet
PointGroup* [13]	31.0	38.2	**23.4**	62.7	38.5	**46.3**	7.2	61.3	**59.5**	33.6	28.2	88.3	2.2
SoftGroup* [53]	**31.9**	**43.9**	21.2	**63.3**	**39.3**	46.2	**8.6**	**63.8**	59.3	**34.6**	**40.4**	**94.3**	**2.4**
PartSLIP† [28]	23.3	67.0	10.6	27.8	18.2	28.6	18.9	26.8	58.9	15.1	1.0	16.2	12.9
Ours (w/o CNVP)†	24.1	62.0	14.8	30.9	26.1	26.6	26.0	22.0	54.4	16.3	38.6	22.3	14.3
Ours†	**28.5**	**74.5**	**15.7**	**35.9**	**26.4**	**29.4**	**32.2**	**33.4**	**64.4**	**21.1**	**40.7**	**44.9**	**15.5**

* fully supervised; † zero-shot; PartSLIP's overall result reproduces by the official code, with the official paper being 18.0% mAP50.

Table 3. Zero-shot unlabeled segmentation results on the AKBSeg benchmark. Object category Average IoUs(%) are shown.

Method	Overall (16)	Ballpoint	Bottle	Box	Bucket	Condiment	Drink	Faucet	Foldingrack	Knife	Sauce	Shampoo	Trashcan
PartSLIP [28]	34.3	3.0	8.7	35.0	49.9	44.4	10.5	11.0	32.4	73.0	22.5	37.6	66.0
Ours (w/o Extending)	49.3	33.7	55.7	**54.1**	74.1	50.7	49.9	44.6	50.6	74.8	35.7	56.5	72.0
Ours	**58.9**	**48.9**	**65.7**	52.5	**75.5**	**65.2**	**67.8**	**52.8**	**64.1**	**84.1**	**45.8**	**63.2**	**79.8**

Table 4. Zero-shot instance segmentation results on the AKBSeg benchmark. Object category mAP50s(%) are shown.

Method	Overall (16)	Ballpoint	Bottle	Box	Bucket	Condiment	Drink	Faucet	Foldingrack	Knife	Sauce	Shampoo	Trashcan
PartSLIP [28]	15.0	1.0	1.1	12.9	34.8	11.4	1.0	3.4	16.7	51.0	5.1	2.8	9.3
Ours (w/o CNVP)	23.9	5.0	**26.0**	13.6	59.6	28.5	35.0	**4.0**	35.4	80.3	9.2	4.9	7.1
Ours	**26.5**	**6.5**	20.2	**16.3**	**77.8**	**41.0**	**36.8**	**4.0**	**36.2**	**80.9**	**10.1**	**6.6**	**10.0**

(50% IoU threshold) as the instance segmentation metric.

4.2. Implementation Details

For our method, the input is an RGB object point cloud. The point cloud rendering resolution of PyTorch3D [40] is set to 800×800. The number of viewpoints is set to 20. For details of the viewpoints, please refer to the supplementary materials. The number of FPS output points in Eq. (3) is set to 256. Note that the FPS here should be distinguished from that in Eq. (6). The merge threshold T is set to 0.3, and the ablation analysis is described in Sec. 4.4.

4.3. Comparison with Existing Methods

First, we conduct the quantitative evaluation on PartNetE with zero-shot methods and 3D fully supervised counterparts (See Tabs. 1 and 2). Second, to further investigate the impact of factors concerning *unseen data, unseen classes, and hyperparameters (which ablation studies cannot fully account for)*, we evaluate all zero-shot baselines on AKBSeg again (See Tabs. 3 and 4). *The same configuration is maintained for each baseline across PartNetE and AKBSeg.*

Zero-shot method. We compare our method with the existing SOTA zero-shot instance segmentation method, PartSLIP. Tabs. 1 and 3 show that, for zero-shot unlabeled segmentation, our method achieves 56.0% and 58.9% Average IoUs and outperforms PartSLIP by large margins. Tabs. 2 and 4 show that, for zero-shot instance segmentation, our method outperforms PartSLIP by 5.2% and 11.5% mAP50s, respectively. From PartNetE to AKBSeg, our method nearly maintains its performance, while PartSLIP declines significantly. It indicates that our method demonstrates superior robustness. As shown in Fig. 5, since PartSLIP is based on superpoints [21], which are similar to

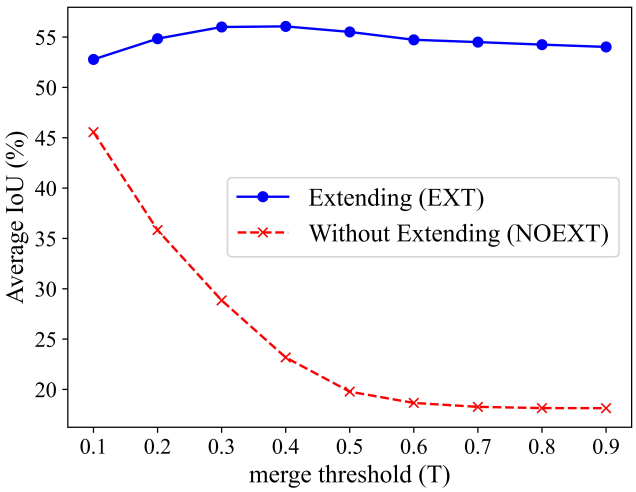

Figure 6. Ablation Study on self-extension by the 'Extending' and 'Without Extending' settings. The Average IoU is the overall result on PartNetE. See Sec. 4.4 for details.

superpixels in 2D, it is challenging to ensure accurate 3D segmentation boundaries. In contrast, our proposed self-extension lifts SAM's mask from 2D to 3D in a training-free manner, retaining zero-shot generalization while producing accurate 3D segmentation boundaries. Overall, our method significantly surpasses PartSLIP in both zero-shot generalization and segmentation performance.

3D fully supervised counterparts (methods). To observe the gap between zero-shot and 3D fully supervised (3D-FS) methods, we evaluate all baselines on the PartNetE. The zero-shot methods do not use any 3D training data, while the 3D-FS methods are trained on $45 \times 8 + 28k$ 3D objects. Since our method improves by 5.2% mAP50 compared to PartSLIP, it reduces the maximum gap between zero-shot and 3D-FS methods from 8.6% to 3.4% mAP50. Moreover, notably zero-shot methods possess a unique advantage in predicting unseen classes, such as directly evaluating on AKBSeg (See Tabs. 2 and 4), unlike 3D-FS methods limited to predefined fixed classes before training.

4.4. Ablation Study

Self-extension. To analyze the effectiveness of the proposed self-extension, we conduct the ablation study by two settings ('Extending' and 'Without Extending'). In the Extending (EXT) setting, we retain all self-extension processes. In the Without Extending (NOEXT) setting, we remove all steps involving continuously extending, namely all SVEs (See Fig. 3). In other words, we skip Eq. (10) in each self-extension. Through the results in Fig. 6, we observe the following: 1) NOEXT is overall lower than EXT. 2) NOEXT is overly dependent on the threshold T. Although fixing T to 0.1 yields relatively good performance, this approach's robustness and stability are suboptimal. 3) Compared to NOEXT, EXT demonstrates minimal sensitivity to changes in T, meanwhile consistently maintaining its superior performance. Overall, EXT demonstrates better robustness and stability compared to NOEXT. On the other hand, as shown in Tab. 3, EXT similarly exhibits a significant performance improvement on the AKBSeg. Moreover, Fig. 7 also shows that EXT produces better 3D consistency than NOEXT. This shows that self-extension effectively leverages the natural relationship between co-viewed regions and SAM's prompt mechanism to lift 2D to 3D. Since EXT is almost independent of T, we set T to 0.3 in Sec. 4.2.

Table 5. Ablation study on the number of viewpoints. The Average IoU is the overall result on PartNetE.

viewpoints	20	8	4
Average IoU	56.0	50.2	36.2

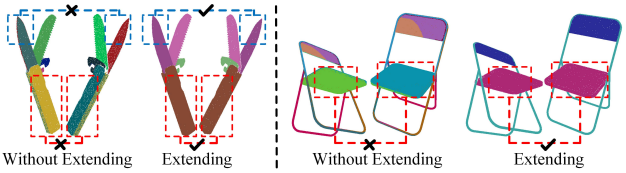

Figure 7. Qualitative results on self-extension (zoom in for details). See Sec. 4.4 for details.

Class Non-highest Vote Penalty (CNVP). We conduct the ablation study on the proposed CNVP. As shown in Tabs. 2 and 4, CNVP consistently maintains performance gains across the PartNetE and AKBSeg benchmarks, with improvements of 4.4% and 2.6% mAP50s, respectively. This indicates that CNVP effectively utilizes multi-view observations, enhancing performance while retaining GLIP's zero-shot generalization.

Number of Viewpoints. We conduct the ablation study on the number of viewpoints. As shown in Tab. 5, when the number drops to 8, the high performance is still maintained. Since it is difficult for 4 viewpoints to cover the entire 3D object uniformly, the performance drops substantially.

5. Conclusion

In this work, we propose a novel zero-shot 3D part segmentation pipeline. We explore the natural relationship between multi-view correspondence and the FMs' prompt mechanisms. The relationship manifests in the pipeline as self-extension, TDCM, and CNVP. Through extensive qualitative and quantitative comparison and ablation studies, our method demonstrates superior zero-shot generalization and segmentation performance than the SOTA method.

Acknowledgements. We thank Minghua Liu for providing the code and participating in discussions. This work was supported by National Natural Science Foundation of China under Grant No. 61702340; and in part by the Start-up Foundation for Introducing Talent of Nanjing University of Information Science & Technology (NUIST) under Grant 2023r063.

References

[1] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. Satr: Zero-shot semantic segmentation of 3d shapes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15166–15179, 2023. 3

[2] Jacopo Aleotti and Stefano Caselli. A 3d shape segmentation approach for robot grasping by parts. *Robotics and Autonomous Systems*, 60(3):358–366, 2012. 1

[3] Zhaochong An, Guolei Sun, Yun Liu, Fayao Liu, Zongwei Wu, Dan Wang, Luc Van Gool, and Serge Belongie. Rethinking few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3996–4006, 2024. 2

[4] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*, 2023. 3

[5] Haozhi Cao, Yuecong Xu, Jianfei Yang, Pengyu Yin, Shenghai Yuan, and Lihua Xie. Mopa: Multi-modal prior aided domain adaptation for 3d semantic segmentation. *arXiv preprint arXiv:2309.11839*, 2023. 2

[6] Wei-Ting Chen, Yu-Jiet Vong, Sy-Yen Kuo, Sizhou Ma, and Jian Wang. Robustsam: Segment anything robustly on degraded images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4081–4091, 2024. 3

[7] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16901–16911, 2024. 3

[8] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In *European Conference on Computer Vision*, pages 681–699. Springer, 2022. 2

[9] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29, 2016. 2

[10] Geng et al. Gapartnet: Cross-category domain-generalizable object perception and manipulation via generalizable and actionable parts. In *CVPR*, 2023. 2

[11] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, pages 473–491. Springer, 2020. 2

[12] 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, 2025. 3

[13] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[14] Amrin Kareem, Jean Lahoud, and Hisham Cholakkal. Paris3d: Reasoning-based 3d part segmentation using large multimodal model. In *European Conference on Computer Vision*, pages 466–482. Springer, 2025. 2

[15] 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, 2024. 3

[16] Hyunjin Kim and Minhyuk Sung. Partstad: 2d-to-3d part segmentation task adaptation. In *European Conference on Computer Vision*, pages 422–439. Springer, 2025. 2

[17] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. 2

[18] 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. *arXiv preprint arXiv:2304.02643*, 2023. 1, 2, 3

[19] Juil Koo, Ian Huang, Panos Achlioptas, Leonidas J Guibas, and Minhyuk Sung. Partglot: Learning shape part segmentation from language reference games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16505–16514, 2022. 2

[20] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8500–8509, 2022. 2

[21] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4558–4567, 2018. 7

[22] 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. 1, 2

[23] Connor Lin, Niloy Mitra, Gordon Wetzstein, Leonidas J Guibas, and Paul Guerrero. Neuform: Adaptive overfitting for neural shape editing. *Advances in Neural Information Processing Systems*, 35:15217–15229, 2022. 1

[24] Haotian Liu, Mu Cai, and Yong Jae Lee. Masked discrimination for self-supervised learning on point clouds. In *European Conference on Computer Vision*, pages 657–675. Springer, 2022. 2

[25] Jinxian Liu, Minghui Yu, Bingbing Ni, and Ye Chen. Self-prediction for joint instance and semantic segmentation of point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 187–204. Springer, 2020. 2

[26] Liu Liu, Wenqiang Xu, Haoyuan Fu, Sucheng Qian, Qiaojun Yu, Yang Han, and Cewu Lu. Akb-48: A real-world articulated object knowledge base. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14809–14818, 2022. 2, 6

[27] Minghua Liu, Xuanlin Li, Zhan Ling, Yangyan Li, and Hao Su. Frame mining: a free lunch for learning robotic manipulation from 3d point clouds. *arXiv preprint arXiv:2210.07442*, 2022. 1

[28] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21736–21746, 2023. 2, 3, 6, 7

[29] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 3

[30] Xueyi Liu, Xiaomeng Xu, Anyi Rao, Chuang Gan, and Li Yi. Autogpart: Intermediate supervision search for generalizable 3d part segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11624–11634, 2022. 2

[31] 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. 2

[32] Guofeng Mei, Luigi Riz, Yiming Wang, and Fabio Poiesi. Geometrically-driven aggregation for zero-shot 3d point cloud understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27896–27905, 2024. 3

[33] 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, 6

[34] Phuc Nguyen, Tuan Duc Ngo, Evangelos Kalogerakis, Chuang Gan, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4018–4028, 2024. 3

[35] Yatian Pang, Wenxiao Wang, Francis EH Tay, Wei Liu, Yonghong Tian, and Li Yuan. Masked autoencoders for point cloud self-supervised learning. In *European conference on computer vision*, pages 604–621. Springer, 2022. 2

[36] 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. 2

[37] 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.

[38] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Advances in Neural Information Processing Systems*, 35:23192–23204, 2022. 2

[39] 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

[40] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 7

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[42] Tianhe Ren, Qing Jiang, Shilong Liu, Zhaoyang Zeng, Wenlong Liu, Han Gao, Hongjie Huang, Zhengyu Ma, Xiaoke Jiang, Yihao Chen, et al. Grounding dino 1.5: Advance the" edge" of open-set object detection. *arXiv preprint arXiv:2405.10300*, 2024. 3

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015. 2

[45] Charu Sharma and Manohar Kaul. Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems*, 33:7212–7221, 2020. 2

[46] Gopal Sharma, Kangxue Yin, Subhransu Maji, Evangelos Kalogerakis, Or Litany, and Sanja Fidler. Mvdecor: Multi-view dense correspondence learning for fine-grained 3d segmentation. In *European Conference on Computer Vision*, pages 550–567. Springer, 2022. 2

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

[48] Anh Thai, Weiyao Wang, Hao Tang, Stefan Stojanov, James M Rehg, and Matt Feiszli. 3×2: 3d object part segmentation by 2d semantic correspondences. In *European Conference on Computer Vision*, pages 149–166. Springer, 2025. 3

[49] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J

Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019. 2

[50] Ardian Umam, Cheng-Kun Yang, Yung-Yu Chuang, Jen-Hui Chuang, and Yen-Yu Lin. Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In *European Conference on Computer Vision*, pages 596–611. Springer, 2022. 1

[51] Ardian Umam, Cheng-Kun Yang, Min-Hung Chen, Jen-Hui Chuang, and Yen-Yu Lin. Partdistill: 3d shape part segmentation by vision-language model distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3470–3479, 2024. 3

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[53] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 2, 7

[54] Lingjing Wang, Xiang Li, and Yi Fang. Few-shot learning of part-specific probability space for 3d shape segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4504–4513, 2020. 2

[55] Ruocheng Wang, Yunzhi Zhang, Jiayuan Mao, Ran Zhang, Chin-Yi Cheng, and Jiajun Wu. Ikea-manual: Seeing shape assembly step by step. *Advances in Neural Information Processing Systems*, 35:28428–28440, 2022. 2

[56] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018. 2

[57] Xiaogang Wang, Xun Sun, Xinyu Cao, Kai Xu, and Bin Zhou. Learning fine-grained segmentation of 3d shapes without part labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10276–10285, 2021. 6

[58] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2

[59] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems*, 35:33330–33342, 2022. 2

[60] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2

[61] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 6

[62] Yunyang Xiong, Bala Varadarajan, Lemeng Wu, Xiaoyu Xiang, Fanyi Xiao, Chenchen Zhu, Xiaoliang Dai, Dilin Wang, Fei Sun, Forrest Iandola, et al. Efficientsam: Leveraged masked image pretraining for efficient segment anything. *arXiv preprint arXiv:2312.00863*, 2023. 3

[63] 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. 3

[64] Xun Xu and Gim Hee Lee. Weakly supervised semantic point cloud segmentation: Towards 10x fewer labels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13706–13715, 2020. 2

[65] Xianghao Xu, Yifan Ruan, Srinath Sridhar, and Daniel Ritchie. Unsupervised kinematic motion detection for part-segmented 3d shape collections. In *ACM SIGGRAPH 2022 Conference Proceedings*, pages 1–9, 2022. 1

[66] Mi Yan, Jiazhao Zhang, Yan Zhu, and He Wang. Maskclustering: View consensus based mask graph clustering for open-vocabulary 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28274–28284, 2024. 3

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

[68] 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

[69] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 2

[70] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3292–3302, 2024. 3

[71] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9491–9500, 2019. 2

[72] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19313–19322, 2022. 2

[73] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8883–8892, 2021. 2

[74] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong.

Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 3

[75] 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

[76] Xu Zhao, Wenchao Ding, Yongqi An, Yinglong Du, Tao Yu, Min Li, Ming Tang, and Jinqiao Wang. Fast segment anything. *arXiv preprint arXiv:2306.12156*, 2023. 3

[77] 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. 2

[78] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2639–2650, 2023. 3