PartSLIP++: Enhancing Low-Shot 3D Part Segmentation via Multi-View Instance Segmentation and Maximum Likelihood Estimation

Yuchen Zhou* Jiayuan Gu* Xuanlin Li Minghua Liu Yunhao Fang Hao Su
UC San Diego

Abstract

Open-world 3D part segmentation is pivotal in diverse applications such as robotics and AR/VR. Traditional supervised methods often grapple with limited 3D data availability and struggle to generalize to unseen object categories and in the wild scenarios. PartSLIP, a recent advancement, has made significant strides in zero- and few-shot 3D part segmentation. This is achieved by harnessing the capabilities of the 2D open-vocabulary detection module, GLIP, and introducing a heuristic method for converting and lifting multi-view 2D bounding box predictions into 3D segmentation masks. In this paper, we introduce PartSLIP++, an enhanced version designed to overcome the limitations of its predecessor. Our approach incorporates two major improvements. First, we utilize a pre-trained 2D segmentation model, SAM, to produce pixel-wise 2D segmentations, yielding more precise and accurate annotations than the 2D bounding boxes used in PartSLIP. Second, PartSLIP++ replaces the heuristic 3D conversion process with an innovative modified Expectation-Maximization algorithm. This algorithm conceptualizes 3D instance segmentation as unobserved latent variables, and then iteratively refines them through an alternating process of 2D-3D matching and optimization with gradient descent. Through extensive evaluations, we show that PartSLIP++ demonstrates better performance over PartSLIP in both low-shot 3D semantic and instance-based object part segmentation tasks. We finally showcase the versatility of PartSLIP++ in enabling applications like semi-automatic part annotation and 3D Instance Proposal Generation.

1. Introduction

3D part segmentation focuses on dividing a 3D shape into distinct parts, which necessitates a comprehensive understanding of the object's structure, semantics, mobility, and functionality. It plays a crucial role in various applications, including robotics, AR/VR, and shape analysis and synthesis [2, 20, 26, 43].

Remarkable progress has been made in developing diverse data-driven approaches for 3D part segmentation [22, 32, 42, 45]. However, standard supervised training necessitates a substantial volume of finely-annotated 3D training shapes, the collection and annotation of which are typically labor-intensive and time-consuming. For instance, the PartNet dataset [27], which is the most extensive publicly available 3D part dataset, comprises 26,000 objects but covers only 24 common everyday categories. Such limited training categories often hinders supervised methods from effectively tackling open-world scenarios and handling out-of-distribution test shapes (e.g., unseen classes).

Contrary to 3D data, 2D images accompanied by text descriptions are more readily available, contributing significantly to the recent advancements in large-scale image-language models [1, 13, 18, 33–35, 48]. A recent work, PartSLIP [21], thus capitalizes on this by utilizing the rich 2D priors and the robust zero-shot capabilities of the image-language model to address the 3D part segmentation task in a zero or few-shot fashion. PartSLIP begins by rendering multi-view images for an input 3D point cloud. These images, along with a text prompt, are fed into the GLIP [18] model, known for its proficiency in open-world 2D detection. To translate the 2D bounding boxes detected by GLIP into 3D semantic and instance segmentation masks, PartSLIP introduces a heuristic pipeline involving superpoint generation, 3D voting, and 3D grouping. While PartSLIP has shown impressive zero-shot and few-shot performance, it does have some notable drawbacks: **(a)** the 2D bounding boxes generated by GLIP can be coarse, lacking pixel-level accurate part annotations; **(b)** the heuristic pipeline might not yield the most accurate 3D segmentation; **(c)** the heuristic relies on multiple hyperparameters, making the final results sensitive to their specific settings.

In this work, we propose **PartSLIP++**, a novel method designed to surpass the aforementioned limitations and further enhance its performance. This method primarily incorporates two significant modifications. Firstly, we generate pixel-wise 2D annotations by utilizing a pre-trained 2D

* Equal contributions; Corresponding Authors: yuz256@ucsd.edu, jigu@ucsd.edu

segmentation model, SAM [16]. Specifically, SAM uses initially-detected bounding boxes from GLIP as prompts to generate precise 2D instance segmentations. These pixel-wise segmentation masks offer more accurate 2D annotations compared to the bounding boxes used in the prior work, PartSLIP. Secondly, rather than relying on a heuristic lifting algorithm in PartSLIP, we formulate the conversion from multi-view 2D segmentation to 3D segmentation as a problem of maximum likelihood estimation with latent variables. To address this, we introduce a modified EM algorithm [7]. Here, the 3D instance segmentation mask is treated as an unobserved latent variable. During the E-step, the Hungarian algorithm is applied to match the predicted 2D instance segmentation masks with the current estimate of projected 3D instance segmentation masks, aiming to calculate the expectation of the log-likelihood. In the M-step, the 3D instance segmentation is updated by minimizing a cost function based on the matches established in the E-step. This algorithm iteratively alternates between these two steps until convergence is reached.

In our comprehensive evaluation using the PartNetE dataset [21], we demonstrate that PartNet++ outperforms PartSLIP in terms of both low-shot 3D semantic and instance segmentation tasks. Additionally, our detailed ablation studies highlight the effectiveness of each module and design technique we propose. Key contributions of our work include:

- Integrating a pre-trained 2D segmentation model into the PartSLIP pipeline, yielding more accurate and precise 2D pixel-wise part annotations than the bounding boxes used in prior work.
- Reformulating the problem of lifting multi-view 2D part segmentation masks to 3D masks as a maximum likelihood estimation problem, and introducing a novel modified Expectation-Maximization (EM) algorithm for effective optimization of this problem.
- Demonstrating that PartSLIP++ outperforms existing low-shot baselines in both 3D semantic and instance-based part segmentation through quantitative and qualitative analysis. The effectiveness of PartSLIP++ further enables applications like semi-automatic part annotation and 3D Instance Proposal Generation.

2. Related Works

2.1. 3D Part Segmentation

There are two main tasks for 3D segmentation: semantic and instance segmentation. Semantic segmentation is to predict a semantic label for each geometric primitive (e.g., point [30], voxel [9], superpoints [17]). For learning-based instance segmentation, there are mainly two lines of works: bottom-up and top-down approaches. The bottom-up approaches [5, 10, 14, 19, 38–40, 47] usually learn instance-aware features and cluster geometric primitives into different instances based on the distance metric defined on those features. The top-down approaches [11, 44, 45] usually first generate region proposals and then segment the foreground within each region of interest. Recently, transformers [37] are also introduced for 3D instance segmentation [23, 36]. Each object instance is represented as an instance query, and a transformer decoder is applied to predict instance masks.

Most works above address scene-level 3D semantic segmentation and object-level 3D instance segmentation. Part-level 3D segmentation [3, 24, 28, 41, 46] has its unique challenges. For example, part instances are closer to each other and smaller than object instances. Besides, some part instances can be encompassed by other objects (e.g., a handle in the door). [46] proposes a method that predicts a fixed number of part instance masks given a point cloud. During training, it uses the Hungarian algorithm to match each predicted instance mask with a ground-truth instance mask for supervision.

2.2. Multi-view 2D-3D Segmentation

Many works have studied how to tackle 3D understanding problems by multi-view approaches, e.g., shape classification [29] and semantic segmentation [6, 12, 25]. Given recent progress in 2D foundation models, several works have explored how to transfer the knowledge of 2D foundation models to 3D in a multi-view fashion. PointCLIP [49] enables low-shot shape classification by aggregating the view-wise features of rendered multi-view depth maps encoded by CLIP [33]. LeRF [15] distills CLIP features into a language embedded radiance field through NeRF-style optimization, which can support pixel-aligned, zero-shot queries. In addition, SA3D [4] generalizes a powerful vision foundation model SAM [16] to segment 3D objects also via NeRF-style optimization. Recently, PartSLIP [21] proposes a pipeline to tackle 3D part segmentation with the help of open-vocabulary 2D object detection models like GLIP [18], detailed in the next section.

3. Method

We first review the prior work PartSLIP in Sec. 3.1. We refer readers to the original paper for more details. Then, we revisit the multi-view 2D-3D segmentation pipeline in Sec. 3.2, and propose a straightforward but effective way to improve 2D segmentation results, which can be a bottleneck for multi-view approaches. Last, we propose a modified EM algorithm to merge multi-vew 2D segmentation results into 3D part labels in Sec. 3.3. Fig. 1 provides an overview of our improved pipeline PartSLIP++.

3.1. Preliminary: PartSLIP

[21] introduces a pipeline called PartSLIP, which leverages GLIP [18], a pretrained open-vocabulary object detection

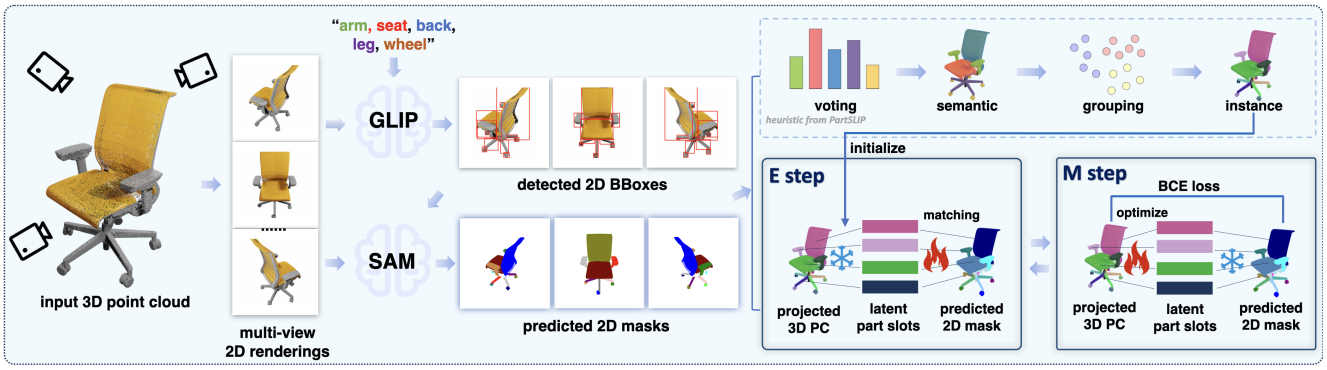

Figure 1. PartSLIP++ begins by taking a dense 3D point cloud as its input. It initially renders multi-view images from this point cloud. These images, along with a text prompt, are then input into the GLIP model, which predicts 2D bounding boxes. Subsequently, we utilize the SAM model to generate 2D instance segmentation masks for each view, using the predicted 2D bounding boxes as prompts. These multi-view 2D instance masks are converted into a 3D part segmentation mask using a novel, modified EM algorithm. During the E-step, the Hungarian algorithm is employed to find the optimal match between the projected 3D segmentation and the 2D predicted instance masks. In the M-step, the found matching is used to refine the 3D segmentation through gradient descent optimization. Lastly, the heuristic method presented by PartSLIP is applied to initialize the 3D instance segmentation.

model, to tackle both semantic and instance segmentation tasks for 3D object parts. Given a colored point cloud, PartSLIP first renders multiple images from K predefined camera poses. Then, each rendered image and a text prompt concatenating all part names of interest and the object category are fed into the GLIP model, which will predict multiple 2D bounding boxes for all part instances visible from the current view. Finally, all 2D bounding boxes from different views are merged into 3D part segmentation labels. [21] proposes a learning-free module to lift the 2D GLIP predictions to 3D part segmentation labels, which mainly contains 3 following components:

3D Superpoint Generation The input point cloud P is first oversegmented into a collection of superpoints [17] $\{SP_i\}$. Points in each superpoint share similar normals and colors, and are assumed to belong to the same instance. Part labels will be calculated based on superpoints instead of points, which can save much computation and lead to potentially better performance due to the 3D prior.

3D Semantic Voting The semantic label of each superpoint is voted by all 2D bounding boxes from multiple views that overlap with its 2D projection. Concretely, for a superpoint SP_i and a part category j, a score $s_{i,j}$ is calculated based on the ratio of visible points covered by 2D detected instances of the part category in each view:

$$s_{i,j} = \frac{\sum_k \sum_{p \in SP_i} [\text{VIS}_k(p)][\exists b \in \mathcal{B}_k^j : \text{INS}_b(p)]}{\sum_k \sum_{p \in SP_i} [\text{VIS}_k(p)]} \quad (1)$$

where $[\cdot]$ is the Iverson bracket (which evaluates to 1 if the predicate inside it is true, and 0 if false); $\text{VIS}_k(p)$ indicates whether the 3D point p is visible in view k; $\mathcal{B}_k^j$ is a set of predicted bounding boxes of category j in view k; and $\text{INS}_b(p)$ indicates whether the projection of point p in view k is inside the bounding box b. The part category with the highest score is assigned to the superpoint as its semantic label.

3D Instance Grouping PartSLIP [21] heuristically groups oversegmented superpoints into instances according to their semantic similarity, spatial adjacency and 2D label consistency across views. Specifically, two superpoints SP_u and SP_v are considered to belong to the same instance if (a) they share the same semantic label, (b) they are neighbors in a KNN graph, and (c) the overlaps between their 2D projections and detected 2D bounding boxes are similar in each view. The overlap between the 2D projection of a superpoint SP_u and a 2D bounding box $b \in \mathcal{B}_k$ in view k is:

$$o(SP_u, b) = \frac{\sum_{p \in SP_u} [\text{VIS}_k(p)][\text{INS}_b(p)]}{\sum_{p \in SP_u} [\text{VIS}_k(p)]} \quad (2)$$

[21] considers a list of 2D bounding boxes $\mathcal{B}'$ from views where both of superpoints SP_u and SP_v are visible, and constructs two feature vectors $I_u, I_v \in \mathbb{R}^{|\mathcal{B}'|}$, where $I_u[i] = o(SP_u, \mathcal{B}'[i])$. The last criterion is satisfied if $\frac{|I_u - I_v|_1}{max(|I_u|_1, |I_v|_1)}$ is smaller then a predefined threshold. 3D instances can be found via the Union-Find algorithm.

3.2. Revisiting Multi-view 2D-3D Segmentation

In this section, we will revisit how 3D segmentation is tackled by multi-view 2D segmentation. Given a colored point cloud P, the goal of 3D segmentation is to predict its label Y. For multi-view 2D-3D segmentation approaches, with K views rendered from the point cloud, a 2D instance segmentation model is first employed to generate instance segmentation masks $\mathcal{M}_k$ for each view k. The key is to merge

2D segmentation results from multiple views. This problem can be formulated as estimating the parameters Y by maximizing the likelihood of $P(\{\mathcal{M}_k\}|Y)$. In other words, we try to find a 3D label assignment that is compatible with observed 2D predictions. Intuitively, if two points belong to the same predicted 2D instance in each view, chances are that they belong to the same 3D instance.

Due to the lack of strong open-vocabulary instance segmentation models at that time, PartSLIP resorted to the open-vocabulary object detection model GLIP, and used bounding boxes as coarse instance masks. However, a bounding box can cover irrelevant pixels from other instances, resulting in noisy 2D instance labels. To address this issue, **we propose to convert GLIP to an open-vocabulary instance segmentation model by using a promptable 2D instance segmentation model to further segment instances within detected bounding boxes**. In this work, we use the Segment Anything Model (SAM)[16]. The predicate $\text{INS}_b(p)$ in Eq. 1 and 2, which indicates whether a point is inside a bounding box, can be replaced with $\text{INS}_M(p)$, where M is the instance mask output by SAM with the bounding box b as the prompt.

Besides, PartSLIP does not directly maximize the likelihood of predicted 3D part instances. As mentioned in Sec. 3.1, it uses the Union-Find algorithm to group superpoints into instances based on distances between heuristically-designed features. Such method can be sensitive to the threshold of feature distance to consider whether two superpoints can be merged. To this end, **given multi-view 2D instance segmentations and initial 3D instances produced by the PartSLIP pipeline, we further refine these 3D instances by proposing a modified expectation-maximization (EM) algorithm to find the maximum-likelihood estimates of 3D instances**, detailed in the next section.

3.3. 2D-3D Part Segmentation with EM Algorithm

3.3.1. Problem Definition

Formally, we define the problem of multi-view 2D-3D segmentation as estimating the label $Y \in \mathbb{L}^n$ of a colored point cloud $P \in \mathbb{R}^{n\times 3}$ by maximizing the likelihood of $P(\mathcal{M}|Y)$, where $\mathcal{M} = \cup_{k=1}^{K}\mathcal{M}_k$ is the union of all predicted 2D instance masks from all K views and $\mathcal{M}_k$ is the set of 2D instance masks in view k. Here, n is the number of points and $\mathbb{L}$ is a predefined set of labels. $\mathbb{L}$ is usually defined as a set of integers, the number of which is either the number of semantic categories for semantic segmentation, or the maximum number of instances for instance segmentation. We denote the number of labels by $l = |\mathbb{L}|$.

Without loss of generality, we take instance segmentation for example in this section. We introduce a parameter (3D instance labels) matrix $\Theta \in \mathbb{R}^{n\times l}$, where the i-th row $\Theta_{i,:}$ is the logit of the i-th point for 3D instance label and $Y_i = argmax_j(\Theta_{i,j})$. Besides, we introduce a latent (2D-3D assignment) matrix $Z \in \{0,1\}^{m\times l}$, where $m = |\mathcal{M}|$ is the total number of 2D predicted instances across views. $Z_{i,j} = 1$ and $Z_{i,\neq j} = 0$ indicate that the i-th 2D predicted instance should belong to the j-th 3D instance j. The maximum likelihood estimate (MLE) of the unknown parameters Θ is determined by maximizing the marginal likelihood of the observed data $\mathcal{M}$:

$$\begin{aligned} L(\Theta;\mathcal{M}) &= P(\mathcal{M}|\Theta) \\ &= \int_Z P(\mathcal{M},Z|\Theta) = \int_Z P(\mathcal{M}|Z,\Theta)P(Z|\Theta) \end{aligned} \tag{3}$$

To find the MLE of 3D instance label parameter Θ, we apply the classical expectation-maximization (EM) algorithm [7] with some modifications. The EM algorithm is an iterative method, consisting of two steps at each EM iteration. An EM iteration alternates between performing an expectation (E) step to build a log likelihood function of parameters using the current estimate, and a maximization (M) step to find the parameters that maximize the likelihood function built in the E step. In this work, we randomly select a view to perform updates at each EM iteration. In the E step (Sec. 3.3.2), we define a cost function (equivalent to a log likelihood function) to match each 2D predicted instance in the selected view with one of 3D instance labels, and update the latent 2D-3D assignment matrix Z^{t+1} with the minimum total cost. In the M step (Sec. 3.3.3), we update the parameter matrix to Θ^{t+1} via minimizing the total cost in the E step by gradient descent. The above problem definition and algorithm also apply to labeling superpoints.

3.3.2. E Step: Matching 2D and 3D Instances

In the E step, we aim to match 2D predicted instances with 3D instance labels and induce a log likelihood function of 3D instance logits Θ. Given the current estimate Θ^t and the instance segmentation masks $\mathcal{M}_k$ in the selected view k, we can define a cost function for a 2D-3D assignment Z. First, for each 3D instance label j, we denote a function Π_k to project its scores $\hat{\Theta}_{:,j}$ to a 2D image $\Pi_k(\hat{\Theta}_{:,j}) \in \mathbb{R}^{H\times W}$, where the score $\hat{\Theta}_{i,:}$ of the i-th point is induced by applying a softmax function to the logit $\Theta_{i,:}$, and H, W are the image height and width. Next, we denote the i-th 2D instance mask in view k by $\mathcal{M}_k^i \in \{0,1\}^{H\times W}$. Then, if the i-th 2D instance is assigned with a 3D instance label j, the cost function is defined as negative log-likelihood:

$$\begin{aligned} C(\Pi_k(\hat{\Theta}_{:,j}),\mathcal{M}_k^i) = -\sum_q \Big(&\mathcal{M}_k^i[q] log \Pi_k(\hat{\Theta}_{:,j})[q] \\ &+(1-\mathcal{M}_k^i[q]) log(1-\Pi_k(\hat{\Theta}_{:,j})[q])\Big) \end{aligned} \tag{4}$$

where q is a pixel position on the image. Given the cost function defined in Eq. 4, we use the Hungarian Algorithm to find the optimal assignment Z^{t+1}.

3.3.3. M Step: Optimizing 3D Instance Logits

In the M step, we can update the 3D instance logits Θ by minimizing the overall cost function $L(\Theta)$ given the assignment Z^{t+1} found in the E step. We use the gradient descent to update the parameters.

$$L(\Theta) = \sum_{i,j} [Z_{i,j} = 1] C(\Pi_k(\hat{\Theta}_{:,j}), \mathcal{M}_k^i) \tag{5}$$

3.3.4. Initialization

The EM algorithm can only find a local minimal, and a good initialization can typically lead to better solutions. Therefore, we use the 3D instance segmentation results from a pretrained PartSLIP checkpoint (introduced in Sec. 3.1) to initialize Θ^0. Assume that $\hat{m} \leq l$ instances are found by grouping superpoints in PartSLIP. For the i-th point and the 3D instance label $j \in \{1, \ldots, \hat{m}\}$, we have $\Theta^0_{i,j} = \log \hat{m}$ while $\Theta^0_{i,\neq j} = 0$.

3.3.5. Post-processing

A single 3D part instance is typically spatially adjacent, i.e., all points in a single instance form a single cluster based on spatial proximity. Our initial analysis finds that a 3D instance mask produced by our EM algorithm could sometimes contain multiple, disconnected instances. Therefore, we propose to further postprocess our 3D instances by splitting among them. Specifically, for each 3D instance, we use a spatial cluster algorithm similar to [14] to obtain one or more disjoint clusters among this instance. When an instance divides into multiple clusters, each becomes a separate 3D instance, retaining the original semantic label.

4. Experiments

In this section, we provide quantitative and qualitative analysis to demonstrate the ability for PartSLIP++ to outperform existing few-shot baselines in both 3D semantic and instance-based part segmentation. Subsequently, we perform an ablation study to justify each design component of PartSLIP++. Beyond these evaluations, we also demonstrate the versatility of PartSLIP++ in two practical applications: semi-automatic annotation of 3D parts and 3D instance proposals generation.

4.1. Datasets and Metrics

Following PartSLIP [21], we adopt the PartNet-Ensemble (PartNet-E) dataset introduced in the paper, which consists of 1906 shapes covering 45 object categories, to evaluate our approach and the baselines. Our experiments encompass two settings: **(a)** Few-shot (45×8): using 8 shapes for each of the 45 object categories. This setting is utilized in both our approach and the baseline. **(b)** Few shot with additional data ($45 \times 8 + 28k$): utilizing 28,367 shapes from PartNet[27] (which has 17 categories that overlap with PartNet-E) in addition to the 45 × 8 shapes. This setting is only utilized in the baseline. We evaluate the semantic segmentation performance with mIoU and the instance segmentation performance with mAP@50.

4.2. Implementation Details

To ensure a fair comparison, we use the dataset released by PartSLIP [21], which contains colored point clouds and camera poses used to render images. We follow the same setting to render each point cloud into 10 RGB images. We use the GLIP model finetuned on the low-shot data (45×8), which is also released by PartSLIP. *Note that the released checkpoint is known to have inferior performance compared to the version reported in the paper, confirmed by the authors of PartSLIP. We denote the released version by PartSLIP*.*

In our approach, to generate 2D instance masks given 2D detection results from GLIP, we utilize the pre-trained SAM [16] model (ViT-H) without further task-specific finetuning, and use the detected bounding boxes as input prompts. For the modified EM algorithm (Sec. 3.3), we use 10 EM iterations and the learning rate for gradient descent is 1.0 in the M step. We adopt a threshold of 0.05 for the spatial clutering algorithm used in post-processing (Sec. 3.3.5).

4.3. Evaluation Results

We compare our PartSLIP++ with PartSLIP [21] on both semantic segmentation and instance segmentation tasks. For semantic segmentation, we additionally compare PartSLIP++ with PointNet++ [31], PointNext [32], and SoftGroup [38]. For instance segmentation, we additionally compare PartSLIP++ with SoftGroup [38] and PointGroup [14].

Semantic Segmentation. We present the semantic segmentation results in Table 1. When training on the low-shot dataset of 45×8 shapes from PartNet-E, our PartSLIP++ attains the best performance compared to previous baselines. In particular, it outperforms released PartSLIP checkpoint by 2.9 mIoU (60.8 vs. 57.9) on the 45 categories in PartNet-E. The findings demonstrate PartSLIP++'s effectiveness in low-shot 3D semantic segmentation.

Instance Segmentation. We present the instance segmentation results in Table 2. We find that our PartSLIP++ also achieves the best performance, with a notable 7.7 mAP improvement (48.0 vs. 40.3) over the released PartSLIP checkpoint. Furthermore, when evaluating PartSLIP++ on the 17 overlapping categories between PartNet-E and PartNet, even though PartSLIP++ is only trained on 8 shapes from each category, it outperforms the best model (SoftGroup) trained on an additional 28,000 shapes from the PartNet dataset by 5.2 mAP (47.6 vs. 42.4). The results demonstrate that PartSLIP++ is a strong model for low-shot 3D instance segmentation.

Table 1. Semantic segmentation mIoU results on the PartNetE Dataset. We present results for the 17 object categories that overlap between PartNetE and PartNet, where in addition to the 8 training shapes from PartNetE per category, some baseline models also include an extra 28,000 shapes from PartNet, resulting in a total of 45x8+28k configurations. We also present results for the 28 unique categories in PartNetE, where models are trained using 8 PartNetE shapes from each category. For a detailed breakdown of performance on all 45 categories, please refer to the supplementary material.

#3D data	method	Overlapping Categories									Non-Overlapping Categories									
		Bottle	Chair	Display	Door	Knife	Lamp	Storage Furniture	Table	Overall (17)	Camera	Cart	Dis-Penser	Kettle	Kitchen-Pot	Oven	Suit-case	Toaster	Overall (28)	Overll (45)
Few-shot w/ extra data (45x8+28k)	PointNet++ [31]	48.8	84.7	78.4	45.7	35.4	68.0	46.9	**63.7**	55.6	6.5	6.4	12.1	20.9	15.8	34.3	40.6	14.7	25.4	36.8
	PointNext [32]	68.4	**91.8**	**89.4**	43.8	58.7	64.9	**68.5**	52.1	**58.5**	33.2	36.3	26.0	45.1	57.0	37.8	13.5	8.3	**45.1**	**50.2**
	SoftGroup [38]	41.4	88.3	62.1	**53.1**	31.3	**82.2**	60.2	54.8	50.2	23.6	23.9	18.9	57.4	45.5	13.6	18.3	26.4	30.7	38.1
Few-shot (45x8)	PointNet++ [31]	27.0	42.2	30.2	20.5	22.2	10.5	8.4	7.3	18.1	9.7	11.6	7.0	28.6	31.7	19.4	3.3	0.0	21.8	20.4
	PointNext [32]	67.6	65.1	53.7	46.3	59.7	55.4	20.6	22.1	39.2	26.0	47.7	22.6	60.5	66.0	36.8	14.5	0.0	41.5	40.6
	SoftGroup [38]	20.8	80.5	39.7	16.3	38.3	38.3	18.9	24.9	32.8	28.6	40.8	42.9	60.7	54.8	35.6	29.8	14.8	41.1	38.0
	ACD [8]	22.4	39.0	29.2	18.9	39.6	13.7	7.6	13.5	19.2	10.1	31.5	19.4	40.2	51.8	8.9	13.2	0.0	25.6	23.2
	Prototype [50]	60.1	70.8	67.3	33.4	50.4	38.2	30.2	25.7	41.1	32.0	36.8	53.4	62.7	63.3	36.5	35.5	10.1	46.3	44.3
	PartSLIP [21]	83.4	85.3	84.8	40.8	65.2	66.0	53.6	42.4	56.3	58.3	88.1	73.7	77.0	69.6	73.5	70.4	60.0	61.3	59.4
	PartSLIP* [21]	81.2	82.7	81.8	43.1	62.5	66.3	52.3	44.3	56.6	61.8	79.0	71.0	73.3	66.5	69.1	64.5	50.1	58.7	57.9
	Ours	**85.8**	85.3	85.1	45.1	**64.3**	67.9	57.2	45.3	**57.0**	**63.2**	**84.8**	**72**	**85.6**	**76.8**	**70.3**	**70.0**	**50.7**	**63.3**	**60.8**

Table 2. Instance segmentation mAP@50 results on the PartNetE Dataset. For more comprehensive performance on all 45 categories, please refer to the supplementary material.

#3D data	method	Overlapping Categories									Non-Overlapping Categories									
		Bottle	Chair	Display	Door	Knife	Lamp	Storage Furniture	Table	Overall (17)	Camera	Cart	Dis-Penser	Kettle	Kitchen-Pot	Oven	Suit-case	Toaster	Overall (28)	Overll (45)
45x8+28k	PointGroup [14]	38.2	87.6	65.1	**23.4**	19.3	62.7	**49.1**	**46.4**	41.7	8.6	29.2	24.0	61.3	59.4	13.8	15.6	7.0	24.6	31.0
	SoftGroup [38]	43.9	**89.1**	68.7	21.2	27.2	63.3	**49.1**	46.2	**42.4**	0.7	28.4	26.4	63.8	59.3	16.4	13.5	7.5	**25.6**	**31.9**
few-shot (45x8)	PointGroup [14]	8.0	77.2	16.7	3.7	15.6	9.8	0.0	0.0	14.6	4.7	28.5	30.7	52.1	57.0	0.0	0.0	0.0	16.8	16.0
	SoftGroup [38]	22.4	87.7	27.5	5.6	10.3	19.4	11.6	14.2	21.3	11.2	29.8	37.8	63.4	65.7	10.4	8.0	10.7	28.4	25.7
	PartSLIP [21]	79.4	84.3	82.9	17.9	43.9	68.3	32.8	32.3	42.5	36.8	**83.3**	**63.5**	75.4	70.5	**64.5**	44.9	38.4	46.2	44.8
	PartSLIP* [21]	74.4	79.3	64.2	14	43.3	**69.5**	29.2	32.1	41.1	29.6	71	59.7	72.5	70.3	46.3	44.6	34.9	39.8	40.3
	Ours	**78.5**	86.0	**74.1**	17.6	**46.0**	66.9	36.7	33.5	**47.6**	**29.7**	80.8	63.2	**81.6**	**80.7**	56.3	**49.6**	**41.5**	**48.2**	**48.0**

Qualitative Analysis. We present qualitative studies in Figure 2 to compare the 3D instance segmentation quality between PartSLIP++ and PartSLIP. Our observations reveal that PartSLIP++ excels in generating 3D instance masks that are more precise, accurate, and exhibit less noise. Notably, in challenging tasks like segmenting thin bucket handles, the base of a computer monitor, or the seat of a swing chair, PartSLIP++ demonstrates superior accuracy. The masks produced by PartSLIP often extend into undesired areas of the object, whereas those from PartSLIP++ maintain a higher level of precision and adherence to the correct object parts.

4.4. Ablation Studies

Design Choices in EM algorithm. Table 3 shows the ablation study on **3** design choices of our EM algorithm used in PartSLIP++: 1) whether to use the EM algorithm to refine initial 3D instance segmentations, 2) whether to initialize EM with 3D instance segmentations from PartSLIP, 3) whether to apply post-processing. We report the mAP@50 of different methods for 3D instance segmentation on different part categories.

We find that PartSLIP++ (full) outperforms PartSLIP++ (w/o EM) by 3.2 mAP (48.0 vs. 44.8), which demonstrates the effectiveness of our proposed modified EM algorithm in refining initial 3D instance masks. Besides, PartSLIP++ (full) outperforms PartSLIP++ (w/o PartSLIP init) by 1.7 mAP (48.0 vs. 46.3). This observation highlights the importance of the quality of 3D instance segmentation initialization in our EM algorithm. Additionally, PartSLIP++ (full) outperforms PartSLIP++ (w/o post-processing) by 1.1 mAP (48.0 vs. 46.9), illustrating that our 3D instance post-processing provides a helpful boost to the 3D instance segmentation performance. Therefore, all three components

Table 3. Ablation study on the EM algorithm used in PartSLIP++. We report the mAP@50 for 3D instance segmentation on all the part categories. The results on three categories (chair, kettle, suitcase) are shown as well.

Method	Chair	Kettle	Suitcase	Overall
PartSLIP++ (full)	86.0	81.6	49.6	48.0
w/o post-processing	82.7	78.6	49.2	46.9
w/o PartSLIP init	67.0	76.4	55.0	46.3
w/o EM	80.4	79.6	44.1	44.8
PartSLIP	79.3	72.5	44.6	40.3

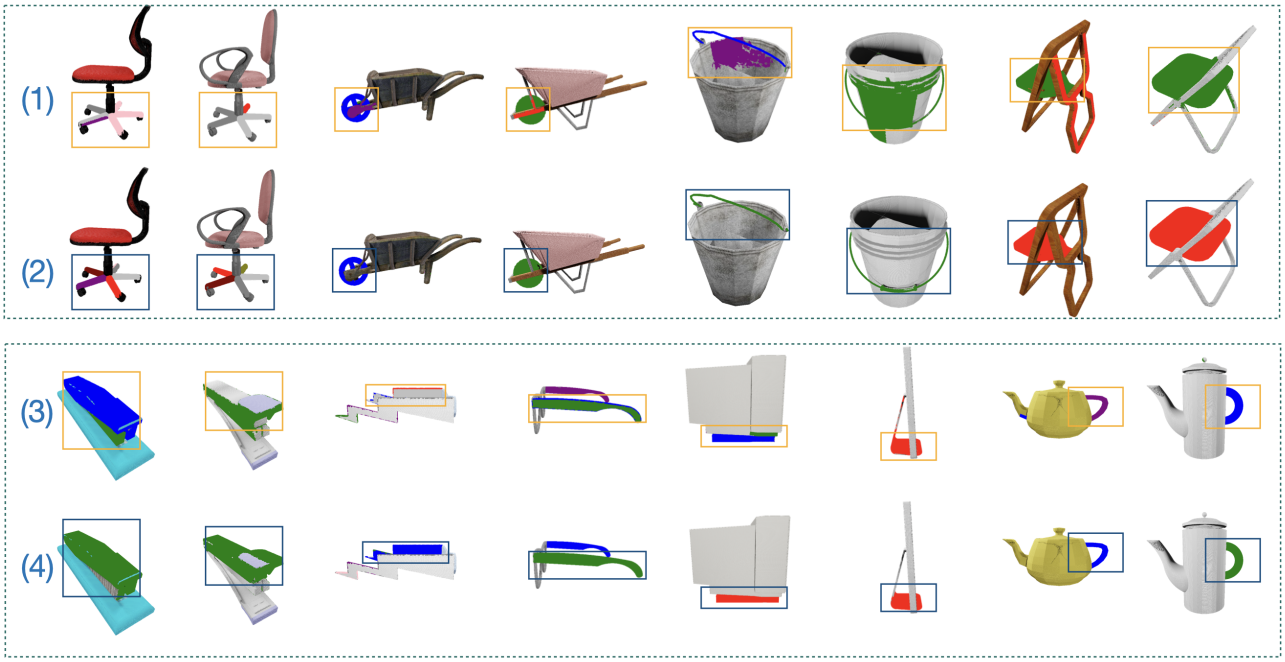

Figure 2. Qualitative analysis of 3D instance segmentation results for PartSLIP and PartSLIP++. Rows (1) and (3) illustrate the results from PartSLIP, and Rows (2) and (4) display the results from PartSLIP++. To enhance clarity, segmented instances are masked with a distinct color to differentiate from the object's original color, and are boxed to delineate the segmented areas. We find that in challenging tasks like segmenting thin bucket handles, the base of a computer monitor, or the seat of a swing chair, PartSLIP++ masks maintain a higher level of precision and adherence to the correct object parts, while PartSLIP masks often extend to undesired object areas.

proposed in PartSLIP++ play a significant role to the overall improvement over PartSLIP.

Refining 2D instance segmentations with SAM. We then perform an ablation to investigate the effectiveness of our design to refine 2D instance segmentations with SAM. Results are shown in Table 3. We find that PartSLIP++ (w/o EM) outperforms PartSLIP by 4.5 mAP (44.8 vs. 40.3), demonstrating the large improvements brought by the more accurate 2D instance segmentation results with the help of the SAM model.

Number of 2D Views. In our main experiments, we used 10 views to render each point cloud. In this ablation study, we investigate whether PartSLIP++ can benefit from more views that more comprehensively cover an object. We report the mAP@50 results for 3D instance segmentation on 3 part categories (Display, Door, Knife) in Table 4. The results confirm that PartSLIP++ produces improved 3D instance segmentation masks when provided with a broader range of views of an object. Furthermore, the benefits become much more modest when the EM module, a key component of PartSLIP++, is removed. This indicates the crucial role of our EM module in maximizing the gains from additional input views.

Table 4. Ablation study on the number of 2D input views (our previous experiments used 10 input views). We report the mAP@50 metric for 3D instance segmentation on three part categories: display, door, knife.

Method	Number of views	Display	Door	Knife
PartSLIP++	10	74.1	17.6	46.0
	24	77.8	24.8	51.1
	Gain	+3.7	+7.2	+5.1
PartSLIP++ w/o EM	10	69.5	17.7	42.6
	24	71.4	19.9	44.2
	Gain	+1.9	+2.2	+1.6

4.5. Application: Part Annotation

In this section, we illustrate the versatility of PartSLIP++ by illustrating its application in semi-automatic 3D object part annotation pipeline. In particular, PartSLIP++ is capable of segmenting 3D parts using multi-view 2D segmentation masks **without requiring the matching relationship between different views**. Based on this capability, we propose an annotation pipeline wherein annotators focus solely on labeling multi-view 2D images, assisted by the Segmen-

Table 5. mAP@50 results of 3D instance segmentation for PartSLIP++ and PartSLIP conditioned on multi-view manually-annotated 2D segmentations.

method	Chair	Suitcase	Knife
PartSLIP++	93.7	96.5	93.1
PartSLIP	88.1	97.3	84.5

Table 6. mAP@50 results of 3D instance segmentation for PartSLIP++ and PartSLIP conditioned on multi-view ground-truth 2D segmentations.

method	Chair	Suitcase	Knife
PartSLIP++	99.6	100	94.0
PartSLIP	96.3	100	94.0

tAnything. Once the multi-view images of a single object are fully annotated, our PartSLIP++ is automatically initiated in the backend. This process is designed to maximize efficiency in part annotation.

To test the robustness of this pipeline, we conduct a preliminary experiment. We randomly select several shapes from the PartNet-E dataset and manually annotate their 2D multi-view images. Subsequently, we independently apply the 3D instance mask generation pipelines in PartSLIP++ and PartSLIP to obtain 3D instance segmentations. Similar to our instance segmentation experiments, we employ mAP@50 as the evaluation metric. Results are shown in Table 5. To facilitate a comparison with human-annotated labels, we conduct an additional experiment where we condition PartSLIP++ and PartSLIP on multi-view ground-truth 2D segmentation masks. Results are presented in Table 6. We find that for both human-annotated 2D masks and ground truth 2D masks, PartSLIP++ produces better 3D instance segmentations than PartSLIP, demonstrating the potential for PartSLIP to enhance the efficiency and accuracy of semi-automatic 3D object part annotation.

4.6. Application: 3D Instance Proposal Generation

In this section, we showcase class-agnostic 3D instance proposal generation powered by SAM and our modified EM algorithm. For many applications like part annotation, semantic information is not mandatory (or can be annotated easily), while the recall over part instances is critical. This motivates us to extend PartSLIP++ for class-agnostic 3D instance proposal generation.

Concretely, we replace GLIP with SAM and leverage the "segment everything" ability of SAM to generate 2D instance proposals for each view. Then, our modified EM algorithm can be applied to merge 2D instance proposals from multiple views to 3D instance proposals. Fig. 3 showcases how this extension performs on *knifes*, which contain many fine-grained parts (e.g., blades) that are especially challenging for open-vocabulary object detection models like GLIP. Compared to the GLIP-based PartSLIP++, the SAM-based extension yields more refined segmentation, as shown by the higher count of successfully segmented parts.

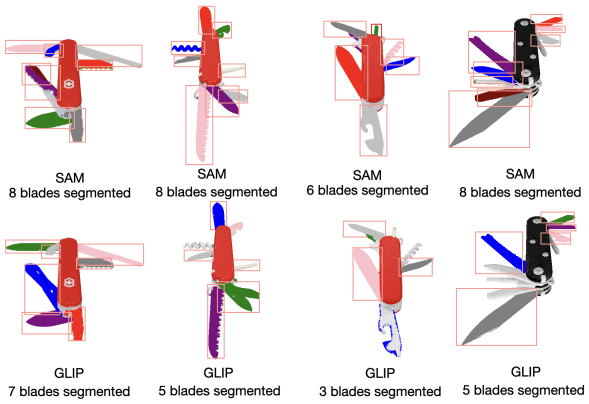

Figure 3. Example of 3D instance proposal generation. We extend PartSLIP++ by using SAM to directly generate class-agnostic instance proposals for each view and merging them with the modified EM algorithm. The first row shows the instance proposals generated by the (SAM-based) extension, and the second row shows the instances found by (GLIP-based) PartSLIP++. The number of blades segmented are shown below the visualization. The SAM-based extension shows a higher recall of part instances.

5. Conclusion

In this work, we propose PartSLIP++, a novel method for low-shot 3D semantic and instance segmentation on object parts that surpasses the limitations in the recent work PartSLIP. Specifically, PartSLIP++ first integrates a pre-trained 2D segmentation model to provide more accurate and precise 2D pixel-wise part annotations than the bounding boxes used in prior work. PartSLIP++ then formulates the problem of obtaining 3D instance segmentation from 2D multi-view instance labels as a maximum likelihood estimation problem, introducing a modified Expectation-Maximization (EM) algorithm for effective optimization. Through quantitative and qualitative analysis, we demonstrate that PartSLIP++ attains the best performance compared to previous approaches, and exhibits strong ability in low-shot 3D semantic and instance-based object part segmentation. We finally illustrate the versatility of PartSLIP++ in enabling diverse applications, such as semi-automatic part annotation and 3D instance proposal generation.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch,

Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022. 1

[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] Alexey Bokhovkin, Vladislav Ishimtsev, Emil Bogomolov, Denis Zorin, Alexey Artemov, Evgeny Burnaev, and Angela Dai. Towards part-based understanding of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7484–7494, 2021. 2

[4] Jiazhong Cen, Zanwei Zhou, Jiemin Fang, Wei Shen, Lingxi Xie, Xiaopeng Zhang, and Qi Tian. Segment anything in 3d with nerfs. *arXiv preprint arXiv:2304.12308*, 2023. 2

[5] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. Icm-3d: Instantiated category modeling for 3d instance segmentation. *IEEE Robotics and Automation Letters*, 7(1):57–64, 2021. 2

[6] Angela Dai and Matthias Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018. 2

[7] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society: series B (methodological)*, 39(1):1–22, 1977. 2, 4

[8] 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 *European Conference on Computer Vision*, pages 473–491. Springer, 2020. 6

[9] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018. 2

[10] Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In *European Conference on Computer Vision*, pages 564–580. Springer, 2020. 2

[11] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 2

[12] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019. 2

[13] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021. 1

[14] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 2, 5, 6

[15] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2

[16] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 2, 4, 5

[17] 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. 2, 3

[18] 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

[19] Jinxian Liu, Minghui Yu, Bingbing Ni, and Ye Chen. Self-prediction for joint instance and semantic segmentation of point clouds. In *European Conference on Computer Vision*, pages 187–204. Springer, 2020. 2

[20] 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

[21] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21736–21746, 2023. 1, 2, 3, 5, 6

[22] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 1

[23] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023. 2

[24] 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

[25] Ruben Mascaro, Lucas Teixeira, and Margarita Chli. Diffuser: Multi-view 2d-to-3d label diffusion for semantic scene segmentation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13589–13595. IEEE, 2021. 2

[26] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas J Guibas. Structurenet: Hierarchi-

cal graph networks for 3d shape generation. *arXiv preprint arXiv:1908.00575*, 2019. 1

[27] 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. 1, 5

[28] Alexandr Notchenko, Vladislav Ishimtsev, Alexey Artemov, Vadim Selyutin, Emil Bogomolov, and Evgeny Burnaev. Scan2part: Fine-grained and hierarchical part-level understanding of real-world 3d scans. *arXiv preprint arXiv:2206.02366*, 2022. 2

[29] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2

[30] 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

[31] 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. 5, 6

[32] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv:2206.04670*, 2022. 1, 5, 6

[33] 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. 1, 2

[34] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[35] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1

[36] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8216–8223. IEEE, 2023. 2

[37] 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

[38] 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, 5, 6

[39] 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.

[40] Xinlong Wang, Shu Liu, Xiaoyong Shen, Chunhua Shen, and Jiaya Jia. Associatively segmenting instances and semantics in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4096–4105, 2019. 2

[41] 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. 2

[42] 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. 1

[43] 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

[44] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019. 2

[45] 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. 1, 2

[46] 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

[47] 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

[48] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Harold Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *arXiv preprint arXiv:2206.05836*, 2022. 1

[49] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8552–8562, 2022. 2

[50] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021. 6