PartSTAD: 2D-to-3D Part Segmentation Task Adaptation

Hyunjin Kim[1†] and Minhyuk Sung[2]

[1] KRAFTON Inc., South Korea
[2] KAIST, South Korea
rlaguswls98@krafton.com, mhsung@kaist.ac.kr

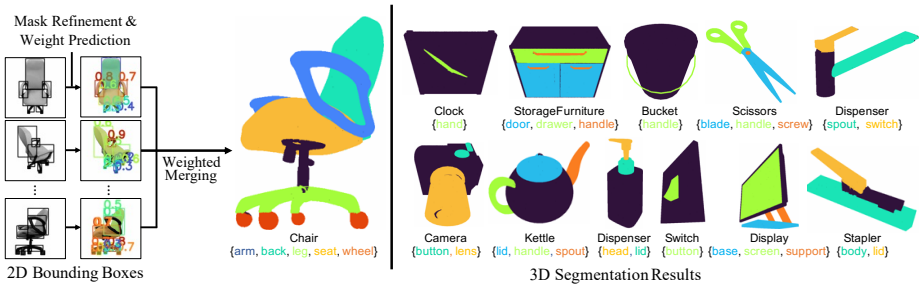

Figure 1: We introduce PartSTAD, a novel few-shot 3D point cloud part segmentation method that leverages 2D-to-3D task adaptation. By obtaining 2D segmentation masks in multi-view images from GLIP [25] and SAM [20] and optimizing the mask weights for 3D segmentation as a learning objective, it can successfully predict fine-grained parts with accurate boundaries, as shown in the figure above.

Abstract. We introduce PartSTAD, a method designed for the task adaptation of 2D-to-3D segmentation lifting. Recent studies have highlighted the advantages of utilizing 2D segmentation models to achieve high-quality 3D segmentation through few-shot adaptation. However, previous approaches have focused on adapting 2D segmentation models for domain shift to rendered images and synthetic text descriptions, rather than optimizing the model specifically for 3D segmentation. Our proposed task adaptation method finetunes a 2D bounding box prediction model with an objective function for 3D segmentation. We introduce weights for 2D bounding boxes for adaptive merging and learn the weights using a small additional neural network. Additionally, we incorporate SAM, a foreground segmentation model on a bounding box, to improve the boundaries of 2D segments and consequently those of 3D segmentation. Our experiments on the PartNet-Mobility dataset show significant improvements with our task adaptation approach, achieving a **7.0%p** increase in mIoU and a **5.2%p** improvement in mAP_{50} for semantic and instance segmentation compared to the SotA few-shot 3D segmentation model. The code is available at `https://github.com/KAIST-Visual-AI-Group/PartSTAD`.

Keywords: part segmentation · few-shot learning · 3D deep learning

† This work was conducted when the author was at KAIST.

1 Introduction

3D segmentation has been a subject of extensive research in computer vision due to its central role in various downstream applications involving the understanding of shape structure, functionality, mobility, and semantics. However, the limited availability of annotated 3D shapes has remained a bottleneck in achieving generalizability in learned segmentation models for diverse 3D data. Annotating 3D segmentation is particularly labor-intensive, time-consuming, and requires expertise in handling 3D models. For this reason, the scale of 3D part-level annotations remains in the tens of thousands [37], while the scale of 2D annotation datasets in the image domain, for example, exceeds a million [20, 30].

Recent research [2,32] has illuminated the potential of visual-language models bridging textual descriptions with images in accomplishing zero-shot or few-shot 3D segmentation results. The basic idea is to render a 3D model from various viewpoints, conduct 2D detection or segmentation on the rendered images, and then aggregate the 2D segmentation results into a 3D representation using either a voting mechanism [32] or a label propagation scheme [2]. This approach is not only effective in enabling zero-shot and few-shot 3D segmentation but also offers an advantage in that the set of part names does not need to be predefined in training but can be determined at test time.

PartSLIP [32] is a notable example that achieves 3D part segmentation results comparable to fully supervised methods while adapting a pretrained 2D detection model with few-shot training for 3D segmentation. It leverages the pretrained GLIP [25] model for 2D detection, associating the output 2D bounding boxes with one of the tokens (part names) provided in the input prompt. Its key component is the GLIP finetuning process, which introduces a small number of learnable parameters to the frozen GLIP model and trains them using few-shot synthetic images and texts employed in the 3D segmentation pipeline. As GLIP was initially trained with real photos and natural language descriptions of objects by humans, this adaptation to rendered images and the unconventional description (a sequence of part names) results in a substantial improvement in the 3D segmentation task.

Despite promising initiatives, it is important to note that the previous approach was limited to achieving *domain adaptation*. In contrast, we emphasize that in the 2D-to-3D segmentation lifting task, not only does the data domain change, but the task itself also shifts from 2D segmentation to 3D segmentation. Thus, the process of applying the pretrained model to the new task requires *task adaptation*, involving modifying the model with an objective function associated with the new task. Particularly for 3D segmentation, it is necessary to integrate 2D segmentation results from multiple viewpoints into a coherent 3D representation. Therefore, it is crucial to control the noise of 2D bounding box predictions in the context of its impact on the final 3D segmentation after integration.

Specifically, our Part Segmentation Task ADaptation method, PartSTAD, adapts the pretrained GLIP model with a relaxed 3D mIoU loss while incorporating bounding boxes across all different views. Instead of typical finetuning, we draw inspiration from recent work [16, 32] that introduces small learnable

parameters to the existing model while keeping existing parameters frozen. As a result, we train a small MLP using 2D bounding box features extracted from the pretrained GLIP model. Since the relaxed mIoU loss is non-differentiable with respect to the bounding box positions, we suggest predicting a *weight* for each bounding box instead of adjusting its position. This results in minimal modification in the 2D-to-3D voting scheme while achieving a significant improvement in 3D segmentation, even with few-shot training, e.g., with eight objects per category, as done in PartSLIP. To further enhance 3D segmentation performance, we leverage SAM [20], a 2D instance segmentation model, to segment the foreground region within each bounding box, obtaining an accurate boundary for each 2D segment.

In our experiments on the PartNet-Mobility [56] dataset, we showcase the superior performance of our method in comparison to recent zero-shot/few-shot 3D semantic and instance segmentation methods that leverage 2D segmentation models. In contrast to the SotA few-shot method, our approach achieves a **7.0%p** improvement in semantic segmentation mIoU and a **5.2%p** improvement in instance segmentation mAP_{50}. These improvements are consistent across all object classes.

2 Related Work

2.1 Supervised 3D Segmentation

Given the availability of open segment-annotated 3D datasets [3,5,7,8,12,14,37, 56], 3D segmentation has been extensively researched in the last several years, focusing on the development of novel network architectures for supervised learning. Regarding the architecture of semantic segmentation, diverse models have been explored, including PointNet [42] and its variants [36, 43, 44, 57, 71], those using CNN [27, 34, 49, 59], GCN [17, 28, 53], and Transformers [55, 60, 67, 73]. For instance segmentation, various approaches based on proposing 3D bounding boxes [15, 61, 66], clustering learned features [11, 19, 31, 38, 50, 51, 69, 74], and combining both semantic and instance segmentations [29, 40, 46, 52] have been introduced. Despite these great advances, the performance of supervised methods has been limited by the scale of 3D datasets; the largest part-annotated 3D dataset, PartNet [37], includes fewer than 30k models.

There have been some attempts to overcome the limitation by leveraging additional weak supervisions, such as text descriptions of objects [21], 3D bounding box annotations [10], sparse labels [26, 35], and geometric priors [23, 24]. However, the scale of this additional data has also been limited to tens of thousands, while datasets in the image domain are on the scale of millions.

2.2 3D Segmentation Using Vision-Language Models

Recent work introduced ideas about exploiting vision-language models (VLM) for 3D segmentation. The advantage of utilizing the learned visual-language

grounding in 3D segmentation lies in the strong generalizability to arbitrary 3D models with zero-shot or few-shot training, and also in the open vocabulary understanding that enables labeling points without specifying the set of part or object names during training. For a VLM, CLIP [45] has been adapted in multiple previous works to lift the 2D grounding to the 3D, detecting parts in an object [13] and objects in a scene [9, 39, 47, 70], while the results are limited to highlighting some regions without providing clear segment boundaries.

To obtain precise segments in 3D, other 2D object detection and segmentation models, such as SAM [20], GLIP [25], and GroundingDINO [33], are also leveraged. SAM3D [63] was an example of directly lifting the 2D masks from SAM to 3D with a bottom-up merging approach, although it was limited to segmenting the parts, not labeling them. SA3D [6] utilized NeRF representation and allowed user interaction for 3D segmentation, but it can be inefficient when the 3D objects are given as a mesh, point cloud, or other common 3D representations. OpenMask3D [47] and OpenIns3D [18] are concurrent works, both of which combine multiple foundation models, SAM [20] and CLIP [45] for OpenMask3D, and GroundingDINO [33] and LISA [22] for OpenIns3D. These works focus on segmenting objects in 3D scenes, while we aim to do finer-grained segmentation, finding parts in 3D objects.

PartSLIP [32] and SATR [2] are notable examples that achieve part segmentation of 3D objects based on a 2D object detection model, GLIP [25]. They obtain 2D bounding boxes on the rendered images from multiple views and integrate them over the 3D object with different merging schemes. The finetuning of GLIP introduced by PartSLIP particularly leads to a performance improvement, but it is limited to domain adaptation, not task adaptation. Building on this, we introduce a novel task adaptation technique as well as integration with SAM [20] to achieve a further significant improvement in 3D segmentation accuracy.

2.3 Task Adaptation

Taskonomy [68] is a seminal work that introduced the concept of task transfer learning for the first time, enabling the transformation of an image dense prediction model trained for a base task to perform other tasks through finetuning. Pruksachatkun *et al.* [41] also introduced a similar idea for language-domain tasks. Later, task adaptation has been extensively studied to leverage features learned from generative models, such as GAN [72] and diffusion models [4, 48, 58, 62], adapting them with a small network for various tasks, including image segmentation [4, 58, 72], depth prediction [48], and keypoint detection [62]. For 3D tasks, Abdelreheem *et al.* [1] are notable examples of using vision language-grounding models for 3D correspondence but are limited to directly applying 2D priors without finetuning. We introduce a novel task adaptation technique that lifts 2D segments to 3D by adapting the 2D features during the training of a small network with a task-specific objective function.

3 Background and Preliminaries

Recent research [2,13,32] has demonstrated how vision-language models designed for image object detection and segmentation can be applied to segment parts in 3D objects. The main idea involves rendering a 3D model from various viewpoints, performing 2D detection or segmentation for the rendered image at each view, and then combining the 2D segmentation results over the 3D model.

Specifically, SATR [2] utilizes GLIP [25] as the 2D detection model. To obtain 2D bounding boxes for each of the specified part names, it feeds a concatenated list of these part names (separated by commas) as the input prompt to the pretrained GLIP model, along with a rendered image. The 2D bounding boxes are then mapped to a single face of the input 3D mesh, located at the center of the respective 2D bounding box, and the part names are propagated throughout the entire mesh based on geodesic distances. Similarly, PartSLIP [32] also leverages GLIP [25] for 2D detection but employs a different voting scheme for the 2D bounding box integration over the 3D object, which utilizes the oversegmentation of the input point cloud. One key distinction between PartSLIP and SATR is that PartSLIP proposes to modify the GLIP model for domain adaptation to the rendered images and text prompts used in 3D segmentation. Specifically, the text prompt is given as a concatenation of part names in the PartSLIP pipeline, which is not a typical and natural description of the object. Therefore, it learns a category-specific constant offset vector for the language embedding feature in the given pairs of prompts and rendered images, while keeping the GLIP parameters frozen.

Our work builds upon the PartSLIP pipeline. In the following subsections, we describe the details of its major components: the voting scheme, and the finetuning process.

3.1 PartSLIP [32]

The main technical components of PartSLIP are the 1) voting scheme based on super points and 2) adaptation to text prompts[1]. We describe the details of each component below.

Voting for Super Points. PartSLIP first over-segments the input 3D object represented as a point cloud $\mathcal{P}$ into a set of super points $P_i \subseteq \mathcal{P}$, which provide geometric priors for the boundaries of the parts. The 3D object is rendered into K viewpoints, and the 2D bounding boxes for each image are predicted using GLIP [25]. Let $\mathcal{B}$ denote the entire set of 2D bounding boxes from all views, and $B_{jk} \subseteq \mathcal{B}$ indicates a subset that includes the bounding boxes classified as the j-th part label from the k-th view. $\mathrm{V}_k : \mathcal{P} \to \{0, 1\}$ is a function indicating whether the input point is visible from the k-th viewpoint. Also, for each 2D

[1] Although multi-view feature aggregation is also introduced, the improvement by this component is marginal, and the author of PartSLIP also did not include this component in their official code. Therefore, we will omit this part in our review.

bounding box $b \in \mathcal{B}$, $\mathrm{I}_b : \mathcal{P} \to \{0, 1\}$ is a function indicating whether the input point is included in the bounding box b. Given these, the voting aggregating the semantic labels of the 2D bounding boxes to the 3D point cloud is performed by calculating the following score s_{ij} for each pair of the i-th super point and the j-th label:

$$s_{ij} = \frac{\sum_k \sum_{p \in P_i} \mathrm{V}_k(p) \left(\max_{b \in (B_{jk} \cup \{b_\emptyset\})} \mathrm{I}_b(p) \right)}{\sum_k \sum_{p \in P_i} \mathrm{V}_k(p)}, \tag{1}$$

where $b_\emptyset$ is a *null* bounding box, and we assume that $\mathrm{I}_{b_\emptyset}$ for the null bounding box returns zero for any input point (simplifying notations for the case when $B_{jk} = \emptyset$). Note that this $s_{ij} \in [0, 1]$ denotes the ratio of visible points from each view included in any of the 2D bounding boxes labeled with the j-th label. Finally, for each i-th super point, the label with the highest score s_{ij} across all the labels is assigned to the super point. Exceptionally, if the highest score is less than a null label threshold $s_\emptyset$, which is set to 0.5 in our experiments, the null label is assigned to the super point.

GLIP Finetuning. While GLIP demonstrates impressive generalizability for unseen images and prompts, it still has limitations when it comes to handling synthetic images and text prompts, such as the rendered images and sequences of part names used in the PartSLIP pipeline. To address this, the authors of PartSLIP propose adding small, learnable parameters to the GLIP model while keeping all the pretrained GLIP parameters frozen. These new parameters are learned with very few-shot examples: 8 annotated 3D objects per category. They represent offset feature vectors for each part name and remain constant as global variables for each category, rather than changing for each input. This adaptation of GLIP significantly improves the accuracy of 3D part segmentation.

Instance Segmentation. The voting scheme described above is designed for semantic segmentation, but PartSLIP also introduces a straightforward merging-based method to achieve instance segmentation based on semantic segmentation. To obtain instance segments, it merges adjacent super points under two conditions: 1) they share the same semantic label, and 2) they are either both included or both excluded for all bounding boxes.

4 PartSTAD: Task Adaptation for 2D-to-3D

PartSLIP has demonstrated that adapting a 2D detection model for 3D segmentation with a small set of trainable parameters can significantly enhance 3D segmentation performance. However, we observe that PartSLIP's adaptation is limited to *domain adaptation*, where it learns new parameters using synthetic images and data used in the PartSLIP framework without altering the *objective function* during training.

When applying the 2D segmentation model to 3D segmentation, it is important to note that not only does the data domain change, but the *task* itself

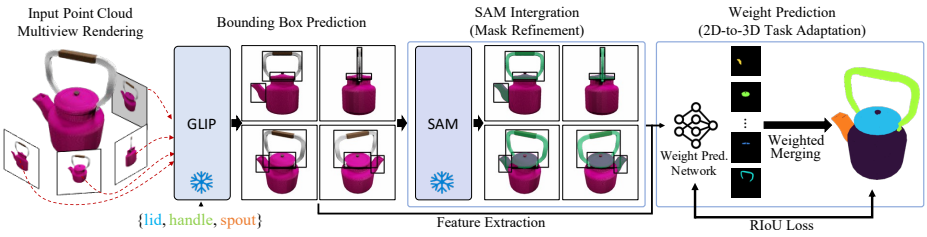

Figure 2: Overall pipeline of PartSTAD. Our approach begins by rendering the provided 3D point cloud from multiple viewpoints. Subsequently, we extract 2D bounding boxes for its parts using GLIP [25] (Bounding Box Prediction); note that we utilize the finetuned GLIP model from PartSLIP [32]. Following this, we convert the bounding boxes into segmentation masks using SAM [20], extracting the foreground region for each bounding box (SAM Mask Integration). Next, we predict weights for all the masks and adaptively combine them into a 3D representation (2D-to-3D task adaptation). The final step involves obtaining the segmentation label for the input point cloud. The GLIP and SAM models are frozen, while only our novel weight prediction network is trained per category in a few-shot setting (8 objects).

changes. Therefore, through the process of incorporating 2D bounding boxes into the 3D object, the 2D bounding box prediction model must align with our ultimate goal: 3D part segmentation.

To address this, we introduce Part Segmentation Task ADaptation method, PartSTAD, a *task adaptation* approach designed to finetune the 2D segmentation model for 3D part segmentation. In Sec. 4.1, we introduce our objective function to finetune the GLIP for the 3D segmentation task. In Sec. 4.2, we describe how we adapt the pretrained GLIP model with the new objective function by introducing additional learnable parameters and modifying the voting scheme (Eq. 1). PartSLIP also faces a performance limitation due to the GLIP that provides 2D bounding boxes, not 2D segments. In Sec. 4.3, we also propose to combine the SAM [20] foreground mask for each bounding box to obtain more precise 2D segments and, consequently, further improve the 3D segmentation.

4.1 3D mRIoU Loss

We begin by describing the loss function that we use for adapting the GLIP to the 3D segmentation task. Let $\mathbf{l}_j$ and $\hat{\mathbf{l}}_j \in \{0,1\}^{|\mathcal{P}|}$ be binary vectors indicating whether each point p in $\mathcal{P}$ has the j-th label in the ground truth and label prediction, respectively. The mean Intersection over Union (mIoU) for the predicted 3D segmentation, the standard evaluation metric for 3D segmentation, is defined as follows:

$$\text{mIoU}(\{\mathbf{l}_j\}, \{\hat{\mathbf{l}}_j\}) = \frac{1}{M}\sum_{j=1}^{M} \frac{\mathbf{l}_j^\top \hat{\mathbf{l}}_j}{\|\mathbf{l}_j\|_1 + \|\hat{\mathbf{l}}_j\|_1 - \mathbf{l}_j^\top \hat{\mathbf{l}}_j}, \tag{2}$$

where M is the number of labels.

Our goal is to use the 3D mIoU as the objective function directly when adapting the GLIP model. However, the mIoU is non-differentiable. Thus, we employ a relaxed version known as mean Relaxed IoU (mRIoU) [24, 65], which simply allows the predicted indicator vectors $\hat{\mathbf{l}}_j$ to be non-binary: $\hat{\mathbf{l}}_j \in [0,1]^{|\mathcal{P}|}$. The loss $\mathcal{L}_{\text{mRIoU}}$ is specifically defined as follows:

$$\mathcal{L}_{\text{mRIoU}} = 1 - \text{mIoU}(\{\mathbf{l}_j\}, \{\hat{\mathbf{l}}_j\}). \tag{3}$$

In the supplementary material, we demonstrate that the choice of the loss function, as compared to other alternatives such as the cross-entropy loss, is crucial to achieving a substantial improvement in our task adaptation.

Note that in the PartSLIP framework, where super points are utilized in 3D segmentation, and $s_{ij} \in [0,1]$ indicates the likelihood of the i-th super point belonging to the j-label, the mRIoU can be calculated by defining $\hat{\mathbf{l}}_j$ based on $\mathbf{s}_j = [s_{1j}, s_{2j}, \cdots]^\top$ as follows:

$$\hat{\mathbf{l}}_j = \mathbf{M}\mathbf{s}_j, \tag{4}$$

where $\mathbf{M} \in \{0,1\}^{|\mathcal{P}| \times |\{P_i\}|}$ is a binary matrix that describes the memberships from each point to the super points.

4.2 Bounding Box Weight Prediction

A typical approach for adapting a pretrained model to a new task would be finetuning the model using an objective function tailored to that new task. However, the finetuning method generally fails to yield meaningful improvements, especially when the goal is to achieve few-shot adaptation (e.g., with only 8 annotated 3D objects per class, as done in our case), while the pretrained model has been trained on an extensive dataset. To address this challenge, recent works like LoRA [16] and PartSLIP [32] have introduced the concept of adding small learnable parameters while keeping the existing pretrained model parameters frozen. This approach not only makes training very efficient but also enables the model to generalize effectively to unseen data.

While we also follow this approach, our specific challenge arises from the fact that when we use the 3D mRIoU (Eq. 3) as the objective function for 3D segmentation, it becomes non-differentiable with respect to the 2D bounding box location, which is the output of GLIP. This non-differentiability arises from the computation of the score s_{ij} (Eq. 1) for each pair of super point and label.

To address this, we propose training a small network that does not refine the location but instead predicts a *weight* for each 2D bounding box, taking the learned features of the boxes as input. If we denote the output weight of the network for bounding box b as $\text{W}(b) \in \mathbb{R}^+$, the score s_{ij} for a pair of super point and label is modified to $\tilde{s}_{ij}$ as follows:

$$\tilde{s}_{ij} = \frac{\sum_k \sum_{p \in P_i} \text{V}_k(p) \left(\max_{b \in (B_{jk} \cup \{b_\emptyset\})} \text{I}_b(p) \text{W}(b) \right)}{\sum_k \sum_{p \in P_i} \text{V}_k(p)}. \tag{5}$$

Note that the change is simply to multiply the weight $\mathrm{W}(b)$ by $\mathrm{I}_b(p)$, while only $\mathrm{I}_b(p)$, indicating whether the point p is included in bounding box b, has been used in Eq. 1. This modified score no longer falls within the range of $[0, 1]$. The final score $\bar{s}_{ij}$ is thus defined by normalizing $\tilde{s}_{ij}$ using the softmax function over the set of labels:

$$\bar{s}_{ij} = \frac{\exp(\tilde{s}_{ij})}{\sum_j \exp(\tilde{s}_{ij})}. \tag{6}$$

We additionally propose to make the unnormalized score for the null label $\tilde{s}_\emptyset$ *learnable* in our case, initialized with 10. We thus compute the softmax above while including $\tilde{s}_\emptyset$. The label of each super point is still chosen as the label giving the highest score $\bar{s}_{ij}$, becoming null if the normalized null label score $\bar{s}_\emptyset$ becomes the highest.

Network Architecture. We design the weight predictor network to take the bounding box feature vectors $\mathbf{f}_b$ from the pretrained GLIP model as input. The feature vectors of all bounding boxes across all views are fed to the network at once and processed with a small shared two-layer MLP. Context normalization [64] is added in the middle of the two layers to incorporate contextual information from the global set of boxes for each box. The output of the MLP for each bounding box is further processed with the following modified ReLU function $\phi(\cdot)$:

$$\phi(\mathbf{x}) = \max\left(\tau + \mathbf{x}, 0\right), \tag{7}$$

where τ is a user-defined constant offset (10 in our experiments).

4.3 SAM [20] Mask Integration

Another limitation of PartSLIP is its reliance on GLIP [25] for 2D segmentation. Since GLIP is, in turn, a 2D object detection model, it produces bounding boxes instead of 2D segments, which cannot provide accurate boundaries of segments. We propose to address this issue by incorporating another pretrained 2D segmentation model, SAM [20]. SAM has the functionality of taking a bounding box as input and producing the foreground mask. Using this, we replace the set of 2D bounding boxes $\mathcal{B}$ with a set of 2D masks, while preserving the remaining steps in the framework. Note that the point-to-bounding-box membership I_b is changed to point-to-mask membership, and we still use the same bounding box features extracted from GLIP corresponding to each mask to train the weight prediction network. We demonstrate the effectiveness of applying the SAM mask in our experiments in Sec. 5.

5 Experiment Results

In the experiments, we compare our method with the SotA 2D-prior-based zero-shot/few-shot 3D segmentation models using the PartNet-Mobility [56] dataset

Figure 3: Qualitative comparison of semantic segmentation results. Our PartSTAD segments 3D parts more precisely with clearer boundaries, even for small (Camera, Chair) and thin (Clock) parts.

Table 1: Quantitative comparison of the semantic segmentation results (mIoU(%)) on the PartNet-Mobility dataset. Ours achieves a **7.0%p** improvement in average mIoU compared to PartSLIP [32], the SotA few-shot 3D segmentation method, while consistently increasing mIoU across all categories. Please refer to the supplementary material for the complete table with results for all 45 categories.

Method	mIoU	Storage Furniture	Table	Chair	Switch	Toilet	Laptop	USB	Remote	Scissors
SATR [2]	29.3	20.6	23.3	33.1	21.4	17.6	11.2	30.2	17.2	36.8
SATR [2]+SP	34.8	28.9	28.0	37.7	37.0	22.1	12.4	33.4	28.0	43.0
PartSLIP [32]	58.0	52.3	44.6	82.8	52.1	50.4	31.2	52.1	36.6	61.4
Ablation Study										
w/o Weight Pred.	61.9	56.6	45.0	85.0	51.9	56.6	31.5	57.1	36.6	60.8
w/o SAM Integ.	62.1	54.0	45.7	83.1	53.0	49.4	33.6	53.6	46.8	67.5
PartSTAD (Ours)	**65.0**	**59.5**	**47.8**	**85.3**	**57.9**	**57.5**	**34.6**	**59.9**	**53.4**	**68.5**

and OmniObject3D [54] dataset. In the **supplementary material**, we present additional results on the ablation study, outcomes with scanned data, and comprehensive results covering all PartNet-Mobility categories.

5.1 Experiment Setup

Dataset. We use the PartNet-Mobility [56] dataset, which is also used in the experiments of PartSLIP [32]. The PartNet-Mobility [56] dataset includes 45 object categories. For each category, the training split includes 8 shapes, while the test split has a range of the number of objects from 6 to 338, totaling 1,906 objects across all 45 categories. Following PartSLIP, 8 shapes from the test set of each category are designated as the validation set. To obtain the 2D bounding boxes for each object, we use 10 images from fixed viewpoints across all our experiments. Note that the PartNet-Mobility [56] dataset is identical to the subset

of PartNet-Ensembled dataset used to train and evaluate PartSLIP [32]. (The remaining part of PartNet-Ensembled was only used to train other supervised baseline models, not PartSLIP [32].)

Evaluation Metrics. As evaluation metrics, we employ the *mean Intersection over Union* (**mIoU**) metric for semantic segmentation and *mean Average Precision* (**mAP**) for instance segmentation.

For semantic segmentation, the average mIoU over the entire dataset is computed by first calculating it per semantic part, averaging them across the parts in the same category, resulting in category-level mIoU, and then averaging the category-level mIoUs again across all the categories.

For instance segmentation, we compute two types of mAP: *part-aware* mAP and *part-agnostic* mAP. For part-aware mAP, similar to mIoU in semantic segmentation, we first calculate the mAP for each semantic part with instances having the part label. We then average them for each category, and then across categories. For part-agnostic mAP, we do not consider the semantic labels and directly compute the category-level mAP with part instances in the same object category, and then average them. All mAP values correspond to mAP_{50} with an IoU threshold of 50%.

Baselines. We compare our method with three baselines: PartSLIP [32], SATR [2], and SAM3D [63].

- **PartSLIP** [32] is the method that we adopt as our base. Note that PartSLIP uses a finetuned GLIP model, and we also employ the same model in our framework to extract bounding box features for our task adaptation.
- **SATR** [2] is similar to PartSLIP, conducting 3D segmentation using 2D bounding boxes from GLIP [25]. However, SATR differs from PartSLIP in four key aspects: 1) it does not involve finetuning the GLIP model, 2) it does not utilize super points, 3) it is designed for mesh segmentation, not point cloud segmentation, and 4) it employs a distinct approach for integrating 2D bounding boxes into 3D (label propagation). To ensure a fair comparison between our method and SATR, we implement the following changes to SATR. First, since we employ the GLIP model finetuned by PartSLIP in our framework, we use the same finetuned model within the SATR framework. Second, we convert the point cloud input to a mesh, performing mesh segmentation with SATR, and apply the results to the input point cloud by finding the closest vertex for each point. Lastly, we introduce a version of SATR using super points, resulting in a comparison between our method and two versions of SATR: one without super points (denoted as *SATR*) and one with super points (denoted as *SATR+SP*). Since SATR does not perform instance segmentation, our comparison with it is focused on semantic segmentation.
- **SAM3D** [63] is another zero-shot 3D segmentation method using a 2D segmentation model, not GLIP [25], but SAM [20]. Since it only performs instance segmentation but does not label each segment, we compare our

method with it only for instance segmentation, using the part-agnostic mAP metric.

Ablation Study. We also conduct an ablation study, comparing our method with two major components being ablated: the weight prediction (Sec. 4.2) and the SAM mask integration (Sec. 4.3). In the **supplementary material**, we also compare our method with cases using cross-entropy loss instead of mRIoU loss (Sec. 4.1) and using GLIP bounding box confidence score instead of our learned weights in the weighted voting (Sec. 4.2). Note that if both weight prediction and SAM mask integration are excluded, our PartSTAD is identical to PartSLIP [32].

5.2 Semantic Segmentation Results

Tab. 1 and Fig. 3 present the quantitative and qualitative comparisons of part semantic segmentation results, respectively. Please refer to the supplementary material for the complete table encompassing all 45 categories. In comparison to PartSLIP [32], a SotA few-shot 3D segmentation method, we achieve a significant **7.0%p mIoU** improvement over the entire set of 45 categories. Furthermore, for each specific category, we consistently demonstrate improvement, surpassing by more than 15%p in some categories (e.g., Remote). Qualitatively, the results also highlight the effectiveness of our task adaptation and the SAM mask integration in segmenting parts more precisely, even for small (e.g., Camera, Chair) and thin (e.g., Clock) parts of objects.

SATR exhibits inferior performance, even when leveraging the finetuned GLIP model and super points (see SATR+SP results) due to the label-propagation-based 2D bounding box integration, which often fails to provide clear boundaries in 3D segmentation, as illustrated in Fig. 3.

The ablation study results demonstrate the influence of each major component in our framework. The quantitative findings indicate that the weight prediction network plays a crucial role in substantial improvement, leading to a 3.1%p decrease in average mIoU when it is ablated. The SAM mask integration also contributes significantly, resulting in a 2.9%p average mIoU decrease when ablated.

5.3 Instance Segmentation Results

Following the instance segmentation idea of PartSLIP, we also perform instance segmentation using our method and compare the quantitative results with those of other methods in Tab. 2 and Tab. 3, reporting part-aware mAPs and part-agnostic mAPs, respectively. Fig. 4 also displays qualitative comparisons across different methods. Please refer to the supplementary material for the complete table of all 45 categories. Similar to semantic segmentation, ours achieves **4.0%p** and **5.2%p** improvements for both average part-aware mAPs and part-agnostic mAPs, respectively. Both the weight prediction network and SAM mask integration also exhibit meaningful improvements in performance, as their influence is

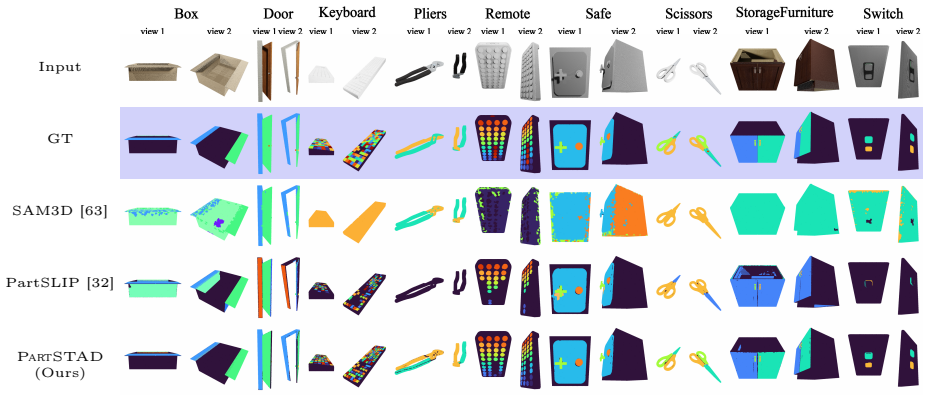

Figure 4: Qualitative comparison of instance segmentation results shows that our PartSTAD successfully segments tiny 3D parts, such as the keys of keyboards and buttons of remote controls, with clear segment boundaries.

Table 2: Part-aware mAPs(%) of the instance segmentation results on the PartNet-Mobility dataset. Ours achieves **4.0%p** improvement in mean part-aware mAPs compared to PartSLIP [32], with consistent enhancements across all categories. Please refer to the supplementary material for the complete table with results for all 45 categories.

Method	mAP	Storage Furniture	Table	Chair	Switch	Toilet	Laptop	USB	Remote	Scissors
PartSLIP [32]	41.6	29.1	32.6	82.2	21.2	36.2	17.8	20.9	19.9	23.6
Ablation Study										
w/o Weight Pred.	44.7	33.8	30.8	**82.5**	**22.2**	**40.8**	24.4	17.4	20.3	25.5
w/o SAM Integ.	44.2	29.4	32.3	82.2	19.7	36.0	23.6	24.8	29.6	23.1
PartSTAD(Ours)	**45.6**	**35.5**	**33.2**	**82.5**	22.1	40.6	**28.9**	**26.5**	**33.7**	**26.5**

Table 3: Part-agnostic mAPs(%) of the instance segmentation results on the PartNet-Mobility dataset. Ours demonstrates **5.2%p** improvement in mean part-agnostic mAPs compared to PartSLIP [32], surpassing SAM3D with a mean mAP that is more than three times larger. Please refer to the supplementary material for the complete table with results for all 45 categories. (*SAM3D is a zero-shot method.)

Method	mAP	Storage Furniture	Table	Chair	Switch	Toilet	Laptop	USB	Remote	Scissors
SAM3D* [63]	12.1	1.2	10.6	5.5	5.4	3.7	1.5	22.6	1.0	7.2
PartSLIP [32]	38.9	29.7	28.7	80.7	21.2	35.4	19.5	20.5	19.9	27.0
Ablation Study										
w/o Weight Pred.	42.6	35.9	**29.2**	81.1	22.2	40.0	26.5	15.2	20.3	**29.4**
w/o SAM Integ.	42.6	34.8	27.6	81.2	19.7	36.6	27.4	**27.3**	29.6	24.9
PartSTAD (Ours)	**44.1**	**41.7**	28.2	**83.3**	**22.4**	**41.0**	**34.1**	23.5	**33.7**	26.2

depicted in the ablation study results. Compared to SAM3D [63], whose part-

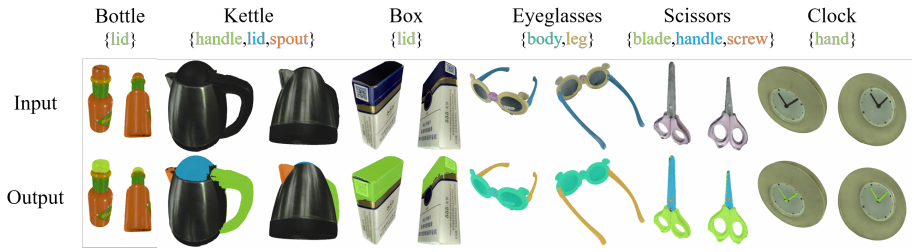

Figure 5: Qualitative comparison of semantic segmentation results on OmniObect3D [54] dataset, a high quality real scanned 3D objects dataset. Our PartSTAD predicts precise boundaries, even in the case that the object has an appearance significantly different from training data such as box category.

agnostic mAPs are shown in Tab. 3, our method achieves much better performance with more than three times greater mAP. Qualitative results also illustrate the outstanding performance of our method, accurately identifying all the tiny instances of parts, such as keys on keyboards, buttons on remotes, and clear boundaries of these instances.

5.4 Part Segmentation with Real Scanned Dataset

Fig. 5 displays additional results on the OmniObject3D [54] dataset, a high-quality dataset featuring real-scanned 3D objects. We present qualitative results only, as ground truth segmentation for OmniObject3D [54] is unavailable. Note that our PartSTAD accurately identifies parts in real scans, demonstrating its effectiveness beyond synthetic data. Particularly impressive is its performance in the box category depicted in Fig. 5, where despite significant discrepancies in appearance compared to the training data, PartSTAD achieves remarkably accurate predictions.

6 Conclusion and Future Work

We presented PartSTAD, a task adaptation method for lifting 2D segmentation to 3D. Instead of directly finetuning a 2D segmentation network, our method learns a small neural network predicting weights for each 2D bounding box with an objective function for 3D segmentation, and then performs 3D segmentation via weighted merging of the boxes in 3D. We further improve the performance of 3D segmentation by integrating SAM foreground masks for each bounding box. We achieve the SotA results in few-shot 3D part segmentation, demonstrating significant improvements in both semantic and instance segmentations.

As PartSTAD still depends on 2D representations for 3D predictions, its understanding of 3D geometry is relatively limited, and thus it cannot account for occluded or interior points, which poses a limitation. In future work, we plan to further investigate combining our approach with the direct utilization of 3D representations, such as leveraging 3D features, for more accurate predictions.

Acknowledgements

We sincerely thank Minghua Liu for providing the code of PartSLIP and answering questions. This work was supported by NRF grant (RS-2023-00209723), IITP grants (2022-0-00594, RS-2023-00227592, RS-2024-00399817), and Alchymist Project Program (RS-2024-00423625) funded by the Korean government (MSIT and MOTIE), and grants from the DRB-KAIST SketchTheFuture Research Center, NAVER-intel, Adobe Research, Hyundai NGV, KT, and Samsung Electronics.

References

1. Abdelreheem, A., Eldesokey, A., Ovsjanikov, M., Wonka, P.: Zero-shot 3D shape correspondence. In: SIGGRAPH Asia (2023)
2. Abdelreheem, A., Skorokhodov, I., Ovsjanikov, M., Wonka, P.: SATR: Zero-shot semantic segmentation of 3D shapes. In: ICCV (2023)
3. Armeni, I., Sax, S., Zamir, A.R., Savarese, S.: Joint 2D-3D-semantic data for indoor scene understanding. arXiv preprint arXiv:1702.01105 (2017)
4. Baranchuk, D., Voynov, A., Rubachev, I., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models. In: ICLR (2022)
5. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In: ICCV (2019)
6. Cen, J., Zhou, Z., Fang, J., Yang, C., Shen, W., Xie, L., Zhang, X., Tian, Q.: Segment anything in 3D with nerfs. In: NeurIPS (2023)
7. Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3D: Learning from rgb-d data in indoor environments. In: 3DV (2017)
8. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An information-rich 3D model repository. arXiv preprint arXiv:1512.03012 (2015)
9. Chen, R., Liu, Y., Kong, L., Zhu, X., Ma, Y., Li, Y., Hou, Y., Qiao, Y., Wang, W.: CLIP2Scene: Towards label-efficient 3D scene understanding by clip. In: CVPR (2023)
10. Chibane, J., Engelmann, F., Tran, T.A., Pons-Moll, G.: Box2Mask: Weakly supervised 3D semantic instance segmentation using bounding boxes. In: ECCV (2022)
11. Chu, R., Chen, Y., Kong, T., Qi, L., Li, L.: ICM-3D: Instantiated category modeling for 3D instance segmentation. IEEE Robotics and Automation Letters (2021)
12. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In: CVPR (2017)
13. Decatur, D., Lang, I., Hanocka, R.: 3D Highlighter: Localizing regions on 3D shapes via text descriptions. In: CVPR (2023)
14. Hackel, T., Savinov, N., Ladicky, L., Wegner, J.D., Schindler, K., Pollefeys, M.: SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences (2017)
15. Hou, J., Dai, A., Nießner, M.: 3D-SIS: 3D semantic instance segmentation of rgb-d scans. In: CVPR (2019)

16. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: LoRA: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
17. Huang, R., Huang, C., Liu, Y., Dai, G., Kong, W.: LSGCN: Long short-term traffic prediction with graph convolutional networks. In: IJCAI (2020)
18. Huang, Z., Wu, X., Chen, X., Zhao, H., Zhu, L., Lasenby, J.: OpenIns3D: Snap and lookup for 3D open-vocabulary instance segmentation. arXiv preprint (2023)
19. Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.W., Jia, J.: PointGroup: Dual-set point grouping for 3D instance segmentation. In: CVPR (2020)
20. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023)
21. Koo, J., Huang, I., Achlioptas, P., Guibas, L.J., Sung, M.: PartGlot: Learning shape part segmentation from language reference games. In: CVPR (2022)
22. Lai, X., Tian, Z., Chen, Y., Li, Y., Yuan, Y., Liu, S., Jia, J.: LISA: Reasoning segmentation via large language model. arXiv preprint arXiv:2308.00692 (2023)
23. Lê, E.T., Sung, M., Ceylan, D., Mech, R., Boubekeur, T., Mitra, N.J.: CPFN: Cascaded primitive fitting networks for high-resolution point clouds. In: ICCV (2021)
24. Li, L., Sung, M., Dubrovina, A., Yi, L., Guibas, L.J.: Supervised fitting of geometric primitives to 3D point clouds. In: CVPR (2019)
25. Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.N., et al.: Grounded language-image pre-training. In: CVPR (2022)
26. Li, M., Xie, Y., Shen, Y., Ke, B., Qiao, R., Ren, B., Lin, S., Ma, L.: Hybridcr: Weakly-supervised 3d point cloud semantic segmentation via hybrid contrastive regularization. In: CVPR (2022)
27. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: PointCNN: Convolution on x-transformed points. In: NeurIPS (2018)
28. Li, Y., Ma, L., Zhong, Z., Cao, D., Li, J.: TGNet: Geometric graph cnn on 3D point cloud segmentation. IEEE Transactions on Geoscience and Remote Sensing (2019)
29. Liang, Z., Yang, M., Li, H., Wang, C.: 3D instance embedding learning with a structure-aware loss function for point cloud segmentation. IEEE Robotics and Automation Letters (2020)
30. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV (2014)
31. Liu, J., Yu, M., Ni, B., Chen, Y.: Self-prediction for joint instance and semantic segmentation of point clouds. In: ECCV (2020)
32. Liu, M., Zhu, Y., Cai, H., Han, S., Ling, Z., Porikli, F., Su, H.: PartSLIP: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In: CVPR (2023)
33. Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al.: Grounding DINO: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:2303.05499 (2023)
34. Liu, Y., Fan, B., Meng, G., Lu, J., Xiang, S., Pan, C.: DensePoint: Learning densely contextual representation for efficient point cloud processing. In: ICCV (2019)
35. Liu, Z., Qi, X., Fu, C.W.: One thing one click: A self-training approach for weakly supervised 3D semantic segmentation. In: CVPR (2021)
36. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In: ICLR (2022)

37. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: CVPR (2019)
38. Narita, G., Seno, T., Ishikawa, T., Kaji, Y.: Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In: IROS (2019)
39. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: OpenScene: 3D scene understanding with open vocabularies. In: CVPR (2023)
40. Pham, Q.H., Nguyen, D.T., Hua, B.S., Roig, G., Yeung, S.K.: JSIS3D: Joint semantic-instance segmentation of 3D point clouds with multi-task pointwise networks and multi-value conditional random fields. In: CVPR (2019)
41. Pruksachatkun, Y., Phang, J., Liu, H., Htut, P.M., Zhang, X., Pang, R.Y., Vania, C., Kann, K., Bowman, S.R.: Intermediate-Task Transfer Learning with Pretrained Language Models: When and why does it work? In: ACL (2020)
42. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: PointNet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
43. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
44. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: PointNext: Revisiting pointnet++ with improved training and scaling strategies. In: NeurIPS (2022)
45. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
46. Schult, J., Engelmann, F., Hermans, A., Litany, O., Tang, S., Leibe, B.: Mask3D: Mask transformer for 3D semantic instance segmentation. In: ICRA (2023)
47. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: OpenMask3D: Open-vocabulary 3D instance segmentation. In: NeurIPS (2023)
48. Tang, L., Jia, M., Wang, Q., Phoo, C., Hariharan, B.: Emergent correspondence from image diffusion. In: NeurIPS (2023)
49. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV (2019)
50. Vu, T., Kim, K., Luu, T.M., Nguyen, X.T., Yoo, C.D.: Softgroup for 3d instance segmentation on 3D point clouds. In: CVPR (2022)
51. Wang, W., Yu, R., Huang, Q., Neumann, U.: SGPN: Similarity group proposal network for 3D point cloud instance segmentation. In: CVPR (2018)
52. Wang, X., Liu, S., Shen, X., Shen, C., Jia, J.: Associatively segmenting instances and semantics in point clouds. In: CVPR (2019)
53. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM TOG (2019)
54. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., Lin, D., Liu, Z.: OmniObject3D: Large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In: CVPR (2023)
55. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. In: NeurIPS (2022)
56. Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., Liu, M., Jiang, H., Yuan, Y., Wang, H., Yi, L., Chang, A.X., Guibas, L.J., Su, H.: SAPIEN: A simulated part-based interactive environment. In: CVPR (2020)
57. Xiang, T., Zhang, C., Song, Y., Yu, J., Cai, W.: Walk in the Cloud: Learning curves for point clouds shape analysis. In: ICCV (2021)

58. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: CVPR (2023)
59. Xu, M., Ding, R., Zhao, H., Qi, X.: PAConv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: CVPR (2021)
60. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: PointANSL: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: CVPR (2020)
61. Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N.: Learning object bounding boxes for 3D instance segmentation on point clouds. In: NeurIPS (2019)
62. Yang, X., Wang, X.: Diffusion model as representation learner. In: ICCV (2023)
63. Yang, Y., Wu, X., He, T., Zhao, H., Liu, X.: SAM3D: Segment anything in 3D scenes. arXiv preprint arXiv:2306.03908 (2023)
64. Yi, K.M., Trulls, E., Ono, Y., Lepetit, V., Salzmann, M., Fua, P.: Learning to find good correspondences. In: CVPR (2018)
65. Yi, L., Huang, H., Liu, D., Kalogerakis, E., Su, H., Guibas, L.: Deep part induction from articulated object pairs. ACM TOG (2018)
66. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. In: CVPR (2019)
67. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-BERT: Pre-training 3D point cloud transformers with masked point modeling. In: CVPR (2022)
68. Zamir, A.R., Sax, A., Shen, W.B., Guibas, L.J., Malik, J., Savarese, S.: Taskonomy: Disentangling task transfer learning. In: CVPR (2018)
69. Zhang, B., Wonka, P.: Point cloud instance segmentation using probabilistic embeddings. In: CVPR (2021)
70. Zhang, J., Dong, R., Ma, K.: CLIP-FO3D: Learning free open-world 3D scene representations from 2D dense clip. arXiv preprint arXiv:2303.04748 (2023)
71. Zhang, R., Wang, L., Wang, Y., Gao, P., Li, H., Shi, J.: Parameter is not all you need: Starting from non-parametric networks for 3D point cloud analysis. In: CVPR (2023)
72. Zhang, Y., Ling, H., Gao, J., Yin, K., Lafleche, J.F., Barriuso, A., Torralba, A., Fidler, S.: DatasetGAN: Efficient labeled data factory with minimal human effort. In: CVPR (2021)
73. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: ICCV (2021)
74. Zhao, N., Chua, T.S., Lee, G.H.: Few-shot 3D point cloud semantic segmentation. In: CVPR (2021)