BiPart: Bi-level Optimization for Generalizable 3D Part Segmentation Prior Distillation from Pre-trained Vision-Language Model

Juan Fang
Shandong University
fangjuan.99@gmail.com

Jian Liu
Shenyang University of Technology
jianliu2006@gmail.com

Lei Wu
Shandong University
i_lily@sdu.edu.cn

Manyi Li*
Shandong University
manyili@sdu.edu.cn

Abstract

Recent works leverage pre-trained Vision-Language Models (VLMs) for the 3D part segmentation task to alleviate the need for human-annotated labels, but suffer from the multi-view inconsistency of VLM predictions. To address this issue, we propose a bi-level optimization algorithm to distill the generalizable 3D part segmentation priors from pre-trained VLM, which resolve the inconsistency among different views by integrating the vision priors into a unified 3D part segmentation model. Different from the common practice that distills the task-relevant priors as a whole, we carefully design the bi-level optimization algorithm and the 3D part segmentation model to disentangle the distilled priors into a category-agnostic feature extractor and a category-specific part segmentation head, where the latter can be quickly adapted to the unseen object category during inference via only a few iterations on a retrieved support set. We also propose a novel support set selection strategy, which retrieves relevant and beneficial shapes for the target object to enhance the fast adaptation performance. The experimental results demonstrate that our approach outperforms the existing methods on the 3D part segmentation task and achieves robust generalization performance on unseen object categories.

***Keywords:** 3D Part Segmentation, Shape Analysis, Vision-Language Model, Deep Learning*

1. Introduction

3D part semantic segmentation [14, 45] refers to the decomposition of an object into multiple semantically and functionally independent components. It involves the joint geometric and semantic analysis of diverse shapes to predict dense part labels. Most existing methods adopt a variety of annotated part segmentation datasets [2, 31, 68, 69] and design network architectures [16, 23, 27, 39, 55, 60, 74] to extract geometric and semantic features. However, due to the significant variations in part semantics, shapes, sizes, and functions across different object categories, current approaches still suffer from poor generalization capabilities.

Recent research has leveraged large pre-trained Vision-Language Models (VLMs) [19, 22, 34, 42] to address the generalization issue. The common practice [25, 72, 75, 76] is to render 3D objects as multi-view images, perform part localization or segmentation on the 2D images, and aggregate them to form 3D part segmentation results. These approaches enable open-vocabulary 3D part segmentation without requiring any 3D semantic annotations. However, the inconsistent semantic labels of multi-view images often lead to missing small parts and inaccurate segmentation results. On the other hand, some works [3, 28, 56] distill 3D priors by using VLM-generated 2D annotations as supervision signals, thus eliminating multi-view inconsistency during the segmentation inference. They also avoid the need for human-annotated part labels, but rely on a large number of 3D shapes to conduct cross-modal knowledge distillation, such as the per-category distillation in PartDistill [56] or the large-scale multi-category distillation in Find3D [28]. The high cost of cross-modal distillation hinders the segmentation model from quickly adapting to unseen object categories or segmentation at varying granularities.

In this paper, we propose a novel cross-modal distillation approach for generalizable 3D part segmentation. The key challenge is *how to distill a generalizable part segmentation prior from the pre-trained VLM to accommodate the unseen object categories*. To address this challenge, we carefully design the bi-level optimization algorithm and the 3D part segmentation model to disentangle the distilled priors into a category-agnostic feature extractor and a category-specific part segmentation head, where the latter can be quickly adapted to the unseen object category during infer-

*Corresponding Author.

ence via only a few iterations on a retrieved support set. The bi-level optimization conducts the cross-modal distillation with an outer loop and an inner loop, where the outer loop iterates over all training samples and optimizes the parameters of the class-agnostic module, and the inner loop leverages the retrieved support set for each target object to adapt the category-specific segmentation head to the target category. We also propose a support set selection strategy to retrieve relevant shapes based on a novel criterion, which effectively augments visual reference information for fast adaptation during inference and enhances the part segmentation accuracy for the target object.

Our method can effectively leverage the capabilities of pre-trained Vision-Language Models (VLMs) for generalizable 3D part segmentation. First, by distilling the visual priors of object parts into the 3D segmentation model, our method avoids inconsistencies caused by multi-view image understanding and obtains more accurate 3D segmentation results. Second, we disentangle the class-agnostic and class-specific part priors during cross-modal distillation, enabling our method to quickly adapt to unseen object categories with only a few iterations of the class-specific segmentation head during inference. Furthermore, we design a retrieval-augmented strategy to select a small support set for co-segmentation iterations of the category-specific segmentation head. This strategy effectively augments visual reference information for fast adaptation and enhances part segmentation accuracy for the target object. And it is flexibly suitable for different scenarios: it can effectively leverage few-shot examples to form the support set, or construct it with 3D generative models to fit the zero-shot setting.

Our contributions are summarized as follows:

- We propose a bi-level optimization approach to distill generalizable 3D part segmentation priors from pre-trained 2D VLM, which eliminates the need for large-scale human annotations while enabling a fast adaptation to unseen object categories.
- We propose a novel criterion for the support set selection during the optimization, which provides augmented references to further enhance the part segmentation accuracy during the fast adaptation.
- Extensive experiments demonstrate that our approach outperforms the existing methods on the 3D part segmentation task and achieves robust generalization performance on unseen object categories.

2. Related Works

2.1. Deep-learning-based 3D Part Segmentation

Deep-learning-based 3D part segmentation approaches represents 3D objects as voxels [27, 62], point clouds [23, 49, 61, 63], meshes [11, 16], etc., and design network architectures to extract dense features and predict part segmentation labels. Regarding the popular point cloud representation with inherent irregularity and unordered property, early approaches such as PointNet [39] and PointNet++ [40] process point clouds using multilayer perceptrons (MLPs) to encode individual points and max-pooling operations to ensure permutation invariance. PointCNN [23] and SpiderCNN [65] extend the conventional convolution operator to irregular point clouds. Further, PointConv [63] implements the continuous convolution via Monte Carlo approximation with density-aware weighting, while KPConv [55] employs deformable kernel points to dynamically fit local surfaces. On the other hand, some works construct varying neighborhoods from point clouds. For example, DGCNN [60], GAC [58], and PU-GCN [41] leverage dynamic graph neural networks to capture local geometric structures. Recently, Point Transformer [74], PCT [10] and Stratified Transformer [21] extract global and local geometric features via hierarchical self-attention mechanisms to enable adaptive feature aggregation. However, the aforementioned approaches heavily rely on the large-scale annotated datasets [31], causing poor generalization performance to unseen object categories that exhibit significant variation in part composition.

Due to the lack of large-scale part annotations, some works investigate weakly supervised or self-supervised learning to mitigate 3D data scarcity. For example, [4, 5, 33] leverages network bias of autoencoders designed for implicit field representation of 3D objects to achieve joint shape segmentation and reconstruction without any part annotations. But it is difficult to guarantee that the segmented parts are semantically meaningful. On the other hand, [48] leverages part annotations from a small dataset and adopts a co-segmentation paradigm to learn part segmentation priors. [59] constructs a small set of well-annotated part segmentations as templates and deforms the target shape to fit the templates to transfer part labels. Overall, the carefully annotated representative shape templates are still essential for weakly supervised methods, preventing them from quickly adapting to unseen categories.

2.2. VLM-based 3D Part Segmentation

With the emerging pre-trained Vision-Language Models (VLMs), recent studies have sought to leverage their open-vocabulary 2D understanding capability to facilitate 3D part segmentation and thereby overcome the lack of large-scale 3D part annotations. The common practice involves projecting 3D objects onto 2D images and then back-projecting the part grounding or segmentation results from these 2D projections to 3D objects. For example, PartSLIP [25] adopts GLIP [22] to detect part-specific bounding boxes from multi-view images and aggregates them

into 3D part segmentation results through superpoint grouping. Further, PartSLIP++ [75] integrates SAM [19] to generate precise pixel-wise 2D segmentation masks and proposes an Expectation-Maximization algorithm to convert multi-view 2D masks to 3D part segmentations through iterative refinement. Similarly, SATR [1] exploits the topological properties of meshes to produce fine-grained segmentation from multi-view images. And some other studies [9, 18, 28, 37, 52, 54, 67] attempt to use more diverse 2D priors [19, 34, 43] to infer 3D part segmentations. However, the content occlusion and inconsistent detections between multi-view images often cause incomplete and inaccurate part segmentation results.

On the other hand, some works use cross-modal knowledge distillation [26, 29, 44, 66, 73] to leverage pre-trained VLMs for 3D semantic segmentation tasks. The goal is to take advantage of the modality with rich-labeled data (i.e. images) to alleviate data shortage in the target modality (i.e. 3D point clouds). For example, OpenScene [36] directly aligns 3D point clouds and 2D images to the multimodal feature space of CLIP [42]. CLIP2Scene [3] transfers CLIP's pre-trained knowledge of 2D image-text alignment to a 3D point cloud network and enforces semantic and spatial-temporal consistency regularization. In terms of 3D part segmentation, PartDistill [56] proposes a bidirectional distillation framework that transfers category-specific knowledge from GLIP [22] to a 3D part segmentation model. Find3D [28] employs a data engine to construct large-scale synthetic part annotations with pre-trained VLMs and trains a 3D model using contrastive loss. Since all cross-modal distillation approaches require large-scale unannotated shape datasets for training, the core challenge remains how to quickly adapt to unseen object categories.

2.3. Fast Adaptation

Fast Adaptation [20, 24] aims to quickly adapt a pre-trained model to a new task or an unseen data domain. In the era of large models, parameter-efficient fine-tuning (PEFT) [13, 15] updates only a subset of a large model's parameters for fast adaptation, while maintaining the rest unchanged to retain the pre-acquired knowledge. In 3D understanding tasks, recent approaches often employ large models pre-trained on 3D data for feature encoding and append a small adapter for downstream tasks, such as OpenScene [36] and PartDistill [56]. However, they require a moderate-sized dataset to train the adapter from scratch, which is unavailable for unseen object categories in the open-vocabulary 3D part segmentation task.

On the other hand, the goal of Fast Adaptation aligns closely with the "learning to learn" paradigm of meta-learning [17], which learns prior knowledge to guide the model in learning new tasks efficiently. Among the related works [7, 8, 30, 32, 47, 50, 57], a widely adopted algorithm is Model-Agnostic Meta-Learning (MAML) algorithm [7], which learns a universal parameter initialization through the bi-level optimization. This learned initialization serves as a strong prior, allowing the model to rapidly converge on unseen tasks with only a few iterations. Therefore, this algorithm has been applied to shape reconstruction [46, 51] and segmentation [12] to quickly adapt to unseen object instances. However, the cross-category generalization remains challenging due to large variation among part compositions of different categories.

By contrast, our approach takes the merits of both directions for fast adaptation. Our model starts with a frozen pre-trained model for point cloud feature encoding, which effectively narrows down the cross-modal domain gap between 3D objects and their rendered images. We then adapt the bi-level optimization of MAML to disentangle class-agnostic priors and class-specific initialization priors into different modules so that it can quickly adapt to unseen object categories with only a few iterations at inference time.

3. Method

3.1. Overview

Our approach aims to distill generalizable 3D part segmentation priors from pre-trained 2D VLM for point clouds. This goal necessitates the distillation approach to hold two key characteristics. First, it should learn the general 3D segmentation priors that are irrelevant to specific object categories to facilitate generalization. Second, it should be capable of rapidly adapting to the semantic segmentation of unseen object categories with low time and data costs.

In this work, we propose the bi-level optimization algorithm and 3D part segmentation model to disentangle the category-agnostic and category-specific priors during cross-modal distillation. The 3D part segmentation model is composed of three modules to learn the priors with different levels of generalization and is trained with the VLM-generated predictions, as described in Section 3.2. During the cross-domain distillation, the bi-level optimization conducts the outer loop to iterate over all training samples and optimize the network parameters of the 3D part segmentation model. Within the outer loop, the inner loop initializes the segmentation head module with current parameters and performs a few iterations to quickly adapt to the semantic category of each target shape. The training and inference processes of the cross-modal distillation are introduced in Section 3.3. To further enhance the part segmentation accuracy on unseen categories, we retrieve a support set for each target shape to assist the iterations of the inner loop for the fast adaptation during inference. A novel criterion for the support set selection is proposed to ensure relevant and diverse visual reference information, as described in Section 3.4.

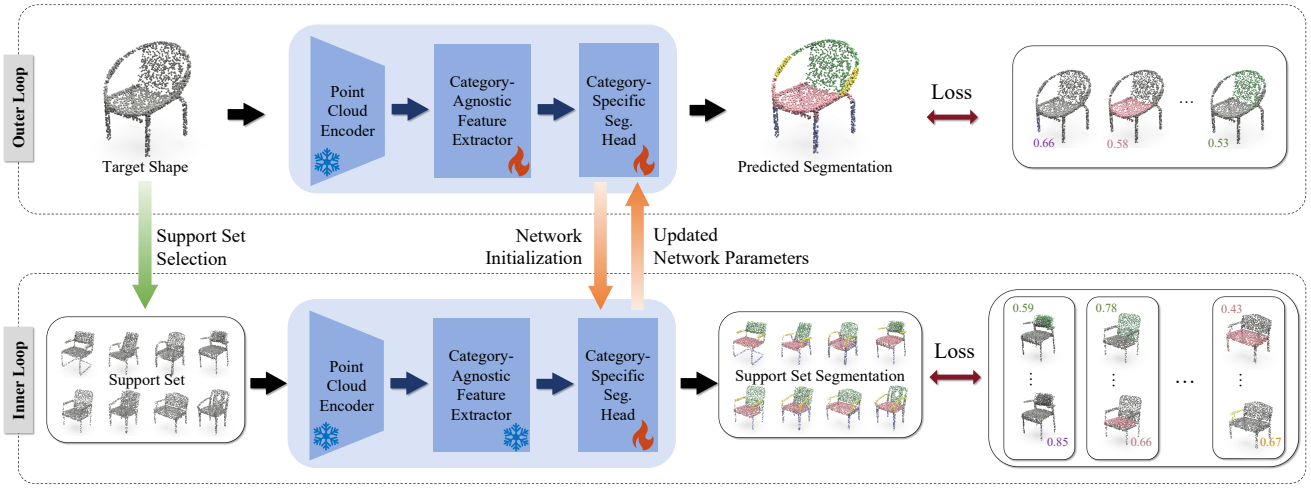

Figure 1: The bi-level optimization pipeline of our approach. The outer loop iterates over each training sample and optimizes the parameters of the class-agnostic feature extraction module. Within each iteration of the outer loop, we select the support set for the target shape and optimize the category-specific segmentation head module with a few iterations to adapt to the category of the target shape. This bi-level optimization enables us to disentangle the category-agnostic and category-specific priors during cross-modal distillation, thus generalizable to unseen categories during inference.

3.2. 3D Part Segmentation Model

Given the point cloud of the target object and the associated part labels, the 3D part segmentation model aims to predict the part probability for each point. Our model consists of three modules. The first is a point cloud encoder, which is pre-trained on a large-scale dataset and frozen throughout our framework. Its role is to extract per-point local features for the input point cloud. The local features are then concatenated with the original point cloud and fed into the second module, i.e. category-agnostic feature extractor. This extractor is implemented as two consecutive Point Transformer blocks [74], which employ the self-attention mechanism over 16-nearest neighbors for each point with trainable relative position encoding, along with the residual connections and pointwise MLPs to refine the features. Finally, the third module, i.e. the segmentation head, is implemented as a two-layer MLP which takes the extracted features as input and predicts the pointwise part probabilities as the final output. In summary, the network processing can be formulated as

$$\hat{Y} = f(S, \pi, \theta, \varphi), \tag{1}$$

where S is the input point cloud. π, θ, and φ denote the network parameters of the encoder, feature extractor, and the segmentation head modules. $\hat{Y}$ represents the predicted pointwise part probabilities.

Our 3D part segmentation model is supervised by the VLM-annotated dataset, eliminating the need for the human-annotated part labels. We adopt the distillation loss proposed in [56]. Specifically, for each point cloud in the training set, we render it into multi-view images, then use the pre-trained GLIP [22] to detect the part bounding boxes in each image. For example, assuming the point cloud $S \in \mathbb{R}^{N\times 3}$ of an object with R semantic parts and D bounding boxes detected from V rendered images, we can apply the 2D-to-3D back-projection to construct the pseudo ground-truth labeling $C = \{(\mathcal{M}^d, Y^d)\}_{d=1}^{D}$. Here $\mathcal{M}^d \in \{0,1\}^N$ is the mask indicating whether a point is associated with the dth detected bounding box, and $Y^d \in \mathbb{R}^{N\times R}$ denotes the pseudo ground-truth probability of each point belonging to each of the R semantic parts. Note that the validity of Y^d is indicated by $\mathcal{M}^d$: points masked out by $\mathcal{M}^d$ have the probability of being zero for all part labels and the rest points share the part probability of this bounding box. Hence, the distillation loss can be formulated as

$$L(\hat{Y}, C) = -\sum_{d=1}^{D} \frac{1}{|\mathcal{M}^d|} \sum_{n=1}^{N} \sum_{r=1}^{R} \mathcal{M}_n^d H_n^d Z_{n,r}^d \log\left(\hat{Y}_{n,r}\right), \tag{2}$$

where $\hat{Y}$ is the predicted part probabilities of the input point cloud. $H_n^d = \max_r \left(Y_{n,r}^d\right)$ indicates the highest ground-truth part probability of the dth bounding box, and $Z_{n,r}^d$ indicates whether the rth part is the most probable part of the dth bounding box. In this way, although the VLM-processed pseudo labels are noisy and inconsistent, this distillation loss aligns the model with high-confidence labels to learn the 3D part segmentation prior from the large-scale 3D shape dataset.

3.3. Bi-Level Optimization Training Strategy

Algorithm 1 The training process of the proposed method

Input: $p(\mathcal{T})$: data distribution of training set;
α: inner-loop learning rate;
β: outer-loop learning rate;
K: inner-loop iteration number;
π: network parameter of point cloud encoder;
M: support set size.
Output: θ: network parameter of feature extractor;
φ: network parameter of segmentation head.

Randomly initialize θ and φ;
while not done **do**
  // $\mathcal{T}_i$ denotes the 3D shape and C_i is the VLM-based pseudo ground-truth
  Sample batch of training data $(\mathcal{T}_i, C_i) \sim p(\mathcal{T})$;
  $L_{outer} \leftarrow 0$;
  // The start of outer loop
  for all $(\mathcal{T}_i, C_i)$ **do**
    Retrieve the support set $(S_{i,j}, C_{i,j})_{j=1}^{M}$;
    $\varphi_0 \leftarrow \varphi$;
    // The start of inner loop
    for $k = 1, 2, \cdots, K$ **do**
      // The loss for all the shapes in the support set
      $L_{inner} \leftarrow \sum_{j=1}^{M} L\left(f(S_{i,j}, \pi, \theta, \varphi_{k-1}), C_{i,j}\right)$;
      Update $\varphi_k \leftarrow \varphi_{k-1} - \alpha \nabla_{\varphi_{k-1}} L_{inner}$;
    end for
    $L_{outer} \leftarrow L_{outer} + L\left(f(\mathcal{T}_i, \pi, \theta, \varphi_K), C_i\right)$;
  end for
  Update $\theta \leftarrow \theta - \beta \nabla_{\theta} L_{outer}$;
  Update $\varphi \leftarrow \varphi - \beta \nabla_{\varphi} L_{outer}$;
end while
return θ, φ;

The training stage utilizes all the shapes across diverse categories in the training set. Due to variations in part composition among different object categories, existing 3D part segmentation networks often predict probability vectors with a dimension equal to the total number of parts in the dataset. This hinders the generalization to unseen object categories: the network cannot dynamically extend the pre-defined part label collection during inference.

To tackle this issue, we set the segmentation head to predict the probability vectors with a dimension equal to the maximum number of parts across all categories in the dataset. The network parameters of the segmentation head can dynamically adapt to the semantic labels of the object category corresponding to each shape. This leads to our bi-level optimization algorithm during training.

3.3.1 Training Algorithm

Algorithm 1 illustrates the bi-level optimization algorithm for the training, given the training set composed of the 3D shapes and their corresponding VLM-generated pseudo ground-truth labeling. The point encoder is initialized with a pre-trained model for point clouds and frozen during training. In our implementation, we use the multi-scale masked autoencoders Point-M2AE [71]. The feature extractor and segmentation head are randomly initialized. Then we sample the data batches from the training set and conduct the bi-level optimization for each batch. During the optimization, the outer loop enumerates all the data in this batch and update the network parameters based on their losses, while the inner loop adapts the segmentation head to the object category of each training data via a few iterations.

Outer Loop. In each iteration of the outer loop, we evaluate the segmentation results with the current network and update the network parameters based on the loss function. Specifically, for each training data in the batch, the outer loop retrieves relevant shapes from the training set to form its support set, which is described in Section 3.4. Note that the support set includes the training data itself. Then we conduct an inner loop, which takes the current segmentation head as initialization φ_0 and iteratively updates it K times based on the support set to obtain φ_K. After the inner loop, we use the current network with the updated segmentation head φ_K to compute the predicted segmentation and the corresponding loss, denoted as L_{outer}, to train the 3D part segmentation model.

Inner Loop. For each training data, the inner loop quickly adapts the segmentation head based on its support set via a few iterations. Specifically, in the kth iteration of the inner loop, it enumerates all the shapes in the support set and computes the predicted segmentation with the current network parameter. The distillation loss regarding the predicted segmentations is computed and used to update the segmentation head only, with the rest of the network parameters unchanged throughout the inner loop. In summary, each iteration in the inner loop can be formulated as

$$\begin{aligned} L_{inner} &= \sum_{j=1}^{M} L\left(f(S_{i,j}, \pi, \theta, \varphi_{k-1}), C_{i,j}\right), \\ \varphi_k &\leftarrow \varphi_{k-1} - \alpha \nabla_{\varphi_{k-1}} L_{inner}, \end{aligned} \tag{3}$$

where $S_{i,j}$ and C_{ij} denote the jth example in the support set of the ith training data and its associated pseudo labels.

After the bi-level optimization, we obtain the trained network parameters of the 3D part segmentation model. The feature extractor learns the category-agnostic segmentation prior since it is trained with the 3D shapes from diverse object categories. The trained segmentation head acts as a good initialization for the fast adaptation to unseen categories, since we conduct a separate inner loop for each training data during training.

Figure 2: The selected support sets (right) with different strategies for the target shape (left). The color indicates ground-truth segmentation for a better visualization purpose.

3.3.2 Inference Algorithm

Given a 3D shape from the test set, we can retrieve the support set from the training set, similar to the training process. Since the pseudo ground-truth data is generated by the pre-trained VLM without human annotation, we produce the pseudo ground-truth for the test shape as well and include it in the support set. Then we take the 3D part segmentation model with the trained network parameters, and perform an inner loop optimization based on the support set to update the segmentation head. After the inner loop, we use the optimized segmentation head to predict the segmentation labels of the test shape as the final output.

3.4. Support Set Selection

During both the training and testing, for each target shape to be segmented, we retrieve relevant shapes to form the support set that serves in the inner loop for the fast adaptation of the segmentation head. The support set consists of the retrieved 3D shapes and their VLM-generated pseudo ground-truth labelings, whose selection influences the part segmentation result.

A straightforward strategy is random selection, where we randomly select M 3D shapes from the training set as the support set for each target shape. Another alternative strategy is the similarity-based selection. That is, we compute the BPS feature [38] of the target shape and the shapes in the training set. Then we select M training shapes based on the cosine similarity of these BPS features.

We additionally propose a more powerful strategy named part-perfect selection. Since the shape instances in one category may exhibit different part composition, e.g. some chair contain armrest while some not, and the role of the support set is to enhance the category-specific prior of the segmentation head, the shapes in the support set should provide the information of diverse parts as much as possible. Therefore, we collect all "part-perfect" shapes, i.e. the shapes with maximum part instances across this category, and select shapes from this collection.

Specifically, the criterion for the "part-perfect" shapes is defined as follows:

$$\text{ratio} = \left\{ \sum_{n=1}^{N} W_{n,r} \cdot \left[W_{n,r} = \max_{k} W_{n,k} \right] \right\}_{r=1}^{R} \tag{4}$$

where $W_{n,r} = \sum_{d=1}^{D} Y_{n,r}^{d}$ estimates the probability of the nth point corresponding to part $\mathbf{r}$, $[\cdot]$ is the Iverson bracket (1 if the condition holds, 0 otherwise). This ratio represents the point probabilities of each part, i.e. a rough estimation of the point number for each point. Then we normalize the ratio and filtering out the parts with probabilities below 0.01. The left shapes are collected as the "part-perfect" shapes. For each target shape, we compute the BPS feature similarity and select the most similar shapes from the "part-perfect" collection as the support set. Moreover, if the number of "part-perfect" shapes is smaller than the predefined support set size, the remaining slots of the support set are filled via similarity-based selection from the "non-part-perfect" shapes. Figure 2 shows the selected support sets with these strategies.

4. Experiments

4.1. Experiment Settings

Datasets. We run experiments on ShapeNetPart [68] and PartNetE [25]. ShapeNetPart contains 16619 shapes from 16 categories, split into 11955/1844/2820 samples

Table 1: Quantitative evaluation on the ShapeNetPart dataset, reported in mIoU(%). PartDistill-J and Ours-J denote the joint training on all categories, while PartDistill-G and Ours-G are the generalization setting trained only on the seen categories.

Method	Airpl	Car	Chair	Lamp	Table	Bag	Cap	Earph	Guitar	Knife	Laptop	Motor	Mug	Pistol	Rocket	Skate	Overall
Find3D	15.42	15.64	31.68	25.42	51.56	10.50	14.00	57.41	30.71	27.67	27.74	12.12	19.72	22.83	22.85	58.00	33.40
PointCLIPv2	33.45	27.21	51.55	44.68	61.14	60.36	52.90	56.52	71.45	76.72	61.53	31.48	48.00	46.07	49.58	43.90	51.79
PartSLIP	52.99	14.82	70.41	43.02	58.63	47.04	0.00	12.29	53.33	46.66	30.97	19.68	18.81	3.64	7.18	20.81	52.28
PartSLIP++	30.60	7.18	54.63	24.34	40.86	33.20	0.00	0.00	29.49	48.21	15.49	10.16	9.66	2.31	5.68	2.47	35.86
PartDistill-J	40.53	8.48	56.33	43.32	38.23	45.25	20.02	11.89	51.89	64.09	57.34	23.71	53.22	8.68	27.19	16.08	42.99
Ours-J	66.18	28.90	82.56	45.66	54.93	66.64	20.85	30.37	86.99	84.40	92.36	31.57	42.73	29.08	45.25	12.61	**62.90**
	Seen Categories					Avg	Unseen Categories										Avg	
PartDistill-G	47.04	10.87	58.52	43.39	34.78	43.17	15.51	27.31	12.10	15.66	11.38	24.95	8.30	31.96	12.75	21.16	6.46	16.31	38.42
Ours-G	67.72	29.98	84.37	46.24	52.56	62.12	78.01	23.28	19.16	73.47	79.84	89.16	20.08	52.24	22.58	44.53	10.81	60.47	<u>61.83</u>

for train/val/test. PartNetE includes 2266 shapes covering 45 categories, which are collected from PartNet [31] and PartNet-Mobility [64] datasets. We split the PartNetE dataset with a 7:3 ratio, with 1562/704 samples for train/test. Additionally, we construct a validation set of 338 shapes by selecting 8 samples per category from the training set or all samples if fewer than 8 exist.

Baselines. We compare with existing approaches on open-vocabulary 3D part segmentation, namely PointCLIP v2 [76], PartSLIP [25], PartSLIP++ [75], Find3D [28], PartDistill [56]. All these methods are training-free or trained with VLM-generated annotations without human labeling. In our experiments, we use their official implementation. For PartDistill, which distills category-level part segmentation prior from the pre-trained VLM, we retrain it on the same training and test sets as ours using their default setting. We adopt the mean intersection over union (mIoU) to evaluate the segmentation results.

Implementation. Our framework is implemented with PyTorch [35]. During training, the outer loop utilizes the Adam optimizer with a learning rate of 1×10^{-4}. It uses a batch size of 8 and trains the network for 10 epochs. The inner loop performs 5 iterations to obtain the final segmentation results, with a learning rate of 0.1 for the gradient descent. The training takes approximately 14 hours with a single NVIDIA GeForce RTX 4090 GPU. During inference, we execute the inner loop with 10 iterations with a learning rate of 0.3, for a better adaptation to the target shapes. The default size of support set is 8 during both the training and inference. For each 3D shape, we render the raw object (dense point cloud for PartNetE and mesh for ShapeNetPart) into RGB images from 10 fixed viewpoints around the object similar to [25].

4.2. Comparisons

Since both PartDistill and ours distill the 3D part segmentation prior from the pre-trained VLM with a collection of training shape, we design two variants to better validate the strength of our approach. One is the joint training on all the categories and test on the unseen shapes of the same categories, denoted as PartDistill-J and Ours-J. The other is the generalization setting which trains on some selected categories, i.e. those with more shape instances among all the categories, while the test shapes come from both the seen and unseen categories, denoted as PartDistill-G and Ours-G. For both settings with our approach, we select the support sets from the training set. The training samples of unseen categories in the generalization experiments are only used during inference, to form the support sets for the fast adaptation. Note that the other approaches are designed for generalization purpose, so we directly employ their default setting to test all the categories.

Comparisons on ShapeNetPart. Figure 3 shows qualitative results on the ShapeNetPart dataset. Among these approaches, PartSLIP and PartSLIP++ frequently fails to assign points to certain parts during segmentation (shown as gray dots in the figure), which may be caused by GLIP's inability to recognize the object or its low confidence for some bounding box predictions. On the other hand, although PartDistill performs relatively well compared to PartSLIP in joint training, in the generalization setting, PartDistill performs poorly on unseen objects and often fails to produce meaningful segmentations. In contrast, our method not only achieves suerior performance in joint training but also maintains robust segmentation accuracy on unseen object categories during generalization testing. Furthermore, our method excels in segmenting the fine-grained components, such as armrests of chairs, heads/tuners of guitars, and nose cones of rockets, demonstrating superior capability in handling small or intricate parts.

Table 1 reports the quantitative results on ShapeNetPart. Our approach consistently outperforms the other methods, especially exhibit significant improvement compared to the most relevant approach PartDistill. For the joint training,

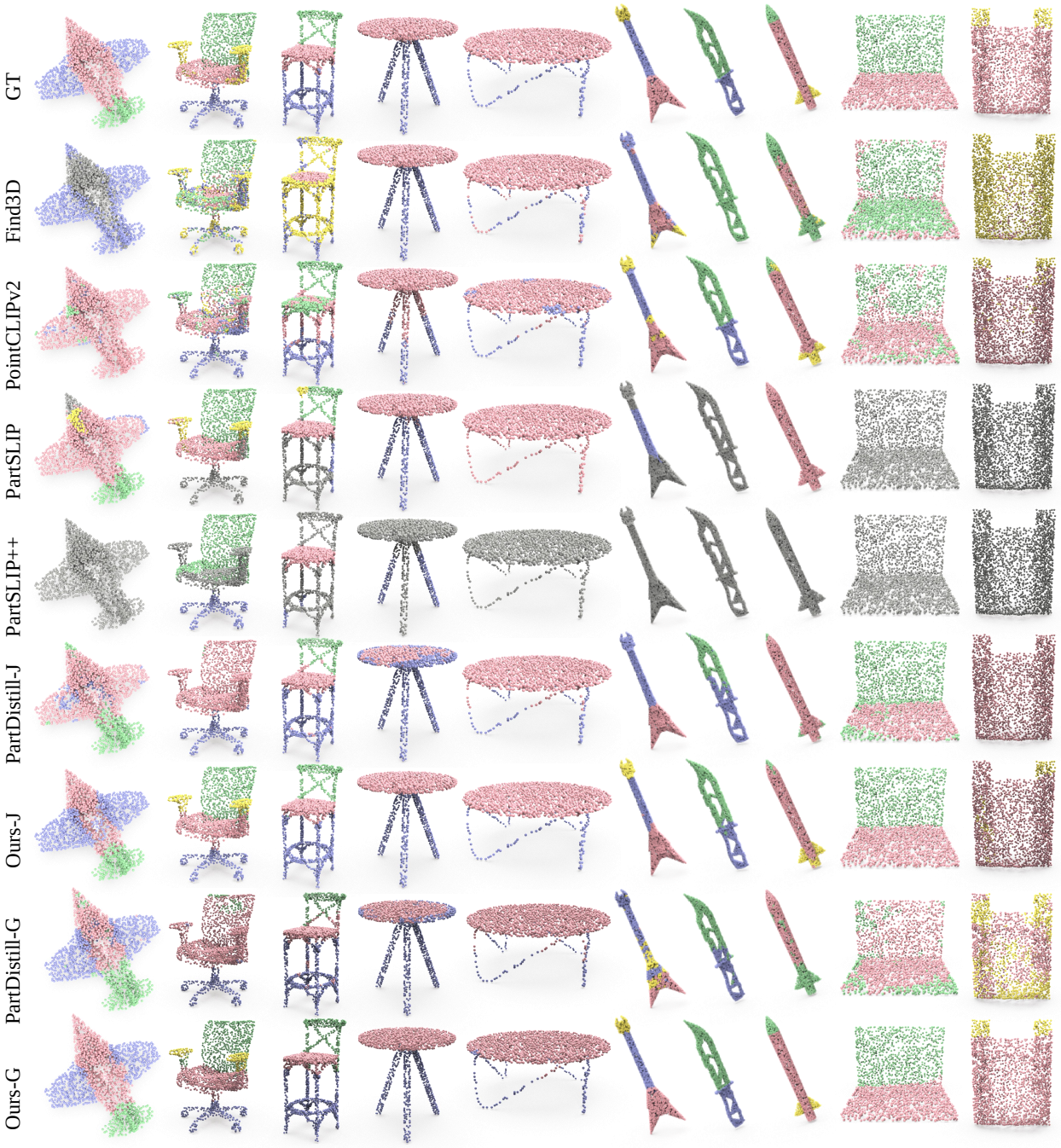

Figure 3: Qualitative comparison between our method and other approaches on ShapeNetPart dataset. For the last two rows with the generalization experiment setting, the left five are from seen categories, while the rest are from unseen categories.

our method achieves 62.9% mIoU on ShapeNetPart, outperforming PartDistill by 19.91%. In the generalization testing, the segmentation model is trained on only five categories (Airplane, Car, Chair, Lamp, Table) from ShapeNetPart, which contains 16 categories in total. Despite this, our method achieves 60.47% mIoU on unseen shapes, outperforming PartDistill by 44.16%.

Comparisons on PartNetE. Table 2 reports the quantitative results on the PartNetE dataset. Our approach achieves the best performance with the joint training setting. The generalization setting of our approach obtains competitive performance compared to PartSLIP, but still significantly outperforms the other methods. This is because the network-based approaches, i.e. Find3D, PartDistill, and

Table 2: Quantitative evaluation on the PartNetE dataset, reported in mIoU(%). PartDistill-J and Ours-J denote the joint training on all categories, while PartDistill-G and Ours-G are the generalization setting trained only on the seen categories.

Method	Cart	Chair	Eyegl	Switch	Table	Bottle	Box	Knife	Lamp	Micro	Print	Remote	Safe	Sciss	Stapl	USB	Overall
Find3D	20.07	21.40	31.60	25.48	18.27	13.10	18.14	12.12	15.17	0.21	0.12	5.42	0.59	12.71	20.67	11.67	13.41
PointCLIPv2	12.27	15.92	15.26	8.77	3.50	13.61	35.68	26.65	8.34	2.05	0.21	8.70	3.67	6.35	20.53	12.51	12.69
PartSLIP	79.31	75.26	7.14	10.40	42.49	81.08	59.87	24.73	31.79	18.64	16.11	7.16	7.00	61.73	26.44	32.13	35.14
PartSLIP++	29.22	40.85	9.49	0.00	16.49	39.91	37.00	24.24	27.84	0.00	0.00	0.00	2.74	50.87	22.95	12.04	15.60
PartDistill-J	67.66	12.27	13.26	53.10	48.45	68.68	62.30	74.22	32.54	25.06	52.36	46.08	26.18	25.01	19.60	25.39	28.94
Ours-J	73.70	74.85	77.45	46.36	46.32	78.56	81.20	72.88	50.78	29.80	55.45	46.04	24.49	58.11	44.37	30.54	**35.52**

	Seen Categories					Avg	Unseen Categories										Avg	
PartDistill-G	72.82	30.58	13.26	48.69	54.86	32.04	20.45	36.42	36.24	27.68	29.63	46.28	43.22	26.18	25.01	19.60	25.39	22.64	26.90
Ours-G	77.17	76.53	85.94	53.47	50.83	40.65	77.37	79.86	50.32	51.73	29.68	55.45	46.32	19.97	49.66	32.43	25.93	30.52	35.11

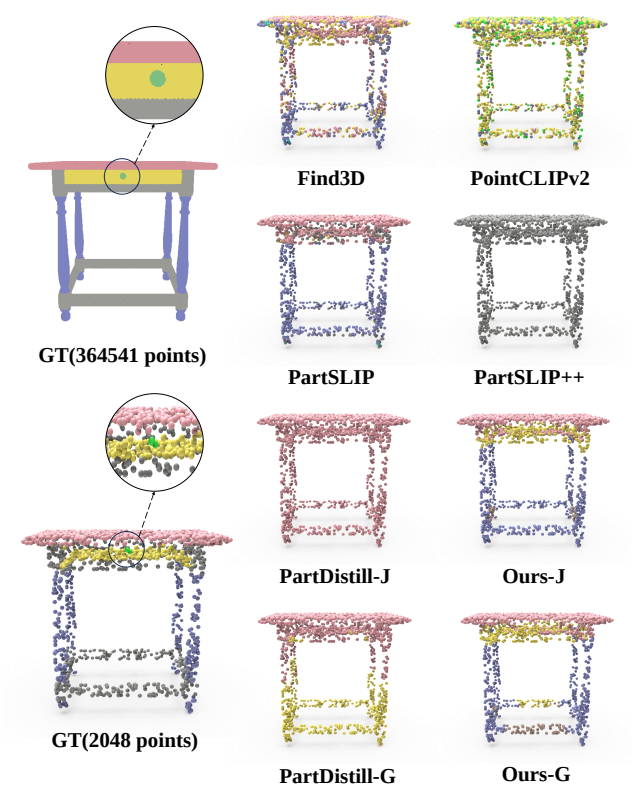

Figure 4: Qualitative comparison between our method and other approaches on the PartNetE dataset. Our method often fails to distinguish the small parts with the sparse point cloud as the input of the network, yet still outperforms most of the existing methods.

ours, allocate the 2048 point clouds as inputs, while this resolution is insufficient to reflect the shape feature of the small-scale parts, which are typical cases in PartNetE. For example, as illustrated in Figure 4, the handle of the table only possesses three points (green dots) in the input point cloud, which is difficult for all the methods to capture this small-scale part. This particularly affects the segmentation models for point cloud representation, since it causes insignificant geometric features from the input point clouds, making it more challenging for the networks to distinguish these small components.

Table 3: Ablation study on the bi-level optimization of the proposed method, reported in mIoU(%).

Exp. Id	Support Set	Bi-level Training	C-agnostic Feature	Ours-J	Ours-G	
					Seen	Unseen
1	×	×	×	52.52	51.99	12.85
2	✓	×	×	55.65	49.92	37.18
3	✓	✓	×	62.20	61.95	56.19
4	✓	✓	✓	62.90	62.12	60.47

Table 4: Ablation study on the support set selection strategy of the proposed method, reported in mIoU(%).

	Ours-J	Ours-G	
		Seen	Unseen
Random Selection	54.53	54.38	44.55
Similarity-based Selection	55.22	55.56	49.22
Part-Perfect Selection	62.90	62.12	60.47

4.3. Ablation Studies

The ablation study evaluates two key components of our approach on the ShapeNetPart dataset. One is the bi-level optimization, which learns category-agnostic segmentation prior and enables fast adaptation to the semantic part labels of unseen categories. The other is the "part-perfect" criterion for the support set selection to enhance the adaptation. We use the averaged mIoU across the entire test set for the quantitative evaluation.

Bi-Level Optimization. We compare four variants of our

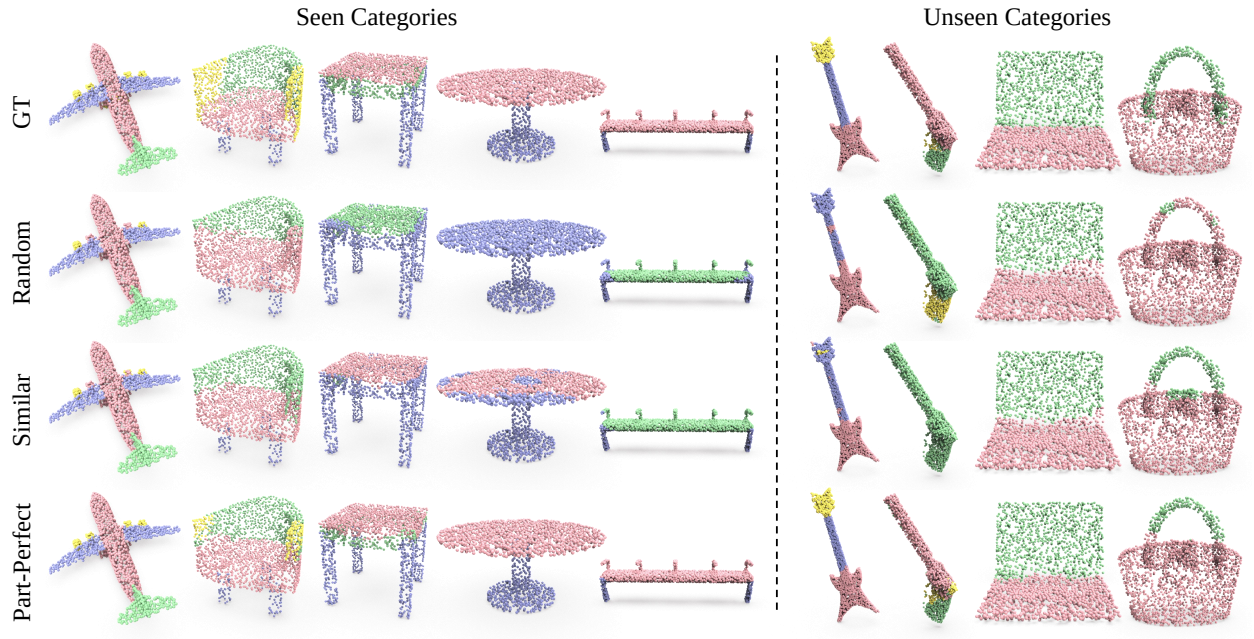

Figure 5: The qualitative results of the ablation study with different support selection strategies on the ShapeNetPart dataset.

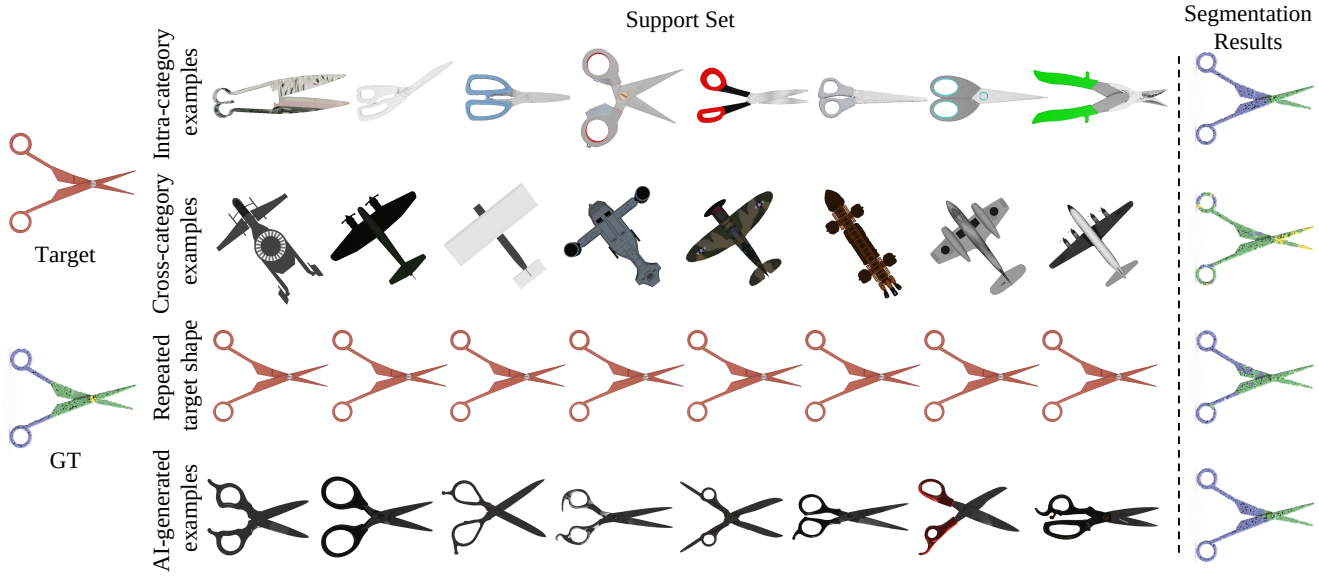

Figure 6: For a specific target shape (left), we collect examples from different sources to form the support sets (middle) and compute the part segmentation results (right). It validates the necessity of intra-category examples (first row) and the effectiveness of the AI-generated examples (fourth row).

approach as illustrated in Table 3. It starts with our part segmentation model without the category-agnostic feature extractor module, i.e. the frozen point cloud encoder directly connected with a trainable segmentation head. In the first experiment, we do not use the bi-level optimization at all, making it similar to PartDistill but with different network architectures. Apparently, it causes relatively poor part segmentation results, especially when generalized to unseen categories. The second experiment doesn't use the bi-level optimization as well, but leverages the support set for a test-time fine-tuning to enhance the segmentation result. The effect is obvious in the generalization setting, where the performance on unseen categories increases because of the fine-tuning. But the performance on the seen category de-

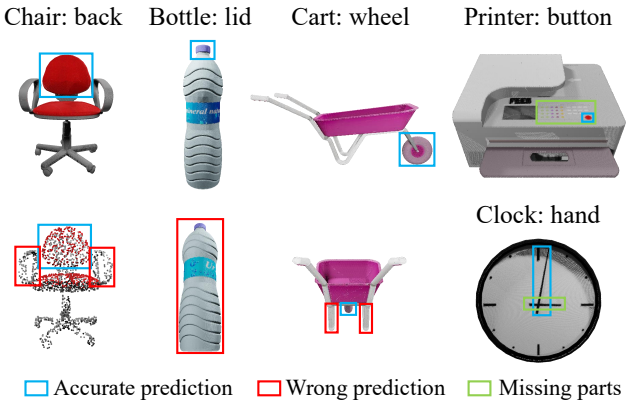

Figure 7: Visualization of VLM-generated annotations on different shapes.

creases because the fine-tuning affects the learned part segmentation prior from the training set. In the third experiment, we employ the bi-level optimization and the proposed support set selection. The only difference with our full approach (the fourth experiment) is that the latter is equipped with a feature extractor module, which learns the category-agnostic prior during training and remains unchanged during inference. By comparing the third and fourth experiments, it proves that our design successfully disentangles the category-agnostic and category-specific priors to maximally leverage the knowledge from pre-trained 2D VLM for the 3D part segmentation task.

Support Set Selection We compare the three support set selection strategies in Table 4. The similarity-based selection performs slightly better than the random selection, and the improvement is more obvious on the unseen categories in the generalization setting. The reason is that similar shapes provide more specific segmentation information that is relevant to the target shape, and the VLM-generated pseudo labels for these similar shapes provide cross-validation among them. Additionally, the part-perfect selection significantly outperforms the other two strategies, since the "part-perfect" shapes balance the part completeness and shape similarity, thus providing more abundant information to guide the adaptation to the target shape. Figure 5 shows the qualitative results of Ours-G, which validates the effectiveness of the part-perfect strategy in capturing the less-frequent parts such as the armrest of chairs.

5. Discussion

5.1. Support Set from Different Sources

Our approach leverages the VLM-generated pseudo labels of the support set to infer the segmentation of the target shape. This raises the question for unseen categories: what if we don't have enough examples for a novel object category to form the support set?

Table 5: Hyper-parameter Setting on the proposed method, reported in mIoU(%).

(a) Size of Support Set

Size of Support Set	Ours-J	Inference Time
1	59.15	0.58s
2	61.69	0.64s
4	62.61	0.78s
8	**62.90**	1.12s
12	62.72	1.31s

(b) Number of Iterations

Number of Iterations	Ours-J	Inference Time
5	61.38	0.62s
10	62.90	1.12s
15	62.75	1.64s
20	63.19	2.13s

We first compare the part segmentation results with support sets from the same or different categories, as shown in Figure 6 (1) and (2). As expected, with intra-category examples in the support set (the first row), it successfully segments the two main parts of the scissor. By contrast, the cross-category examples (the second row) provide distracting information since different categories don't share the same set of part labels, resulting in poor segmentations.

For a novel category without extra shapes to serve the support set, we propose two solutions. A straightforward solution is to repeat the target shape and the associated VLM-generated pseudo labels multiple times as the support set. This is equivalent to a quick fine-tuning with the VLM annotations. As shown in Figure 6 (3), it results in semantically meaningful segmentation. But the limitation is that it may overfit to the noisy annotations produced by the pre-trained 2D VLM, especially when there exist large regions with incorrect part labels.

We propose a more practical solution, which makes use of the 3D content generation approaches to construct the support set. We employ the Hunyuan3D 2.0 model [53] to generate multiple 3D shapes with the object category name as input prompt. Figure 6 (4) shows the support set and the corresponding part segmentation of the target shape. It indicates that the state-of-the-art 3D generative models are capable of generating diverse shapes of the target category to form the support set, enabling our approach for any open-vocabulary 3D part segmentation task.

5.2. VLM-Generated Annotations on Different Shapes

We analyze the performance of GLIP since it provides pseudo-labels during both training and inference. Empirically, GLIP's part detection results are influenced by rendering quality, viewpoint selection, and the inherent characteristics of the shapes themselves. Figure 7 illustrates the impact of these factors. The left column indicates that the model performs better on dense, colored point cloud renderings, which resemble the natural images in its training set. The middle two columns show that part detection improves when the shape is posed to fully expose the relevant parts. As in the bottom row, a bottom-up view causes the model to fail in grounding the lid of the bottle. Finally, the right column demonstrates that GLIP often ignores some small-sized parts, especially when there are multiple instances belonging to the same part category.

This leads to our rendering setting as described in Section 4.1. It is worth noting that, as many objects in the datasets lack texture information, they are rendered in a default gray color. A potential improvement involves rendering depth images instead and converting them to RGB images using image-to-image models such as ControlNet [70].

5.3. Hyper-Parameter Selection

We analyze the influence of two key hyper-parameters in the inner loop, i.e. the size of the support set and the number of iterations. As quantitatively evaluated in Table 5, our approach converges to satisfactory segmentation results with a support set of 8 examples. And 10 iterations achieve a desirable trade-off between the segmentation accuracy and inference time.

6. Conclusion

In this work, we present BiPart, a bi-level optimization approach to distill generalizable prior knowledge for 3D part segmentation from pre-trained 2D VLMs. With the carefully-designed modules of the 3D part segmentation model and the proposed bi-level optimization algorithm, we can disentangle the category-agnostic and category-specific priors into different modules during the cross-modal distillation. It enables fast adaptation to the unseen object categories during inference via a few iterations with a retrieved support set, thus facilitating the open-vocabulary 3D part segmentation. We also propose practical strategies to select the relevant and beneficial support set for each target shape. Experimental results demonstrate that our approach achieves robust segmentation performance in terms of generalization and outperforms other state-of-the-art open vocabulary part segmentation methods.

Our approach also holds some limitations. First, the cross-modal distillation requires a large-scale dataset with diverse 3D shapes as the training set to capture the category-agnostic prior. Training on a large-scale dataset with more diverse 3D shapes, such as Objaverse-XL [6], would further enhance the generalization performance of our approach. Second, since our 3D part segmentation model takes the input point clouds with a point number of 2048, it prevents the network from learning the geometric features from sparse point clouds for the small parts. Incorporating the network architectures with more suitable 3D representations would alleviate this issue and thus improve the part segmentation performance.

In addition, 3D part segmentation in the real-world scenarios involves multi-level, fine-grained part decomposition. For example, in the challenging ABO dataset, which provides photo-realistic 3D models and fine-grained part annotations, our approach exhibits a decrease on the part segmentation accuracy (46.07% for bed category and 68.17% for chair category). Although our method still yields clear improvements over the baseline setting without the fast adaptation (40.32% for bed category and 66.89% for chair category), it is essential to explore more effective and generalizable strategies for handling the geometric occlusion and intra-class variation among the fine-grained parts.

Acknowledgement

This work is supported by the Shandong Province Excellent Young Scientists Fund Program (Overseas) (No.2023HWYQ-034), National Natural Science Foundation of China (No.62302269), Shandong Provincial Natural Science Foundation (No.ZR2023QF077), the Key R&D Program of Shandong Province(No.2023CXGC010801), and the Joint Fund General Project of Liaoning Provincial Department of Science and Technology (2024-MSLH-352).

References

[1] A. Abdelreheem, I. Skorokhodov, M. Ovsjanikov, and P. Wonka. SATR: zero-shot semantic segmentation of 3d shapes. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 15120–15133. IEEE, 2023. 3

[2] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 1

[3] R. Chen, Y. Liu, L. Kong, X. Zhu, Y. Ma, Y. Li, Y. Hou, Y. Qiao, and W. Wang. Clip2scene: Towards label-efficient 3d scene understanding by CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 7020–7030. IEEE, 2023. 1, 3

[4] Z. Chen, Q. Chen, H. Zhou, and H. Zhang. Dae-net: Deforming auto-encoder for fine-grained shape co-segmentation. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–11, 2024. 2

[5] Z. Chen, K. Yin, M. Fisher, S. Chaudhuri, and H. Zhang. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8490–8499, 2019. 2

[6] M. Deitke, R. Liu, M. Wallingford, H. Ngo, O. Michel, A. Kusupati, A. Fan, C. Laforte, V. Voleti, S. Y. Gadre, E. VanderBilt, A. Kembhavi, C. Vondrick, G. Gkioxari, K. Ehsani, L. Schmidt, and A. Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 12

[7] C. Finn, P. Abbeel, and S. Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In D. Precup and Y. W. Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR, 2017. 3

[8] M. Garnelo, D. Rosenbaum, C. Maddison, T. Ramalho, D. Saxton, M. Shanahan, Y. W. Teh, D. J. Rezende, and S. M. A. Eslami. Conditional neural processes. In J. G. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1690–1699. PMLR, 2018. 3

[9] M. Garosi, R. Tedoldi, D. Boscaini, M. Mancini, N. Sebe, and F. Poiesi. 3d part segmentation via geometric aggregation of 2d visual features. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2025, Tucson, AZ, USA, February 26 - March 6, 2025*, pages 3257–3267. IEEE, 2025. 3

[10] M. Guo, J. Cai, Z. Liu, T. Mu, R. R. Martin, and S. Hu. PCT: point cloud transformer. *Comput. Vis. Media*, 7(2):187–199, 2021. 2

[11] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. Meshcnn: a network with an edge. *ACM Transactions on Graphics (ToG)*, 38(4):1–12, 2019. 2

[12] Y. Hao, H. Huang, S. Yuan, and Y. Fang. Meta-learning 3d shape segmentation functions. In *2024 10th International Conference on Automation, Robotics and Applications (ICARA)*, pages 516–520. IEEE, 2024. 3

[13] R. He, L. Liu, H. Ye, Q. Tan, B. Ding, L. Cheng, J.-W. Low, L. Bing, and L. Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. *arXiv preprint arXiv:2106.03164*, 2021. 3

[14] Y. He, H. Yu, X. Liu, Z. Yang, W. Sun, S. Anwar, and A. Mian. Deep learning based 3d segmentation in computer vision: A survey. *Information Fusion*, 115:102722, 2025. 1

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. 3

[16] S.-M. Hu, Z.-N. Liu, M.-H. Guo, J.-X. Cai, J. Huang, T.-J. Mu, and R. R. Martin. Subdivision-based mesh convolution networks. *ACM Transactions on Graphics (TOG)*, 41(3):1–16, 2022. 1, 2

[17] M. Huisman, J. N. van Rijn, and A. Plaat. A survey of deep meta-learning. *Artif. Intell. Rev.*, 54(6):4483–4541, 2021. 3

[18] H. Kim and M. Sung. Partstad: 2d-to-3d part segmentation task adaptation. In A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, editors, *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part V*, volume 15063 of *Lecture Notes in Computer Science*, pages 422–439. Springer, 2024. 3

[19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W. Lo, P. Dollár, and R. B. Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 3992–4003. IEEE, 2023. 1, 3

[20] W. M. Kouw and M. Loog. A review of domain adaptation without target labels. *IEEE transactions on pattern analysis and machine intelligence*, 43(3):766–785, 2019. 3

[21] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia. Stratified transformer for 3d point cloud segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8490–8499. IEEE, 2022. 2

[22] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J. Hwang, K. Chang, and J. Gao. Grounded language-image pre-training. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10955–10965. IEEE, 2022. 1, 2, 3, 4

[23] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. In S. Bengio, H. M. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 828–838, 2018. 1, 2

[24] J. Liang, R. He, and T. Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 133(1):31–64, 2025. 3

[25] M. Liu, Y. Zhu, H. Cai, S. Han, Z. Ling, F. Porikli, and H. Su. Partslip: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 21736–21746. IEEE, 2023. 1, 2, 6, 7

[26] Y. Liu, L. Kong, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu. Segment any point cloud sequences by distilling vision foundation models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. 3

[27] Z. Liu, H. Tang, Y. Lin, and S. Han. Point-voxel cnn for efficient 3d deep learning. *Advances in neural information processing systems*, 32, 2019. 1, 2

[28] Z. Ma, Y. Yue, and G. Gkioxari. Find any part in 3d. *CoRR*, abs/2411.13550, 2024. 1, 3, 7

[29] A. Mahmoud, J. S. K. Hu, T. Kuai, A. Harakeh, L. Paull,

and S. L. Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 7102–7110. IEEE, 2023. 3

[30] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 3

[31] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 909–918. Computer Vision Foundation / IEEE, 2019. 1, 2, 7

[32] A. Nichol, J. Achiam, and J. Schulman. On first-order meta-learning algorithms. *CoRR*, abs/1803.02999, 2018. 3

[33] C. Niu, M. Li, K. Xu, and H. Zhang. Rim-net: Recursive implicit fields for unsupervised learning of hierarchical shape structures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11779–11788, 2022. 2

[34] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P. Huang, S. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jégou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024. 1, 3

[35] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Z. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8024–8035, 2019. 7

[36] S. Peng, K. Genova, C. M. Jiang, A. Tagliasacchi, M. Pollefeys, and T. A. Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 815–824. IEEE, 2023. 3

[37] S. R. K. Perla, A. Vora, S. Nag, A. Mahdavi-Amiri, and H. Zhang. ASIA: adaptive 3d segmentation using few image annotations. In T. Komura, M. Wimmer, and H. Fu, editors, *Proceedings of the SIGGRAPH Asia 2025 Conference Papers, SA Conference Papers 2025, Hong Kong, December 15-18, 2025*, pages 168:1–168:12. ACM, 2025. 3

[38] S. Prokudin, C. Lassner, and J. Romero. Efficient learning on point clouds with basis point sets. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 4331–4340. IEEE, 2019. 6

[39] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 77–85. IEEE Computer Society, 2017. 1, 2

[40] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5099–5108, 2017. 2

[41] G. Qian, A. Abualshour, G. Li, A. K. Thabet, and B. Ghanem. PU-GCN: point cloud upsampling using graph convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 11683–11692. Computer Vision Foundation / IEEE, 2021. 2

[42] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021. 1, 3

[43] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 10674–10685. IEEE, 2022. 3

[44] C. Sautier, G. Puy, S. Gidaris, A. Boulch, A. Bursuc, and R. Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 9881–9891. IEEE, 2022. 3

[45] A. Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008. 1

[46] V. Sitzmann, E. Chan, R. Tucker, N. Snavely, and G. Wetzstein. Metasdf: Meta-learning signed distance functions. *Advances in Neural Information Processing Systems*, 33:10136–10147, 2020. 3

[47] J. Snell, K. Swersky, and R. S. Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 4077–4087, 2017. 3

[48] C. Sun, Y. Yang, H. Guo, P. Wang, X. Tong, Y. Liu, and H. Shum. Semi-supervised 3d shape segmentation with multilevel consistency and part substitution. *Comput. Vis. Media*, 9(2):229–247, 2023. 2

[49] Y. Sun, X. Zhang, and Y. Miao. A review of point cloud segmentation for understanding 3d indoor scenes. *Vis. Intell.*, 2(1):14, 2024. 2

[50] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1199–1208. Computer Vision Foundation / IEEE Computer Society, 2018. 3

[51] M. Tancik, B. Mildenhall, T. Wang, D. Schmidt, P. P. Srinivasan, J. T. Barron, and R. Ng. Learned initializations for optimizing coordinate-based neural representations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2846–2855, 2021. 3

[52] G. Tang, W. Zhao, L. Ford, D. Ben-Haim, and P. Zhang. Segment any mesh: Zero-shot mesh part segmentation via lifting segment anything 2 to 3d. *CoRR*, abs/2408.13679, 2024. 3

[53] T. H. Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025. 11

[54] A. Thai, W. Wang, H. Tang, S. Stojanov, M. Feiszli, and J. M. Rehg. 3x2: 3d object part segmentation by 2d semantic correspondences. *CoRR*, abs/2407.09648, 2024. 3

[55] H. Thomas, C. R. Qi, J. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 6410–6419. IEEE, 2019. 1, 2

[56] A. Umam, C. Yang, M. Chen, J. Chuang, and Y. Lin. Partdistill: 3d shape part segmentation by vision-language model distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 3470–3479. IEEE, 2024. 1, 3, 4, 7

[57] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638, 2016. 3

[58] L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan. Graph attention convolution for point cloud semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 10296–10305. Computer Vision Foundation / IEEE, 2019. 2

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

[60] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.*, 38(5):146:1–146:12, 2019. 1, 2

[61] Y. Wang, L. Wang, Q. Hu, Y. Liu, Y. Zhang, and Y. Guo. Panoptic segmentation of 3d point clouds with gaussian mixture model in outdoor scenes. *Vis. Intell.*, 2(1), 2024. 2

[62] Z. Wang and F. Lu. Voxsegnet: Volumetric cnns for semantic part segmentation of 3d shapes. *IEEE transactions on visualization and computer graphics*, 26(9):2919–2930, 2019. 2

[63] W. Wu, Z. Qi, and F. Li. Pointconv: Deep convolutional networks on 3d point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 9621–9630. Computer Vision Foundation / IEEE, 2019. 2

[64] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 11094–11104. Computer Vision Foundation / IEEE, 2020. 7

[65] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*, volume 11212 of *Lecture Notes in Computer Science*, pages 90–105. Springer, 2018. 2

[66] C. Yang, M. Chen, Y. Chuang, and Y. Lin. 2d-3d interlaced transformer for point cloud segmentation with scene-level supervision. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 977–987. IEEE, 2023. 3

[67] Y. Yang, Y. Huang, Y. Guo, L. Lu, X. Wu, E. Y. Lam, Y. Cao, and X. Liu. Sampart3d: Segment any part in 3d objects. *CoRR*, abs/2411.07184, 2024. 3

[68] L. Yi, V. G. Kim, D. Ceylan, I. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, and L. J. Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6):210:1–210:12, 2016. 1, 6

[69] F. Yu, Y. Qian, F. Gil-Ureta, B. Jackson, E. Bennett, and H. Zhang. Hal3d: Hierarchical active learning for fine-grained 3d part labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 865–875, 2023. 1

[70] L. Zhang, A. Rao, and M. Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 12

[71] R. Zhang, Z. Guo, P. Gao, R. Fang, B. Zhao, D. Wang, Y. Qiao, and H. Li. Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. 5

[72] R. Zhang, Z. Guo, W. Zhang, K. Li, X. Miao, B. Cui, Y. Qiao, P. Gao, and H. Li. Pointclip: Point cloud understanding by CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 8542–8552. IEEE, 2022. 1

[73] Z. Zhang, R. Girdhar, A. Joulin, and I. Misra. Self-supervised pretraining of 3d features on any point-cloud. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 10232–10243. IEEE, 2021. 3

[74] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada,*

October 10-17, 2021, pages 16239–16248. IEEE, 2021. 1, 2, 4

[75] Y. Zhou, J. Gu, X. Li, M. Liu, Y. Fang, and H. Su. Partslip++: Enhancing low-shot 3d part segmentation via multi-view instance segmentation and maximum likelihood estimation. *CoRR*, abs/2312.03015, 2023. 1, 3, 7

[76] X. Zhu, R. Zhang, B. He, Z. Guo, Z. Zeng, Z. Qin, S. Zhang, and P. Gao. Pointclip V2: prompting CLIP and GPT for powerful 3d open-world learning. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 2639–2650. IEEE, 2023. 1, 7