PartDistill: 3D Shape Part Segmentation by Vision-Language Model Distillation

Ardian Umam[1]   Cheng-Kun Yang[2]   Min-Hung Chen[3]   Jen-Hui Chuang[1]   Yen-Yu Lin[1]

[1]National Yang Ming Chiao Tung University   [2]MediaTek   [3]NVIDIA

Abstract

This paper proposes a cross-modal distillation framework, PartDistill, which transfers 2D knowledge from vision-language models (VLMs) to facilitate 3D shape part segmentation. PartDistill addresses three major challenges in this task: the lack of 3D segmentation in invisible or undetected regions in the 2D projections, inconsistent 2D predictions by VLMs, and the lack of knowledge accumulation across different 3D shapes. PartDistill consists of a teacher network that uses a VLM to make 2D predictions and a student network that learns from the 2D predictions while extracting geometrical features from multiple 3D shapes to carry out 3D part segmentation. A bidirectional distillation, including forward and backward distillations, is carried out within the framework, where the former forward distills the 2D predictions to the student network, and the latter improves the quality of the 2D predictions, which subsequently enhances the final 3D segmentation. Moreover, PartDistill can exploit generative models that facilitate effortless 3D shape creation for generating knowledge sources to be distilled. Through extensive experiments, PartDistill boosts the existing methods with substantial margins on widely used ShapeNetPart and PartNetE datasets, by more than 15% and 12% higher mIoU scores, respectively. The code for this work is available at https://github.com/ardianumam/PartDistill.

1. Introduction

3D shape part segmentation is essential to various 3D vision applications, such as shape editing [23, 45], stylization [29], and augmentation [40]. Despite its significance, acquiring part annotations for 3D data, such as point clouds or mesh shapes, is labor-intensive and time-consuming.

Zero-shot learning [8, 41] generalizes a model to unseen categories without annotations and has been notably uplifted by recent advances in vision-language models (VLMs) [21, 22, 37, 46]. By learning on large-scale image-text data pairs, VLMs show promising generalization abilities on various 2D recognition tasks. Recent research efforts [1, 24, 48, 52] have been made to utilize VLMs for zero-shot 3D part segmentation, where a 3D shape is pro-
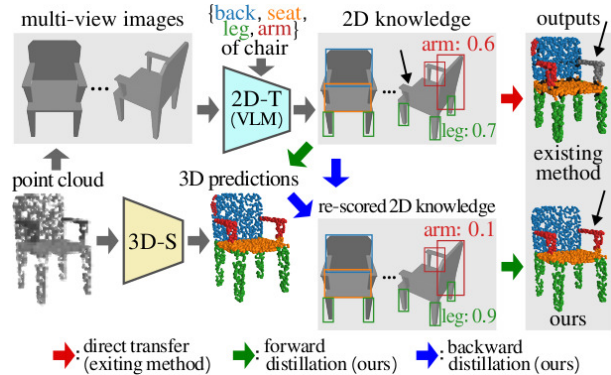

Figure 1. We present a distillation method that carries out zero-shot 3D shape part segmentation with a 2D vision-language model. After projecting an input 3D point cloud into multi-view 2D images, the 2D teacher (2D-T) and the 3D student (3D-S) networks are applied to the 2D images and 3D point cloud, respectively. Instead of direct transfer, our method carries bi-directional distillations, including forward and backward distillations, and yields better 3D part segmentation than the existing method.

jected into multi-view 2D images, and a VLM is applied to these images for 2D prediction acquisition. Specifically, PointCLIP [48] and PointCLIPv2 [52] produce 3D point-wise semantic segmentation by averaging their corresponding 2D pixel-wise predictions. Meanwhile, PartSLIP [24] and SATR [1] present a designated weighting mechanism to aggregate multi-view bounding box predictions.

The key step of zero-shot 3D part segmentation with 2D VLMs, *e.g.*, [1, 24, 48, 52], lies in the transfer from 2D pixel-wise or bounding-box-wise predictions to 3D point segmentation. This step is challenging due to three major issues. First ($\mathcal{I_1}$), some 3D regions lack corresponding 2D predictions in multi-view images, which are caused by occlusion or not being covered by any bounding boxes, illustrated with black and gray points, respectively, in Fig. 1. This issue is considered a limitation in the previous work [1, 24, 48, 52]. Second ($\mathcal{I_2}$), there exists potential inconsistency among 2D predictions in multi-view images caused by inaccurate VLM predictions. Third ($\mathcal{I_3}$), existing work [1, 24, 48, 52] directly transfers 2D predictions to segmentation of a single 3D shape. The 2D predictions yielded based on appearance features are not optimal for 3D geo-

metric shape segmentation, while geometric evidence given across different 3D shapes is not explored.

To alleviate the three issues $\mathcal{I}_1 \sim \mathcal{I}_3$, unlike existing methods [1, 24, 48, 52] directly transferring 2D predictions to 3D segmentation, we propose a cross-modal distillation framework with a teacher-student model. Specifically, a VLM is utilized as a 2D teacher network, accepting multi-view images of a single 3D shape. The VLM is pre-trained on large-scale image-text pairs and can exploit appearance features to make 2D predictions. The student network is developed based on a point cloud backbone. It is derived from multiple, unlabeled 3D shapes and can extract point-specific geometric features. The proposed distillation method, ***PartDistill***, leverages the strengths of both networks, hence improving zero-shot 3D part segmentation.

The student network learns from not only the 2D teacher network but also 3D shapes. It can extract point-wise features and segment 3D regions uncovered by 2D predictions, hence tackling issue $\mathcal{I}_1$. As a distillation-based method, PartDistill tolerates inconsistent predictions between the teacher and student networks, which alleviates issue $\mathcal{I}_2$ of negative transfer caused by wrong VLM predictions. The student network considers both appearance and geometric features. Thus, it can better predict 3D geometric data and mitigate issue $\mathcal{I}_3$. As shown in Fig. 1, the student network can correctly predict the undetected arm of the chair (see the black arrows) by learning from other chairs.

PartDistill carries out a bi-directional distillation. It first *forward distills* the 2D knowledge to the student network. We observe that after the student integrates the 2D knowledge, we can jointly refer both teacher and student knowledge to perform *backward distillation* which re-scores the 2D knowledge based on its quality. Those of low quality will be suppressed with lower scores, such as from 0.6 to 0.1 for the falsely detected arm box in Fig. 1, and vice versa. Finally, this re-scored knowledge is utilized by the student network to seek better 3D segmentation.

The main contributions of this work are summarized as follows. First, we introduce PartDistill, a cross-modal distillation framework that transfers 2D knowledge from VLMs to facilitate 3D part segmentation. PartDistill addresses three identified issues present in existing methods and generalizes to both VLM with bounding-box predictions (B-VLM) and pixel-wise predictions (P-VLM). Second, we propose a bi-directional distillation, which involves enhancing the quality of 2D knowledge and subsequently improving the 3D predictions. Third, PartDistill can leverage existing generative models [31, 33] to enrich knowledge sources for distillation. Extensive experiments demonstrate that PartDistill surpasses existing methods by substantial margins on widely used benchmark datasets, ShapeNet-Part [44] and PartNetE [24], with more than 15% and 12% higher mIoU scores, respectively. PartDistill consistently outperforms competing methods in zero-shot and few-shot scenarios on 3D data in point clouds or mesh shapes.

2. Related Work

Vision-language models. Based on learning granularity, vision-language models (VLMs) can be grouped into three categories, including the image-level [15, 37], pixel-level [21, 27, 53], and object-level [22, 25, 46] categories. The second and the third categories make pixel-level and bounding box predictions, respectively, while the first category produces image-level predictions. Recent research efforts on VLMs have been made for cross-level predictions. For example, pixel-level predictions can be derived from an image-level VLM via up-sampling the 2D features into the image dimensions, as shown in PointCLIPv2 [52]. In this work, we propose a cross-modal distillation framework that learns and transfers knowledge from a VLM in the 2D domain to 3D shape part segmentation.

3D part segmentation using vision-language models. State-of-the-art zero-shot 3D part segmentation [1, 24, 52] is developed by utilizing a VLM and transferring its knowledge in the 2D domain to the 3D space. The pioneering work PointCLIP [48] utilizes CLIP [37]. PointCLIPv2 [52] extends PointCLIP by making the projected multi-view images more realistic and proposing LLM-assisted text prompts [4], hence producing more reliable CLIP outputs for 3D part segmentation.

Both PointCLIP and PointCLIPv2 rely on individual pixel predictions in 2D views to get the predictions of the corresponding 3D points, but individual pixel predictions are less unreliable. PartSLIP [24] suggests to extract superpoints [20] from the input point cloud. Therefore, 3D segmentation is estimated for each superpoint by referring to a set of relevant pixels in 2D views. PartSLIP uses GLIP [22] to output bounding boxes and further proposes a weighting mechanism to aggregate multi-view bounding box predictions to yield 3D superpoint predictions. SATR [1] shares a similar idea with PartSLIP but handles 3D mesh shapes instead of point clouds.

Existing methods [1, 24, 48, 52] directly transfer VLM predictions from 2D images into 3D spaces and pose three issues: ($\mathcal{I}_1$) uncovered 3D points, ($\mathcal{I}_2$) negative transfer, and ($\mathcal{I}_3$) cross-modality predictions, as discussed before. We present a distillation-based method to address all three issues and make substantial performance improvements.

2D to 3D distillation. Seminal work of knowledge distillation [5, 14] aims at transferring knowledge from a large model to a small one. Subsequent research efforts [26, 28, 39, 43, 50] adopt this idea of transferring knowledge from a 2D model for 3D understanding. However, these methods require further fine-tuning with labeled data. Open-Scene [34] and CLIP2Scene [7] require no fine-tuning and share a similar concept with our method of distilling VLMs
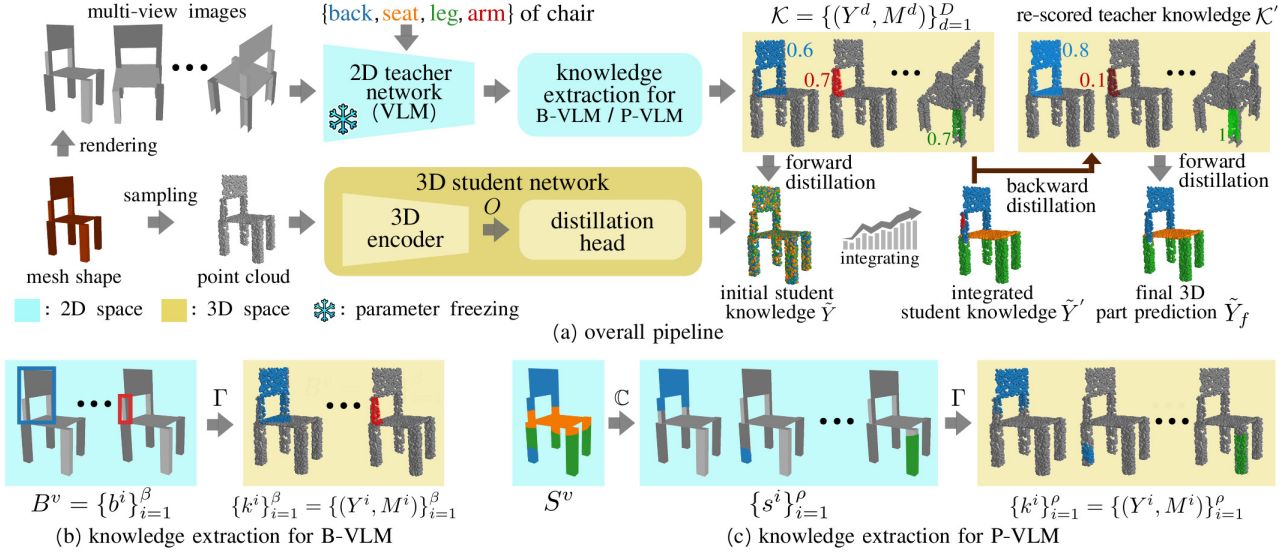
Figure 2. **Overview of the proposed method.** (a) The overall pipeline where the knowledge extracted from a vision-language model (VLM) is distilled to carry out 3D shape part segmentation by teaching a 3D student network. Within the pipeline, *backward distillation* is introduced to re-score the teacher's knowledge based on its quality and subsequently improve the final 3D part prediction. (b) & (c) Knowledge is extracted by back-projection when we adopt (b) a bounding-box VLM (B-VLM) or (c) a pixel-wise VLM (P-VLM), where Γ and $\mathbb{C}$ denote 2D-to-3D back-projection and connected component labeling [3], respectively.

for 3D understanding, with ours designed for part segmentation and theirs for indoor/outdoor scene segmentation. The major difference is that our method can enhance the knowledge sources in the 2D modality via the proposed backward distillation. Moreover, our method is generalizable to both P-VLM (pixel-wise VLM) and B-VLM (bounding-box VLM), while their methods are only applicable to P-VLM.

3. Proposed Method

3.1. Overview

Given a set of 3D shapes, this work aims to segment each one into R semantic parts without training with any part annotations. To this end, we propose a cross-modal bi-directional distillation framework, **PartDistill**, which transfers 2D knowledge from a VLM to facilitate 3D shape part segmentation. As illustrated in Fig. 2, our framework takes triplet data as input, including the point cloud of the shape with N 3D points, multi-view rendered images from the shape in V different poses, and R text prompts with each describing the target semantic parts within the 3D shapes.

For the 2D modality, the V multi-view images and the text prompts are fed into a Bounding-box VLM (B-VLM) or Pixel-wise VLM (P-VLM). For each view v, a B-VLM produces a set of bounding boxes, $B^v = \{b^i\}_{i=1}^{\beta}$ while a P-VLM generates pixel-wise predictions S^v. We then perform *knowledge extraction* (Sec 3.2) for each B^v or S^v; Namely, we transfer the 2D predictions into the 3D space through back-projection for a B-VLM or connected-component labeling [3] followed by back-projection for a P-VLM, as shown in Fig. 2 (b) and Fig. 2 (c), respec-

tively. Subsequently, a set of D *teacher* knowledge units, $\mathcal{K} = \{k\}_{d=1}^{D} = \{Y^d, M^d\}_{d=1}^{D}$, is obtained by aggregating from all V multi-view images. Each unit d comprises point-wise part probabilities, $Y^d \in \mathbb{R}^{N \times R}$, from the teacher VLM network, accompanied with a mask, $M^d \in \{0, 1\}^N$, identifying the points included in this knowledge unit.

For the 3D modality, the point cloud is passed into the 3D *student* network with a 3D encoder and a distillation head, producing point-wise part predictions, $\tilde{Y} \in \mathbb{R}^{N \times R}$. With the proposed bi-directional distillation framework, we first *forward distill* teacher's 2D knowledge by aligning $\tilde{Y}$ with $\mathcal{K}$ via minimizing the proposed loss, $\mathcal{L}_{distill}$, specified in Sec 3.2. The 3D student network integrates 2D knowledge from the teacher through optimization. The integrated student knowledge $\tilde{Y}'$ and the teacher knowledge $\mathcal{K}$ are then jointly referred to perform *backward distillation* from 3D to 2D, detailed in Sec 3.3, which re-scores each knowledge unit k^d based on its qualities, as shown in Fig. 2. Finally, the re-scored knowledge $\mathcal{K}'$ is used to refine the student knowledge to get final part segmentation predictions $\tilde{Y}_f$ by assigning each point to the part with the highest probability.

3.2. Forward distillation: 2D to 3D

Our method extracts the teacher's knowledge in the 2D modality and distills it in the 3D space. In the 2D modality, V multi-view images $\{I^v \in \mathbb{R}^{H \times W}\}_{v=1}^{V}$ are rendered from the 3D shape, *e.g.*, using the projection method in [52]. These V multi-view images together with the text prompts T of R parts are passed to the VLM to get the knowledge in 2D spaces. For a B-VLM, a set of β bounding

boxes, $B^v = \{b^i\}_{i=1}^\beta$, is obtained from the v-th image, with $b^i \in \mathbb{R}^{4+R}$ encoding the box coordinates and the probabilities of the R parts. For a P-VLM, a pixel-wise prediction map $S^v \in \mathbb{R}^{H \times W \times R}$ is acquired from the v-th image. We apply *knowledge extraction* to each B^v and each S^v to obtain a readily distillable knowledge $\mathcal{K}$ in the 3D space, as illustrated in Fig. 2 (b) and Fig. 2 (c), respectively.

For a B-VLM, bounding boxes can directly be treated as the teacher knowledge. For a P-VLM, knowledge extraction starts by applying connected-component labeling [3] to S^v to get a set of ρ segmentation components, $\{s^i \in \mathbb{R}^{H \times W \times R}\}_{i=1}^\rho$, indicating if the r-th part is with the highest probability in each pixel. We summarize the process when applying a VLM to a rendered image and the part text prompts as

$$\text{VLM}(I^v, T) = \begin{cases} B^v = \{b^i\}_{i=1}^\beta, & \text{for B-VLM,} \\ \mathbb{C}(S^v) = \{s^i\}_{i=1}^\rho, & \text{for P-VLM,} \end{cases} \quad (1)$$

where $\mathbb{C}$ denotes connected-component labeling.

We then back-project each box b^i or each prediction map s^i to the 3D space, *i.e.*,

$$k^i = (Y^i, M^i) = \begin{cases} \Gamma(b^i), & \text{for B-VLM,} \\ \Gamma(s^i), & \text{for P-VLM,} \end{cases} \quad (2)$$

where Γ denotes the back-projection operation with the camera parameters [49] used for multi-view image rendering, $Y^i \in \mathbb{R}^{N \times R}$ is the point-specific part probabilities, and $M^i \in \{0,1\}^N$ is the mask indicating which 3D points are covered by b^i or s^i in the 2D space. The pair (Y^i, M^i) yields a knowledge unit, k^i, upon which the knowledge re-scoring is performed in the backward distillation.

For the 3D modality, a 3D encoder, *e.g.*, Point-M2AE [47], is applied to the point cloud and obtains per-point features, $O \in \mathbb{R}^{N \times E}$, capturing local and global geometrical information. We then estimate point-wise part prediction, $\tilde{Y} \in R^{N \times R}$, by feeding the point features O into the distillation head. The cross-modal distillation is performed by teaching the student network to align the part probability from the 3D modality $\tilde{Y}$ to their 2D counterparts Y via minimizing our designated distillation loss.

Distillation loss. Via Eq. 1 and Eq. 2, we assume that D knowledge units, $\mathcal{K} = \{k^d\}_{d=1}^D = \{Y^d, M^d\}_{d=1}^D$, are obtained from the multi-view images. The knowledge $\mathcal{K}$ exploits 2D appearance features and is incomplete as several 3D points are not covered by any 2D predictions, *i.e.*, issue $\mathcal{I}_1$. To distill this incomplete knowledge, we utilize a masked cross-entropy loss defined as

$$\mathcal{L}_{distill} = -\sum_{d=1}^D \frac{1}{|M^d|} \sum_{n=1}^N \sum_{r=1}^R M_n^d C_n^d Z_{n,r}^d \log(\tilde{Y}_{n,r}), \quad (3)$$

where $C_n^d = \max_r(Y_n^d(r))$ is the confidence score of k^d on point n. $Z_{n,r}^d$ takes value 1 if part r receives the highest

probability on k^d, and 0 otherwise. $|M^d|$ is the area covered by the mask M^d.

By minimizing Eq. 3, we teach the student network to align its prediction $\tilde{Y}$ to the distilled prediction Y by considering the points covered by the mask and using the confidence scores as weights. Despite learning from incomplete knowledge, the student network extracts point features that capture geometrical information of the shape, thus enabling it to reasonably segment the points that are not covered by 2D predictions, hence addressing issue $\mathcal{I}_1$. This can be regarded as interpolating the learned part probability in the feature spaces by the distillation head.

As a distillation-based method, our method allows partial inconsistency among the extracted knowledge $\mathcal{K} = \{k^d\}_{d=1}^D$ caused by inaccurate VLM predictions, thereby alleviating issue $\mathcal{I}_2$ of negative transfer. In our method, the teacher network works on 2D appearance features, while the student network extracts 3D geometric features. After distillation via Eq. 3, the student network can exploit both appearance and geometric features from multiple shapes, hence mitigating issue $\mathcal{I}_3$ of cross-modal transfer. It is worth noting that unlike the conventional teacher-student models [11, 13, 14] which solely establish a one-to-one correspondence, we further re-score each knowledge unit k^d based on its quality (Sec. 3.3), and improve distillation by suppressing low-quality knowledge units.

3.3. Backward distillation: 3D to 2D

In Eq. 3, we consider all knowledge units $\{k^d\}_{d=1}^D$, weighted by their confidence scores. However, due to the potential VLM mispredictions, not all knowledge units are reliable. Hence, we refine the knowledge units by assigning higher scores to those of high quality and suppressing the low-quality ones. We observe that once the student network has thoroughly integrated the knowledge from the teacher, we can jointly refer both teacher and integrated student knowledge $\tilde{Y}'$ to achieve the goal, by re-scoring the confidence score C^d to C_{bd}^d as:

$$C_{bd}^d = \frac{|M^d(\text{argmax}(Y^d) \Leftrightarrow \text{argmax}(\tilde{Y}'))|}{|M^d|}, \quad (4)$$

where $\Leftrightarrow$ denotes the element-wise equality (comparison) operation. In this way, each knowledge unit k^d is re-scored: Those with high consensus between teacher $\mathcal{K}$ and integrated student knowledge $\tilde{Y}'$ have higher scores, such as those on the chair legs shown in Fig. 3, and those with low consensus are suppressed by the reduced scores, such as those on the chair arm (B-VLM) and back (P-VLM) in Fig. 3. Note that for simplicity, we only display two scores in each shape of Fig. 3 and show the average pixel-wise scores in P-VLM. To justify that the student network has thoroughly integrated the teacher's knowledge, from initial knowledge $\tilde{Y}$ to integrated knowledge $\tilde{Y}'$, we track the
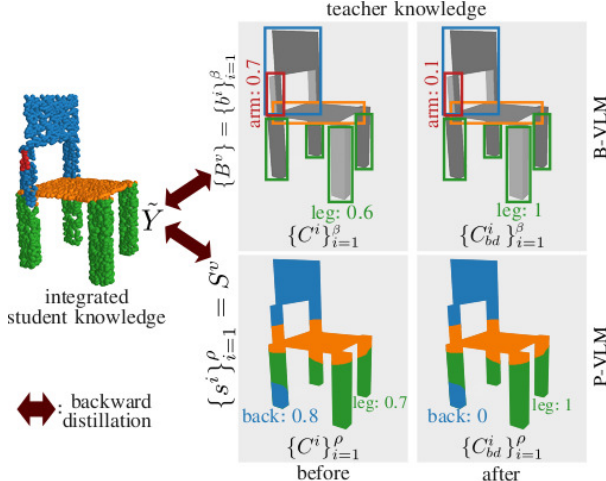
Figure 3. Given the VLM output of view v, B^v or S^v, we display the confidence scores before (C) and after (C_{bd}) performing backward distillation via Eq. 4, with Y and M obtained via Eq. 2. With backward distillation, inaccurate VLM predictions have lower scores, such as the arm box in B-VLM with the score reduced from 0.7 to 0.1, and vice versa.

moving average of the loss value for every epoch and see if the value in a subsequent epoch is lower than a specified threshold τ. Afterward, the student network continues to learn with the re-scored knowledge $\mathcal{K}'$ by minimizing the loss in Eq. 3 with C being replaced by C_{bd}, and produces the final part segmentation predictions $\tilde{Y}_f$.

3.4. Test-time alignment

In general, our method performs the alignment with a shape collection before the student network is utilized to carry the 3D shape part segmentation. If such a pre-alignment is not preferred, we provide a special case of our method, test-time alignment (TTA), where the alignment is performed for every single shape in test time. To maintain the practicability, TTA needs to achieve a near-instantaneous completion. To that end, TTA employs a readily used 3D encoder, *e.g.*, pre-trained Point-M2AE [47], freezes its weights, and only updates the learnable parameters in the distillation head, which significantly fastens the TTA completion.

3.5. Implementation Details

The proposed framework is implemented in PyTorch [32] and is optimized for 25 epochs via Adam optimizer [19] with a learning rate and batch size of 0.001 and 16, respectively. Unless further specified, the student network employs Point-M2AE [47] pre-trained in a self-supervised way on the ShapeNet55 dataset [6] as the 3D extractor, freezes its weights, and only updates the learnable parameters in the distillation head. A multi-layer perceptron consisting of 4 layers, with ReLU activation [2], is adopted for the distillation head. To fairly compare with the competing methods [1, 24, 48, 52], we follow their respective

settings, including the used text prompts and the 2D rendering. Their methods render each shape into 10 multi-view images, either from a sparse point cloud [48, 52], a dense point cloud [24], or a mesh shape [1]. Lastly, we follow [18, 38] to specify a small threshold value, $\tau = 0.01$ in our backward distillation, and apply class-balance weighting [9] during the alignment, based on the VLM predictions in the zero-shot setting, with additional few-shot labels in the few-shot setting.

4. Experiments

4.1. Dataset and evaluation metric

We evaluate the effectiveness of our method on two main benchmark datasets, ShapeNetPart [44] and PartNetE [24]. While ShapeNetPart dataset contains 16 categories with a total of 31,963 shapes, PartNetE contains 2,266 shapes, covering 45 categories. The mean intersection over union (mIoU) [30] is adopted to evaluate the segmentation results on the test-set data, measured against the ground truth label.

4.2. Zero-shot segmentation

To compare with the competing methods [1, 24, 48, 52], we adopt each of their settings and report their mIoU performances from their respective papers. Specifically, for P-VLM, we follow PointCLIP [48] and PointCLIPv2 [52] to utilize CLIP [37] with ViT-B/32 [10] backbone and use their pipeline to obtain the pixel-wise predictions from CLIP. For B-VLM, a GLIP-Large model [22] is employed in our method to compare with PartSLIP and SATR which also use the same model. While most competing methods report their performances on the ShapeNetPart dataset, PartSLIP evaluates its method on the PartNetE dataset. In addition, we compare with OpenScene [34] by extending it for 3D part segmentation and use the same Point-M2AE [47] backbone and VLM CLIP for a fair comparison.

Accordingly, we carry out the comparison separately to ensure fairness, based on the employed VLM model and the shape data type, *i.e.*, point cloud or mesh data, as shown in Tables 1 and 2. In Table 1, we provide two versions of our method, including test-time alignment (TTA) and pre-alignment (Pre) with a collection of shapes from the train-set data. Note that in the Pre version, our method does not use any labels (only unlabeled shape data are utilized).

First, we compare our method to PointCLIP and PointCLIPv2 (both utilize CLIP) on the zero-shot segmentation for the ShapeNetPart dataset, as can be seen in the first part of Table 1. It is evident that our method for both TTA and pre-alignment versions achieves substantial improvements in all categories. For the overall mIoU, calculated by averaging the mIoUs from all categories, our method attains 5.4% and 15.5% higher mIoU for TTA and pre-alignment versions, respectively, compared to the best mIoU from the other methods. Such results reveal that our method which simultaneously exploits appearance and geometric features

Table 1. Zero-shot segmentation on the ShapeNetPart dataset, reported in mIoU (%).*

VLM	Data type	Method	Airplane	Bag	Cap	Chair	Earphone	Guitar	Knife	Laptop	Mug	Table	Overall
CLIP [37]	point cloud	PointCLIP [48]	22.0	44.8	13.4	18.7	28.3	22.7	24.8	22.9	48.6	45.4	31.0
		PointCLIPv2 [52]	35.7	53.3	53.1	51.9	48.1	59.1	66.7	61.8	45.5	49.8	48.4
		OpenScene [34]	34.4	63.8	56.1	59.8	62.6	69.3	70.1	65.4	51.0	60.4	52.9
		Ours (TTA)	**37.5**	62.6	55.5	56.4	55.6	**71.7**	**76.9**	**67.4**	53.5	62.9	**53.8**
		Ours (Pre)	**40.6**	75.6	67.2	65.0	66.3	85.8	79.8	92.6	83.1	68.7	63.9
GLIP [22]	point cloud	Ours (TTA)	57.3	62.7	56.2	74.2	45.8	60.6	78.5	85.7	82.5	62.9	54.7
		Ours (Pre)	69.3	70.1	67.9	86.5	51.2	76.8	85.7	91.9	85.6	79.6	64.1
	mesh	SATR [1]	32.2	32.1	21.8	25.2	19.4	37.7	40.1	50.4	76.4	22.4	32.3
		Ours (TTA)	**53.2**	**61.8**	**44.9**	**66.4**	**43.0**	**50.7**	**66.3**	**68.3**	**83.9**	**58.8**	**49.5**
		Ours (Pre)	**64.8**	**64.4**	**51.0**	**67.4**	**48.3**	**64.8**	**70.0**	**83.1**	**86.5**	**79.3**	**56.3**

*Results for other categories, including those of Table 2 and Table 3, can be seen in the supplementary material.

Table 2. Zero-shot segmentation on the PartNetE dataset, reported in mIoU (%).

VLM	Data type	Method	Bottle	Cart	Chair	Display	Kettle	Knife	Lamp	Oven	Suitcase	Table	Overall
GLIP [22]	point cloud	PartSLIP [24]	76.3	87.7	60.7	43.8	20.8	46.8	37.1	33.0	40.2	47.7	27.3
		Ours (TTA)	**77.4**	**88.5**	**74.1**	**50.5**	**24.2**	**59.2**	**58.8**	**34.2**	**43.2**	**50.2**	**39.9**

can better aggregate the 2D predictions for 3D part segmentation than directly averaging the corresponding 2D predictions as in the competing methods, where geometric evidence is not explored. We further compare with OpenScene [34] under the same setting as ours (Pre) and our method substantially outperforms it. One major reason is that our method can handle the inconsistency of VLM predictions (issue $\mathcal{I}_2$) better by backward distillation.

Next, as shown in the last three rows of Table 1, we compare our method to SATR [1] that works on mesh data shapes. To obtain the mesh face predictions, we propagate the point predictions via a nearest neighbors approach as in [17], where each face is voted from its five nearest points. Our method achieves 17.2% and 24% higher overall mIoU than SATR for TTA and pre-alignment versions, respectively. Then, we compare our method with PartSLIP [24] in Table 2 wherein only results from TTA are provided since the PartNetE dataset does not provide train-set data. One can see that our method consistently obtains better segmentations, with 12.6% higher overall mIoU than PartSLIP.

In PartSLIP and SATR, as GLIP is utilized, the uncovered 3D regions (issue $\mathcal{I}_1$) could be intensified by possible undetected areas, and the negative transfer (issue $\mathcal{I}_2$) may also be escalated due to semantic leaking, where the box predictions cover pixels from other semantics. On the other hand, our method can better alleviate these issues, thereby achieving substantially higher mIoU scores. In our method, the pre-alignment version achieves better segmentation results than TTA. This is expected since in the pre-alignment version, the student network can distill the knowledge from a collection of shapes, instead of individual shape.

Besides foregoing quantitative comparisons, a qualitative comparison of the segmentation results is presented in Fig. 4. It is readily observed that the competing methods suffer from the lack of 3D segmentation for the uncovered regions (issue $\mathcal{I}_1$) caused by either occlusion or not being covered by any bounding box, drawn with black and gray

colors, respectively. Moreover, these methods may also encounter negative transfers caused by inaccurate VLM outputs (issue $\mathcal{I}_2$), such as those pointed by blue arrows, with notably degraded outcomes in SATR due to semantic leaking. Nonetheless, our method performs cross-modal distillation and alleviates these two issues, as can be seen in Fig. 4. In addition, due to a direct transfer of 2D predictions to 3D space which relies on each independent shape as in the competing methods, erroneous 2D predictions will just remain as incorrect 3D segmentation (issue $\mathcal{I}_3$), such as the missed detected chair arms and guitar heads pointed by red arrows. Our method also addresses this issue, by exploiting geometrical features across multiple shapes.

4.3. Few-shot segmentation

We further demonstrate the effectiveness of our method in a few-shot scenario by following the setting used in Part-SLIP [24]. Specifically, we employ the fine-tuned GLIP model [22] provided by PartSLIP via 8-shot labeled shapes of the PartNetE dataset [44] for each category. In addition to the alignment via Eq. 3, we ask the student network to learn parameters that minimize both Eq. 3 and a standard cross-entropy loss for segmentation on the 8 labeled shapes.

As shown in Table 3, the methods dedicated to few-shot 3D segmentation, ACD [12] and Prototype [51], are adapted to PointNet++ [51] and PointNext [36] backbones, respectively, and can improve the performances (on average) of these backbones. PartSLIP, on the other hand, leverages multi-view GLIP predictions for 3D segmentation and further improves the mIoU, but there are still substantial performance gaps compared to our method which distills the GLIP predictions instead. We also present the results from fine-tuning Point-M2AE with the few-shot labels, which shows lower performances than ours, highlighting the significant contribution of our distillation framework. For more qualitative results, see the supplementary materials.

point cloud - CLIP	mesh - GLIP	point cloud - GLIP	mesh

PointCLIP PointCLIPv2 Ours (TTA) Ours (Pre) | SATR Ours (TTA) Ours (Pre) | PartSLIP Ours (TTA) Ours (Pre) | Ground truths
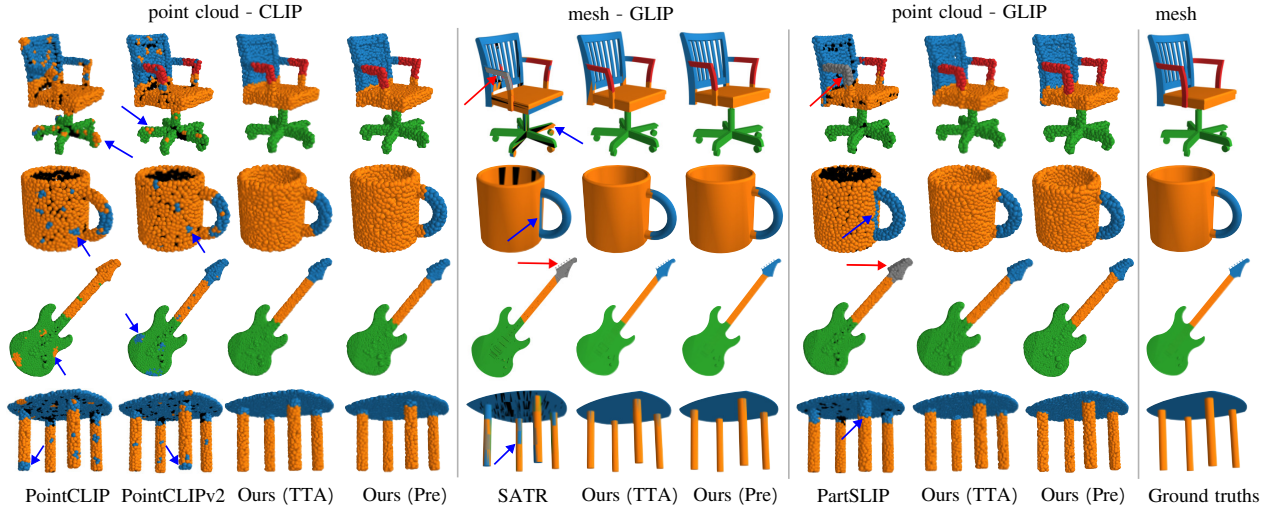
Figure 4. Visualization of the zero-shot segmentation results, drawn in different colors, on the ShapeNetPart dataset. We render PartSLIP results on the ShapeNetPart data to have the same visualization of shape inputs. While occluded and undetected regions (issue $\mathcal{I}_1$) are shown with black and gray colors, respectively, the blue and red arrows highlight several cases of issues $\mathcal{I}_2$ and $\mathcal{I}_3$.

Table 3. Few-shot segmentation on the PartNetE dataset, reported in mIoU (%).

	Method	Bottle	Cart	Chair	Display	Kettle	Knife	Lamp	Oven	Suitcase	Table	Overall
	PointNet++ [35]	27.0	11.6	42.2	30.2	28.6	22.2	10.5	19.4	3.3	7.3	20.4
	PointNext [36]	67.6	47.7	65.1	53.7	60.6	59.7	55.4	36.8	14.5	22.1	40.6
Non-VLM-based	ACD [12]	22.4	31.5	39.0	29.2	40.2	39.6	13.7	8.9	13.2	13.5	23.2
	Prototype [51]	60.1	36.8	70.8	67.3	62.7	50.4	38.2	36.5	35.5	25.7	44.3
	Point-M2AE [47]	72.4	74.5	83.4	74.3	64.3	68.0	57.6	53.3	57.5	33.6	56.4
VLM-based	PartSLIP [24]	83.4	88.1	85.3	84.8	77.0	65.2	60.0	**73.5**	70.4	42.4	59.4
(GLIP [22])	Ours	**84.6**	**90.1**	**88.4**	**87.4**	**78.6**	**71.4**	**69.2**	72.8	**73.4**	**63.3**	**65.9**

4.4. Leveraging generated data

Since only unlabeled 3D shape data are required for our method to perform cross-modal distillation, existing generative models [31, 33] can facilitate an effortless generation of 3D shapes, and the generated data can be smoothly incorporated into our method. Specifically, we first adopt DiT-3D [31] which is pre-trained on the ShapeNet55 dataset [6] to generate point clouds of shapes, 500 shapes for each category, and further employ SAP [33] to transform the generated point clouds into mesh shapes. These generated mesh shapes can then be utilized in our method for distillation. Table 4 shows the results evaluated on the test-set data of ShapeNetPart [44] and COSEG [42] datasets for several shape categories, using GLIP as the VLM.

One can see that with distilling from the generated alone, our method already achieves competitive results on the ShapeNetPart dataset compared to distilling from the train-set data. Since the generated data via DiT-3D is pre-trained on the ShapeNet55 dataset which contains the ShapeNet-Part data, we also evaluate its performance on the COSEG dataset to show that such results can be well transferred to shapes from another dataset. Finally, Table 4 (the last row) reveals that using generated data as a supplementary knowledge source can further increase the mIoU performance. Such results suggest that if a collection of shapes is

available, generated data can be employed as supplementary knowledge sources, which can improve the performance. On the other hand, if a collection of shapes does not exist, generative models can be employed for shape creation and subsequently used in our method as the knowledge source.

4.5. Ablation studies

Proposed components. We perform ablation studies on the proposed components, and the mIoU scores in 2D[1] and 3D spaces on three categories of the ShapeNetPart dataset are shown in (1) to (9) of Table 5. In (1), only GLIP box predictions are utilized to get 3D segmentations, *i.e.*, part labels are assigned by voting from all visible points within the multi-view box predictions. These numbers serve as baselines and are subject to issues $\mathcal{I}_1 \sim \mathcal{I}_3$. In (2) and (3), 3D segmentations are achieved via *forward distillation* from the GLIP predictions to the student network using Eq. 3, for test-time alignment (TTA) and pre-alignment (Pre) versions, resulting in significant improvements compared to the baselines, with more than 10% and 14% higher mIoUs, respectively. Such results demonstrate that the proposed cross-modal distillation can better utilize the 2D multi-view predictions for 3D part segmentation, alleviating $\mathcal{I}_1 \sim \mathcal{I}_3$.

---

[1]Calculated between the VLM predictions and their corresponding 2D ground truths projected from 3D, and weighted by the confidence scores. See supplementary material for the details.

Table 4. Segmentation mIoU (%) by leveraging generated data.

Distilled data	ShapeNetPart [44]			COSEG [42]	
	Airplane	Chair	Guitar	Chair	Guitar
Train-set (baseline)	69.3	86.2	76.8	96.4	68.0
Gen. data	69.0	85.3	75.6	96.1	67.5
Gen. data & train-set	70.8	88.4	78.3	97.4	70.2

Table 5. Ablation study on the proposed method.

No	VLM	Pre	BD	Student network	Airplane		Chair		Knife	
					2D	3D	2D	3D	2D	3D
(1)					42.8	40.2	60.2	60.1	53.6	57.2
(2)				✓	42.8	56.2	60.2	73.5	53.6	77.6
(3)		✓		✓	42.8	64.3	60.2	84.2	53.6	84.5
(4)			✓	✓	44.3	57.3	61.7	74.2	54.8	78.5
(5)	GLIP [22]	✓	✓	✓	48.2	**69.3**	63.2	86.5	55.0	**85.7**
(6)		✓	✓	exclude $\mathcal{I}_1$	48.2	62.5	63.2	80.4	55.0	81.2
(7)			✓	w/o pretrain	48.2	69.1	63.2	**86.7**	55.0	85.3
(8)	CLIP [37]	✓		✓	34.6	38.4	50.4	63.6	66.8	77.4
(9)		✓	✓	✓	37.8	**40.6**	54.2	**65.0**	68.4	**78.9**

We further add *backward distillation* (BD) in (4) and (5), which substantially improves the knowledge source in 2D, *e.g.*, from 42.8% to 48.2% for the airplane category in the pre-alignment version, and subsequently enhances the 3D segmentation. We observe a higher impact (improvement) on the pre-alignment compared to TTA versions, *i.e.*, in (4) and (5), as the student network of the former can better integrate the knowledge from a collection of shapes. A similar trend of improvement can be observed for a similar ablation performed with CLIP [37] used as the VLM (in (8) and (9)).

In (6), we exclude our method's predictions for those uncovered points to simulate issue $\mathcal{I}_1$, and the reduced mIoUs compared to (5), *e.g.*, from 86.5% to 80.4% for the chair category, reveal that our method can effectively alleviate issue $\mathcal{I}_1$. Finally, instead of using pre-trained weights of Point-M2AE [47] and freezing them as the 3D decoder as in (5), we initialize these weights (by default PyTorch [32] initialization) and set them to be learnable as in (7). Both settings produce comparable results (within 0.4%). The main purpose of using the pre-trained weights and freezing them is for faster convergence, especially for the test-time alignment purpose. Please refer to the supplementary material for the comparison of convergence curves.

Number of views. We render $V = 10$ multi-view images for each shape input in our main experiment, and Fig. 5 (left) shows the mIoU scores with different values of V. A substantial drop is observed when utilizing $V < 6$, and small increases are obtained when a larger V is used.

Various shape types for 2D multi-view rendering. We render 10 multi-view images from various shape data types, *i.e.*, (i) gray mesh, (ii) colored mesh, (iii) dense colored point cloud ($\sim$300k points) as used in PartSLIP [24], and (iv) sparse gray point cloud (2,048 points) using PyTroch3D [16] and the rendering method in [52] to render (i)-(iii) and (iv), respectively. Fig. 5 (right) summarizes such results on the ShapeNetPart dataset, with GLIP used as the VLM. Note that the first three shape types produce
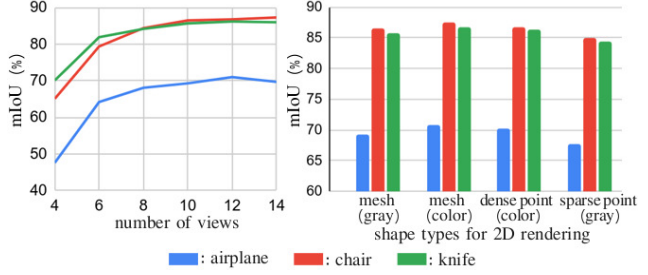

Figure 5. Ablation study on number of views and various shape types for 2D multiview rendering on the ShapeNetPart dataset.

comparable mIoUs with slightly higher scores when colored mesh or dense colored point cloud is utilized. When sparse gray point cloud data type is used, a mild mIoU decrease is observed. Please refer to the supplementary material to see more results for (i)-(iv).

Limitation. The main limitation of our method is that the segmentation results are impacted by the quality of the VLM predictions, where VLMs are generally pre-trained to recognize object- or sample-level categories (not part-level of object categories). For instance, GLIP can satisfactorily locate part semantics for the chair category but with lower qualities for the earphone category, while CLIP can favorably locate part semantics for the earphone category but with less favorable results for the airplane category. Hence, exploiting multiple VLMs can be a potential future work. Nonetheless, the proposed method which currently employs a single VLM model can already boost the segmentation results significantly compared to the existing methods.

5. Conclusion

We present a cross-modal distillation framework that transfers 2D knowledge from a vision-language model (VLM) to facilitate 3D shape part segmentation, which generalizes well to both VLM with bounding-box and pixel-wise predictions. In the proposed method, backward distillation is introduced to enhance the quality of 2D predictions and subsequently improve the 3D segmentation. The proposed approach can also leverage existing generative models for shape creation and can be smoothly incorporated into the method for distillation. With extensive experiments, the proposed method is compared with existing methods on widely used benchmark datasets, including ShapeNetPart and PartNetE, and consistently outperforms existing methods with substantial margins both in zero-shot and few-shot scenarios on 3D data in point clouds or mesh shapes.

References

[1] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. SATR: Zero-shot semantic segmentation of 3D shapes. In *ICCV*, 2023. 1, 2, 5, 6

[2] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint*, 2018. 5

[3] Federico Bolelli, Stefano Allegretti, Lorenzo Baraldi, and Costantino Grana. Spaghetti labeling: Directed acyclic graphs for block-based connected components labeling. *TIP*, 2019. 3, 4

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 2020. 2

[5] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *KDD*, 2006. 2

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3D model repository. *arXiv preprint*, 2015. 5, 7

[7] Runnan Chen, Youquan Liu, Lingdong Kong, Xinge Zhu, Yuexin Ma, Yikang Li, Yuenan Hou, Yu Qiao, and Wenping Wang. Clip2scene: Towards label-efficient 3D scene understanding by clip. In *CVPR*, 2023. 2

[8] Ali Cheraghian, Shafin Rahman, Townim F Chowdhury, Dylan Campbell, and Lars Petersson. Zero-shot learning on 3D point cloud objects and beyond. *IJCV*, 2022. 1

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 5

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 5

[11] Yingchao Feng, Xian Sun, Wenhui Diao, Jihao Li, and Xin Gao. Double similarity distillation for semantic image segmentation. *TIP*, 2021. 4

[12] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *ECCV*, 2020. 6, 7

[13] Kyle Genova, Xiaoqi Yin, Abhijit Kundu, Caroline Pantofaru, Forrester Cole, Avneesh Sud, Brian Brewington, Brian Shucker, and Thomas Funkhouser. Learning 3D semantic segmentation with only 2D image supervision. In *3DV*, 2021. 4

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *NIPS Workshop*, 2015. 2, 4

[15] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[16] Justin Johnson, Nikhila Ravi, Jeremy Reizenstein, David Novotny, Shubham Tulsiani, Christoph Lassner, and Steve Branson. Accelerating 3D deep learning with pytorch3d. In *arXiv preprint*. 2020. 8

[17] Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, and Siddhartha Chaudhuri. 3D shape segmentation with projective convolutional networks. In *CVPR*, 2017. 6

[18] John D Kelleher, Brian Mac Namee, and Aoife D'arcy. *Fundamentals of machine learning for predictive data analytics: algorithms, worked examples, and case studies*. MIT press, 2020. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint*, 2014. 5

[20] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *CVPR*, 2018. 2

[21] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and Rene Ranftl. Language-driven semantic segmentation. In *ICLR*, 2022. 1, 2

[22] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 1, 2, 5, 6, 7, 8

[23] Connor Lin, Niloy Mitra, Gordon Wetzstein, Leonidas J Guibas, and Paul Guerrero. Neuform: Adaptive overfitting for neural shape editing. *NeurIPS*, 2022. 1

[24] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. PartSLIP: Low-shot part segmentation for 3D point clouds via pretrained image-language models. In *CVPR*, 2023. 1, 2, 5, 6, 7, 8

[25] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint*, 2023. 2

[26] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. *NeurIPS*, 2023. 2

[27] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *CVPR*, 2022. 2

[28] Anas Mahmoud, Jordan SK Hu, Tianshu Kuai, Ali Harakeh, Liam Paull, and Steven L Waslander. Self-supervised image-to-point distillation via semantically tolerant contrastive loss. In *CVPR*, 2023. 2

[29] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022. 1

[30] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *CVPR*, 2019. 5

[31] Shentong Mo, Enze Xie, Ruihang Chu, Lewei Yao, Lanqing Hong, Matthias Nießner, and Zhenguo Li. DiT-3D: Exploring plain diffusion transformers for 3D shape generation. *arXiv preprint*, 2023. 2, 7

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch:

An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 5, 8

[33] Songyou Peng, Chiyu Jiang, Yiyi Liao, Michael Niemeyer, Marc Pollefeys, and Andreas Geiger. Shape as points: A differentiable poisson solver. *NeurIPS*, 2021. 2, 7

[34] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3D scene understanding with open vocabularies. In *CVPR*, 2023. 2, 5, 6

[35] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *NeurIPS*, 2017. 7

[36] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *NeurIPS*, 2022. 6, 7

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1, 2, 5, 6, 8

[38] Douglas J Santry. *Demystifying deep learning: An introduction to the mathematics of neural networks*. Wiley, 2023. 5

[39] Corentin Sautier, Gilles Puy, Spyros Gidaris, Alexandre Boulch, Andrei Bursuc, and Renaud Marlet. Image-to-lidar self-supervised distillation for autonomous driving data. In *CVPR*, 2022. 2

[40] Ardian Umam, Cheng-Kun Yang, Yung-Yu Chuang, Jen-Hui Chuang, and Yen-Yu Lin. Point mixswap: Attentional point cloud mixing via swapping matched structural divisions. In *ECCV*, 2022. 1

[41] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019. 1

[42] Yunhai Wang, Shmulik Asafi, Oliver Van Kaick, Hao Zhang, Daniel Cohen-Or, and Baoquan Chen. Active co-analysis of a set of shapes. *TOG*, 2012. 7, 8

[43] Cheng-Kun Yang, Min-Hung Chen, Yung-Yu Chuang, and Yen-Yu Lin. 2D-3D interlaced transformer for point cloud segmentation with scene-level supervision. In *ICCV*, 2023. 2

[44] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3D shape collections. *ToG*, 2016. 2, 5, 6, 7, 8

[45] Yu-Jie Yuan, Yu-Kun Lai, Tong Wu, Lin Gao, and Ligang Liu. A revisit of shape editing techniques: From the geometric to the neural viewpoint. *Journal of Computer Science and Technology*, 2021. 1

[46] Haotian Zhang, Pengchuan Zhang, Xiaowei Hu, Yen-Chun Chen, Liunian Li, Xiyang Dai, Lijuan Wang, Lu Yuan, Jenq-Neng Hwang, and Jianfeng Gao. Glipv2: Unifying localization and vision-language understanding. *NeurIPS*, 2022. 1, 2

[47] Renrui Zhang, Ziyu Guo, Peng Gao, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, and Hongsheng Li. Point-m2ae: multi-scale masked autoencoders for hierarchical point cloud pre-training. *NeurIPS*, 2022. 4, 5, 7, 8

[48] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. In *CVPR*, 2022. 1, 2, 5, 6

[49] Zhengyou Zhang. A flexible new technique for camera calibration. *TPAMI*, 2000. 4

[50] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *ICCV*, 2021. 2

[51] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3D point cloud semantic segmentation. In *CVPR*, 2021. 6, 7

[52] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyao Zeng, Shanghang Zhang, and Peng Gao. Pointclip v2: Adapting clip for powerful 3D open-world learning. *ICCV*, 2023. 1, 2, 3, 5, 6, 8

[53] Xueyan Zou, Jianwei Yang, Hao Zhang, Feng Li, Linjie Li, Jianfeng Gao, and Yong Jae Lee. Segment everything everywhere all at once. *NeurIPS*, 2023. 2