PartDistill: 3D Shape Part Segmentation by Vision-Language Model Distillation (Supplementary Material)

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

A. 3D segmentation scores for full categories

We provide 3D segmentation scores, reported in mIoU, for full categories of the ShapeNetPart [R1] and PartE [R7] datasets in Tables A and B, respectively. Table A is associated with Table 1 in the main paper, while Table B is associated with Tables 2 and 3. In Table A, 16 categories of the ShapeNetPart dataset are reported, while 45 categories of the PartE dataset are presented in Table B. For the tables, it is readily observed that the proposed method, ***PartDistill***, attains substantial improvements compared to the competing methods [R4]–[R7] in most categories.

B. Evaluating 2D predictions

In the ablation studies of our method's components presented in Table 5, we provide mIoU scores in 2D space, mIoU$^{\text{2D}}$, to evaluate the quality of the 2D predictions measured against the 2D ground truths before and after performing *backward distillation* which re-scores the confidence scores of each knowledge unit. Here, the 2D ground truths are obtained by projecting the 3D mesh (faces) part segmentation labels to 2D space using the camera parameters utilized when performing 2D multi-view rendering.

We first explain how to calculate the mIoU$^{\text{2D}}$ if a vision-language model (VLM) which outputs pixel-wise predictions (P-VLM) is used in our method and later explain if a VLM which outputs bounding-box predictions (B-VLM) is employed. In each view, let $\{s^i\}_i^\rho$ be the prediction maps (see Eq. 1 in the main paper) of P-VLM with C^i denoting the confidence score of s^i and $\mathcal{G}$ be the corresponding 2D ground truth. We first calculate the IoU$^{\text{2D}}$ for each semantic part r as,

$$\text{IoU}^{\text{2D}}(r) = \frac{\mathcal{I}(r)}{\mathcal{I}(r) + \lambda(r) + \gamma(r)} \tag{A}$$

where

$$\mathcal{I}(r) = \sum_{i \in \phi(r)} \text{Avg}(C^i)(s^i \cap \mathcal{G}^r), \tag{B}$$

$$\lambda(r) = \text{Avg}(C^{\phi(r)})\left(\left(\bigcup_{i \in \phi(r)} s^i\right) \notin \mathcal{G}^r\right), \tag{C}$$

and

$$\gamma(r) = \mathcal{G}^r \notin \bigcup_{i \in \phi(r)} s^i, \tag{D}$$

with $\phi(r)$ denoting a function returning indices of $\{s^i\}_{i=1}^\rho$ that predict part r, "Avg" denoting an averaging operation and $\mathcal{G}^r$ indicating the ground truth of part r.

While Eq. B represents the intersection of the pixels between the 2D predictions and the corresponding ground truths, weighted by their confidence scores, Eq. C tells the union of the 2D predictions pixels that do not intersect with the corresponding ground truths, which is weighted by the average of all confidence scores associated with part r. As for Eq. D, it tells the ground truth pixels that do not intersect with the corresponding 2D predictions union. We then calculate the IoU$^{\text{2D}}$ score for each semantic part r in every v view and compute the mean of them as mIoU$^{\text{2D}}$.

Note that we involve the confidence scores as weights to calculate the mIoU$^{\text{2D}}$. This allows us to compare the quality of the 2D predictions before and after applying *backward distillation*, using the confidence scores before and after this process. To compute the mIoU$^{\text{2D}}$ scores when a B-VLM is used in our method, we can use Eq. A with s^i in Eq. B$\sim$ Eq. D being replaced by $\mathcal{F}(b^i)$, where $\mathcal{F}$ denotes an operation excluding the background pixels covered in b^i.

C. Additional visualizations

C.1. Visualization of few-shot segmentation

In Figure A, we present several visualizations for few-shot segmentation obtained via our method, associated with Table 3 in the main paper. Following the prior work [R7], 8-shot labeled shapes are utilized to carry the few-shot segmentation. From the figure, it is evident that our method successfully achieves satisfactory segmentation results.

Table A. Zero-shot segmentation on all 16 categories of the ShapeNetPart dataset [R1], reported mIoU (%). In this table, TTA and Pre denote the test-time alignment and pre-alignment versions of our method, while VLM stands for vision-language model (see main paper for details).

Category	VLM - CLIP [R2]				VLM - GLIP [R3]				
	point cloud input				point cloud input		mesh input		
	PointCLIP [R4]	PointCLIP v2 [R5]	Ours (TTA)	Ours (Pre)	Ours (TTA)	Ours (Pre)	SATR [R6]	Ours (TTA)	Ours (Pre)
Airplane	22.0	35.7	**37.5**	**40.6**	57.3	69.3	32.2	**53.2**	**64.8**
Bag	44.8	53.3	**62.6**	**75.6**	62.7	70.1	32.1	**61.8**	**64.4**
Cap	13.4	53.1	**55.5**	**67.2**	56.2	67.9	21.8	**44.9**	**51.0**
Car	30.4	34.5	**36.4**	**41.2**	32.4	39.2	22.3	**30.2**	**32.3**
Chair	18.7	51.9	**56.4**	**65.0**	74.2	86.5	25.2	**66.4**	**67.4**
Earphone	28.3	48.1	**55.6**	**66.3**	45.8	51.2	19.4	**43.0**	**48.3**
Guitar	22.7	59.1	**71.7**	**85.8**	60.6	76.8	37.7	**50.7**	**64.8**
Knife	24.8	66.7	**76.9**	**79.8**	78.5	85.7	40.1	**66.3**	**70.0**
Lamp	39.6	44.7	**45.8**	**63.1**	34.5	43.5	21.6	**30.5**	**35.2**
Laptop	22.9	61.8	**67.4**	**92.6**	85.7	91.9	50.4	**68.3**	**83.1**
Motorbike	26.3	31.4	**33.4**	**38.2**	30.6	37.8	25.4	**28.8**	**32.5**
Mug	48.6	45.5	**53.5**	**83.1**	82.5	85.6	76.4	**83.9**	**86.5**
Pistol	42.6	46.1	**48.2**	**55.8**	39.6	48.5	34.1	**37.4**	**40.9**
Rocket	22.7	46.7	**49.3**	**49.5**	36.8	48.9	33.2	**41.1**	**45.3**
Skateboard	42.7	45.8	**47.7**	**49.2**	34.2	43.5	22.3	**26.2**	**34.5**
Table	45.4	49.8	**62.9**	**68.7**	62.9	79.6	22.4	**58.8**	**79.3**
Overall	31.0	48.4	**53.8**	**63.9**	54.7	64.1	32.3	**49.5**	**56.3**

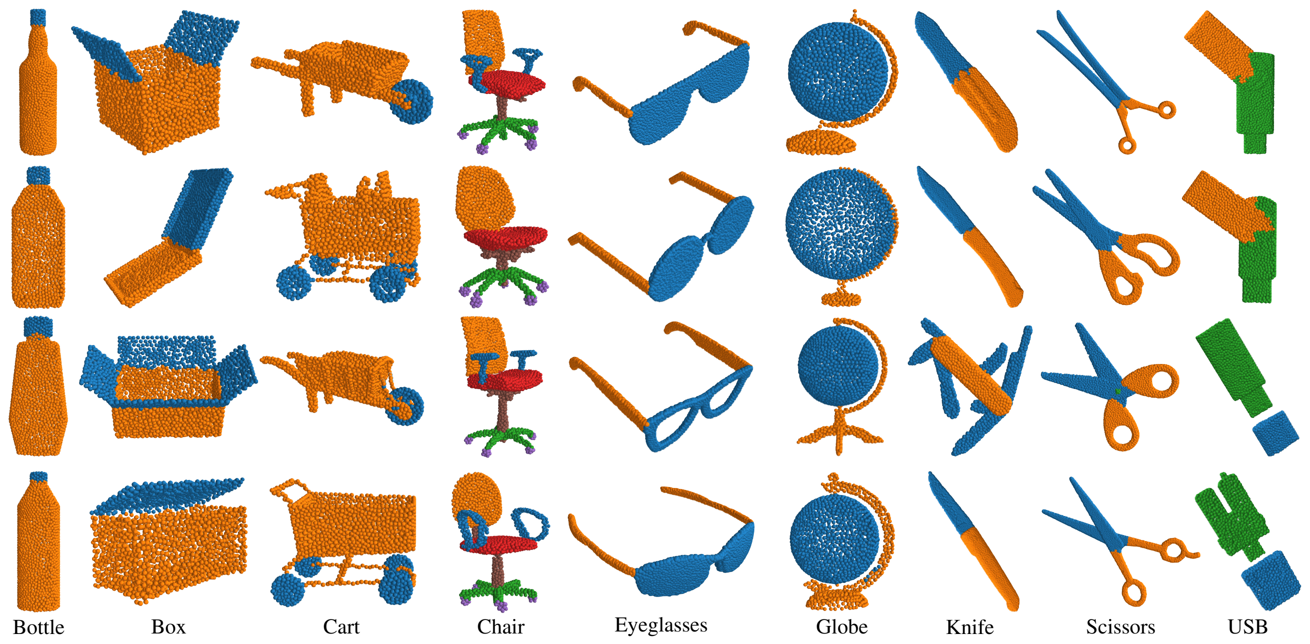

Figure A. Visualization of few-shot segmentation results derived using our method on the PartE dataset [R7]. Each semantic part is drawn in different colors.

C.2. Convergence curves

In the ablation studies presented in Table 5, we compare two approaches used in the 3D encoder of our student network. First, we employ a pre-trained PointM2AE [R8] backbone, freeze the weights and only update the learnable parameters in the student network's distillation head. Second, we utilize a PointM2AE backbone with its weights initialized by PyTorch [R9] default initialization and set them to be learnable, together with the parameters in the distillation head. From the table, we observe comparable results between both settings (see rows (5) and (7) for the first and second approaches respectively).

We then visualize the convergence curves in both settings, as depicted in Figure B. From the figure, it can be seen that the loss in the first approach converges significantly faster than in the second approach. As a result, the

Table B. Segmentation on all 45 categories of the PartE dataset [R7], reported in mIoU (%). In this table, TTA denotes our method with test-time alignment (see main paper for details).

Category	Zero-shot		Few-shot	
	PartSLIP [R7]	Ours (TTA)	PartSLIP [R7]	Ours
Bottle	76.3	**77.4**	83.4	**84.6**
Box	57.5	**69.7**	84.5	**87.9**
Bucket	2.0	**16.8**	36.5	**50.7**
Camera	21.4	**29.4**	58.3	**60.1**
Cart	87.7	**88.5**	88.1	**90.1**
Chair	60.7	**74.1**	85.3	**88.4**
Clock	**26.7**	23.6	**37.6**	37.2
Coffee machine	25.4	**26.8**	37.8	**40.2**
Dishwasher	10.3	**18.6**	62.5	**60.2**
Dispenser	16.5	**11.4**	73.8	**74.7**
Display	43.8	**50.5**	84.8	**87.4**
Door	2.7	**41.1**	40.8	**55.5**
Eyeglasses	1.8	**59.7**	88.3	**91.1**
Faucet	6.8	**33.3**	71.4	**73.5**
Folding chair	91.7	**89.7**	86.3	**90.7**
Globe	34.8	**90.0**	95.7	**97.4**
Kettle	20.8	**24.2**	77.0	**78.6**
Keyboard	37.3	**38.5**	53.6	**70.8**
Kitchenpot	4.7	**36.8**	69.6	**69.7**
Knife	46.8	**59.2**	65.2	**71.4**
Lamp	37.1	**58.8**	66.1	**69.2**
Laptop	27.0	**37.1**	29.7	**40.0**
Lighter	35.4	**37.3**	64.7	**64.9**
Microwave	16.6	**23.2**	42.7	**43.8**
Mouse	**27.0**	18.6	44.0	**46.9**
Oven	33.0	**34.2**	**73.5**	72.8
Pen	14.6	**15.7**	71.5	**74.4**
Phone	36.1	**37.3**	48.4	**50.8**
Pliers	5.4	**51.9**	33.2	**90.4**
Printer	0.8	**3.3**	4.3	**6.3**
Refrigerator	20.2	**25.2**	55.8	**58.1**
Remote	11.5	**13.2**	38.3	**40.7**
Safe	**22.4**	18.2	32.2	**58.6**
Scissors	21.8	**64.4**	60.3	**68.8**
Stapler	20.9	**65.1**	84.8	**86.3**
Storage furniture	29.5	**30.6**	53.6	**56.5**
Suitcase	40.2	**43.2**	70.4	**73.4**
Switch	9.5	**30.3**	59.4	**60.7**
Table	47.7	**50.2**	42.5	**63.3**
Toaster	**13.8**	11.4	**60.0**	58.7
Toilet	20.6	**22.5**	53.8	**55.0**
Trash can	30.1	**49.3**	22.3	**70.0**
Usb	10.9	**39.1**	54.4	**64.3**
Washing machine	12.5	**12.9**	53.5	**55.1**
Window	5.2	**45.3**	75.4	**78.1**
Overall	27.3	**39.9**	59.4	**65.9**

first approach also starts to perform *backward distillation* in a substantially earlier epoch than the second one.

C.3. 2D rendering from various shape types

We present several 2D rendering images from various shape types, including (i) gray mesh, (ii) colored mesh, (iii) dense colored point cloud, and (iv) sparse gray point cloud, which can be seen in Figure C. While PartSLIP [R7] renders the multi-view images using type (iii), SATR [R6] uses type (i). As for PointCLIP [R4] and PointCLIPv2 [R5], they use type (iv) to render their multi-view images.

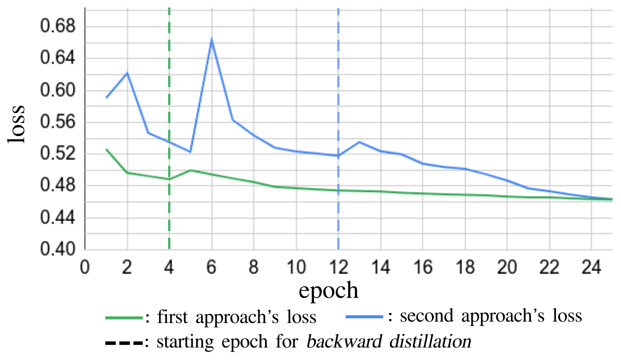

Figure B. Convergence curves of our method's losses during optimization epochs. While the first approach employs a pre-trained PointM2AE [R8] model and freezes its weights, the second approach initializes the Point2MAE's weights from scratch and sets them to be learnable.

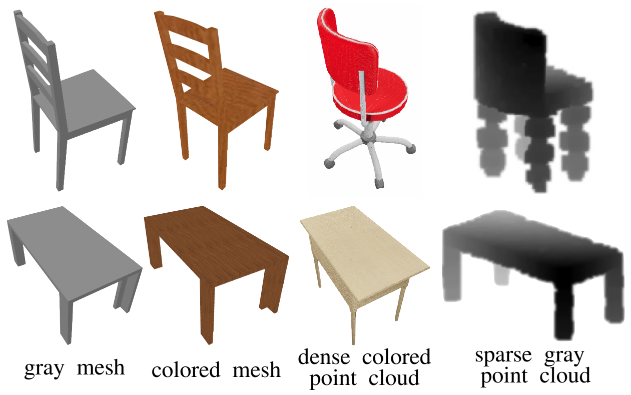

Figure C. Several examples of 2D images rendered from various shape types.

References

[R1] L. Yi, V. G. Kim, D. Ceylan, *et al.*, "A scalable active framework for region annotation in 3D shape collections," *ToG*, 2016.

[R2] A. Radford, J. W. Kim, C. Hallacy, *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[R3] L. H. Li, P. Zhang, H. Zhang, *et al.*, "Grounded language-image pre-training," in *CVPR*, 2022.

[R4] R. Zhang, Z. Guo, W. Zhang, *et al.*, "Pointclip: Point cloud understanding by clip," in *CVPR*, 2022.

[R5] X. Zhu, R. Zhang, B. He, Z. Zeng, S. Zhang, and P. Gao, "Pointclip v2: Adapting clip for powerful 3D open-world learning," *ICCV*, 2023.

[R6] A. Abdelreheem, I. Skorokhodov, M. Ovsjanikov, and P. Wonka, "SATR: Zero-shot semantic segmentation of 3D shapes," in *ICCV*, 2023.

[R7] M. Liu, Y. Zhu, H. Cai, *et al.*, "PartSLIP: Low-shot part segmentation for 3D point clouds via pretrained image-language models," in *CVPR*, 2023.

[R8] R. Zhang, Z. Guo, P. Gao, *et al.*, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," *NeurIPS*, 2022.

[R9] A. Paszke, S. Gross, F. Massa, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *NeurIPS*, 2019.