PartSLIP: Low-Shot Part Segmentation for 3D Point Clouds via Pretrained Image-Language Models – Supplementary Material

Minghua Liu[1], Yinhao Zhu[2], Hong Cai[2], Shizhong Han[2], Zhan Ling[1], Fatih Porikli[2], Hao Su[1]
[1]UC San Diego, [2]Qualcomm

S. Supplementary Material

S.1. PartNet-Ensembled Dataset

Table S1 shows the statistics of the proposed PartNet-Ensembled (PartNetE) dataset. The few-shot and test shapes come from PartNet-Mobility [12], and the additional training shapes come from PartNet [5]. All three sets share consistent part definitions. To construct a diverse, clear, and consistent 3D object-part dataset, we select a subset of 100 object parts from the original PartNet and PartNet-Mobility annotations, and manually annotate three additional parts (i.e., Kettle spout, KitchenPot handle, and Mouse cord). Specifically, we filter out extremely fine-grained parts (e.g., "back_frame_vertical_bar" for chairs), ambiguous parts, inconsistently annotated parts, and rarely seen parts of the original datasets. As a result, each object category contains 1-6 parts in our PartNetE dataset, covering both common coarse-grained parts (e.g., chair back and tabletop) and fine-grained parts (e.g., wheel, handle, button, knob, switch, touchpad) that may be useful in downstream tasks such as robotic manipulation. For shapes from PartNet-Mobility, they have textures, while for shapes from PartNet, they do not. The unbalanced data distribution is a critical issue when using the additional 28k training shapes. We may have nearly 10k shapes for common categories (e.g., Table) but only 8 for some non-overlapping categories. We believe our dataset could benefit future works on low-shot and text-driven 3D part annotation, which do not rely on large-scale supervised learning to infer part definitions.

S.2. Real-World Demo

Figure S1 shows more examples when our method and baseline approaches are applied to point clouds captured by an iPhone 12 Pro Max equipped with a LiDAR sensor. Specifically, we utilize the APP "polycam" to scan daily objects and generate fused point clouds with color. We use MeshLab to remove ground points and compute point normals. For baseline approaches, we randomly sample 10,000 points as input.

As shown in the figure, our method can directly generalize to iPhone-scanned point clouds without significant domain gaps, while baseline methods perform poorly. For PointNext [7] of the "45x8+28k" setting (third row), it uses the additional 28k training data but still fails to recognize many parts (e.g., cart wheels, trashcan footpedal, lid and head of the dispenser, chair wheels, suitcase wheels, drawers and handles of the storage furniture, handle of the kettle). The few-shot version (fourth row) performs even worse and can only identify a few parts.

S.3. Visualization of Ablation Studies

Few-Shot Prompt Tuning Figure S2 shows the comparison before and after few-shot prompt tuning. The pretrained GLIP model (first row) fails to understand the meaning of many part names. However, after prompt tuning with only one or a few segmented 3D shapes (second row), the GLIP model quickly adapts to part definitions and can generalize to unseen instances.

Multi-View Visual Feature Aggregation Figure S3 shows the comparison with and without multi-view visual feature aggregation. When there is no multi-view visual feature aggregation (first row), the GLIP model fails to detect parts from some unfamiliar camera views. However, after aggregating visual features from multiple views (second row), the GLIP model can comprehensively understand input 3D shapes and make more accurate predictions for those unfamiliar views.

Variations of Input Point Clouds To evaluate the robustness of our method, we have tried multiple variations of input point clouds (see Table 4 of the main paper). Figure S4 exemplifies 2D images used to generate input point clouds and point cloud renderings fed to the FLIP model. In the original setting, we use 6 RGB-D images with a resolution of 512x512 to generate the fused point cloud, which is then projected to 10 2D images with a resolution of 800x800. Note that when point clouds are sparse, we increase the point size to reduce the artifacts of point cloud renderings. Please zoom in to find the differences between point cloud renderings. As shown in Table 4 of the main paper, our proposed method is robust against various input point cloud variations.

Table S1. The table shows the statistics of the PartNetE dataset: category name, part names, number of few-shot shapes, test shapes, and additional training shapes (if applicable). The 17 overlapping object categories are bolded.

category	parts	few-shot	test	extra-train	category	parts	few-shot	test	extra-train
Bottle	lid	8	49	471	**Microwave**	display, door, handle, button	8	8	234
Box	lid	8	20	0	Mouse	button, cord, wheel	8	6	0
Bucket	handle	8	28	0	Oven	door, knob	8	22	0
Camera	button, lens	8	29	0	Pen	cap, button	8	40	0
Cart	wheel	8	53	0	Phone	lid, button	8	10	0
Chair	arm, back, leg, seat, wheel	8	73	8000	Pliers	leg	8	17	0
Clock	hand	8	23	593	Printer	button	8	21	0
CoffeeMachine	button, container, knob, lid	8	46	0	**Refrigerator**	door, handle	8	36	195
Dishwasher	door, handle	8	40	179	Remote	button	8	41	0
Dispenser	head, lid	8	49	0	Safe	door, switch, button	8	22	0
Display	base, screen, support	8	29	954	**Scissors**	blade, handle, screw	8	39	60
Door	frame, door, handle	8	28	237	Stapler	body, lid	8	15	0
Eyeglasses	body, leg	8	57	0	**StorageFurniture**	door, drawer, handle	8	338	2260
Faucet	spout, switch	8	76	681	Suitcase	handle, wheel	8	16	0
FoldingChair	seat	8	18	0	Switch	switch	8	62	0
Globe	sphere	8	53	0	**Table**	door, drawer, leg, tabletop, wheel, handle	8	93	9799
Kettle	lid, handle, spout	8	21	0	Toaster	button, slider	8	17	0
Keyboard	cord, key	8	29	165	Toilet	lid, seat, button	8	61	0
KitchenPot	lid, handle	8	17	0	**TrashCan**	footpedal, lid, door	8	62	358
Knife	blade	8	36	505	USB	cap, rotation	8	43	0
Lamp	base, body, bulb, shade	8	37	3246	WashingMachine	door, button	8	9	0
Laptop	keyboard, screen, shaft, touchpad, camera	8	47	430	Window	window	8	50	0
Lighter	lid, wheel, button	8	20	0	**45 in total**	**103 in total**	**360**	**1,906**	**28,367**

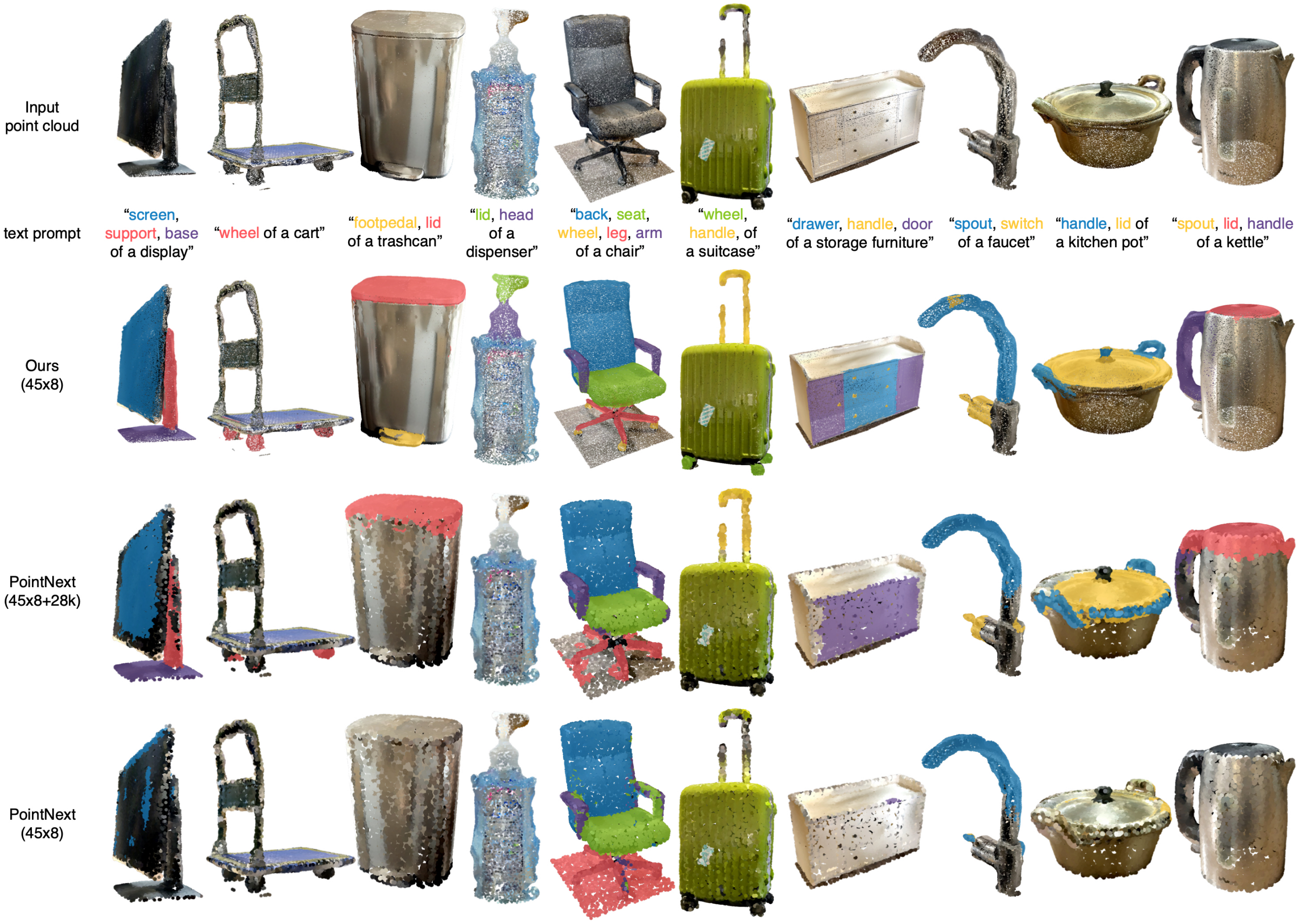
Input point cloud
text prompt
"screen, support, base of a display"
"wheel of a cart"
"footpedal, lid of a trashcan"
"lid, head of a dispenser"
"back, seat, wheel, leg, arm of a chair"
"wheel, handle, of a suitcase"
"drawer, handle, door of a storage furniture"
"spout, switch of a faucet"
"handle, lid of a kitchen pot"
"spout, lid, handle of a kettle"
Ours (45x8)
PointNext (45x8+28k)
PointNext (45x8)

Figure S1. Real-world demo: iPhone-scanned point clouds (first row), text prompt for our method (second row), results of our method and baseline approaches (third to fifth rows). "45x8" indicates the few-shot setting, where the model is trained with 8 shapes per object category. "45x8+28k" indicates the setting where the additional 28k shapes are used for training. Zoom in for details.

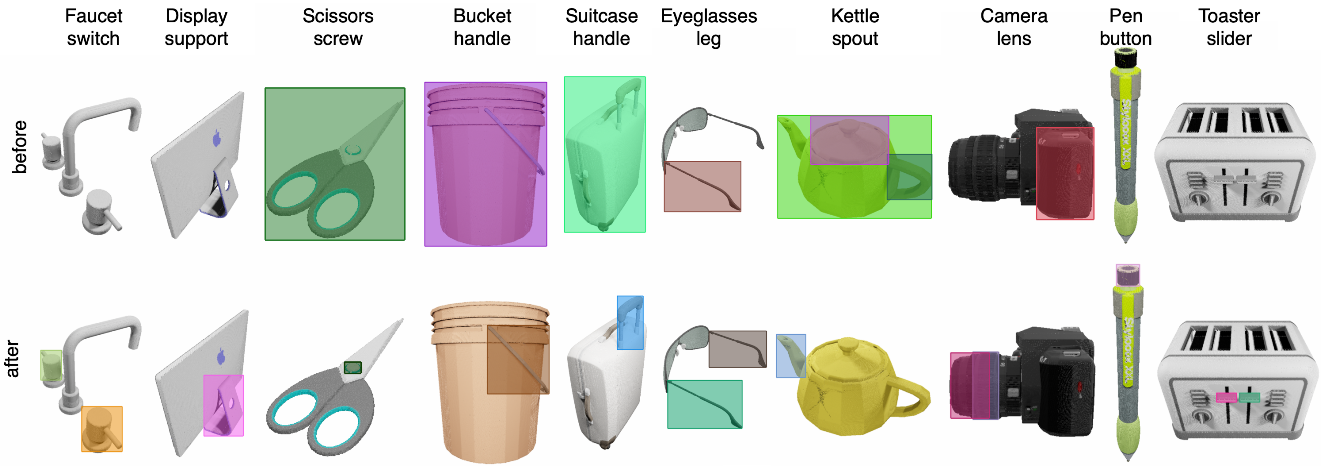

Figure S2. Ablation study of few-shot prompt tuning. First row: 2D part detection results of the GLIP pretrained model (zero-shot). Second row: detection results after 8-shot prompt tuning.

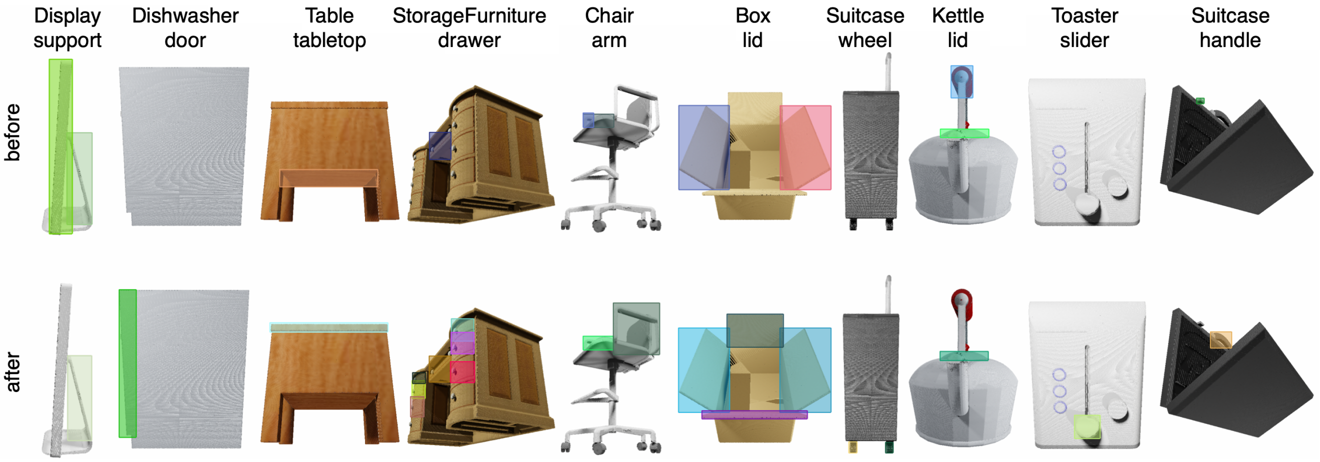

Figure S3. Ablation study of multi-view visual feature aggregation. First row: 2D part detection results without the multi-view visual feature aggregation. Second row: detection results with the multi-view feature aggregation. Both models are prompt-tuned.

S.4. Text Prompts

In our experiments, our few-shot version (with prompt tuning) only utilized the concatenation of the part names as the text prompt (i.e., "arm, back, leg, seat, wheel"). In our zero-shot experiments, we incorporated the object category into the text prompt (i.e., "arm, back, leg, seat, wheel of a chair"). However, we recently found that removing them (only using part names) can lead to overall better performance (mIoU from 27.2 to 34.8 for semantic segmentation).

S.5. CLIP vs. GLIP

We have also considered using other pretrained vision-language models, such as CLIP [8], to help with part segmentation tasks. However, the CLIP model mainly focuses on the image classification task and cannot directly generate region-level output (e.g., 2D segmentation masks or bounding boxes). Moreover, as shown in Figure S5, we find that the pretrained CLIP model fails to tell whether an object has a fine-grained part. We conjecture that the CLIP model is pretrained using image-level supervision, with fewer supervision signals about object parts. In contrast, the GLIP model is pretrained on 2D detection and grounding tasks and is thus more sensitive to fine-grained object parts. As a result, the GLIP model is more suitable for our 3D part segmentation task.

S.6. Qualitative Comparison on PartNetE

Figure S6 shows the qualitative comparison between our method and baseline approaches. Our few-shot version (45x8) outperforms all existing few-shot methods and even produces better results than the "45x8+28k" version of PointNext, where the additional 28k 3D shapes are used for training. In particular, our method is good at detecting small object parts (i.e., wheel, bulb, screw, handle, knob, and button). Without any 3D training, our zero-shot version also achieves impressive results.

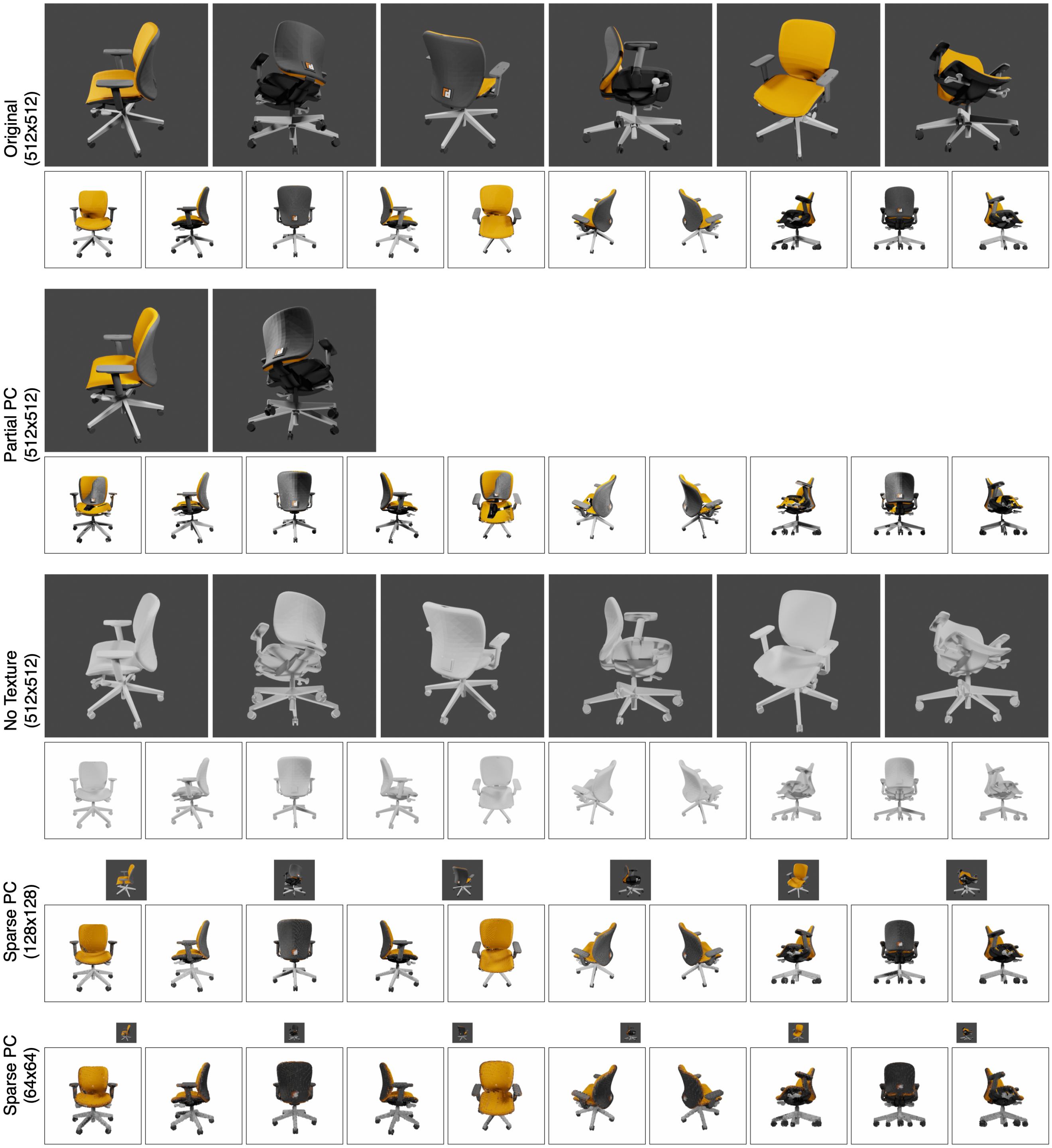

Figure S4. Five variants of input point clouds. For each variant, the first row shows mesh renderings by BlenderProc [1], which are used to fuse and generate the input point cloud. The resolutions of the images are shown in parentheses. The second row shows renderings of the input point cloud by Pytorch3D [9], which are fed to the GLIP model. The image resolution is 800x800. Artifacts of point cloud renderings (last row) can be seen when zoomed in.

S.7. Why not Use ShapeNetSeg?

We acknowledge that ShapeNetSeg [13] is a commonly used benchmark in prior research. However, we would like to note that prior studies have used point clouds sampled from meshes, including interior structures, as input. Our method, on the other hand, focuses on point clouds fused

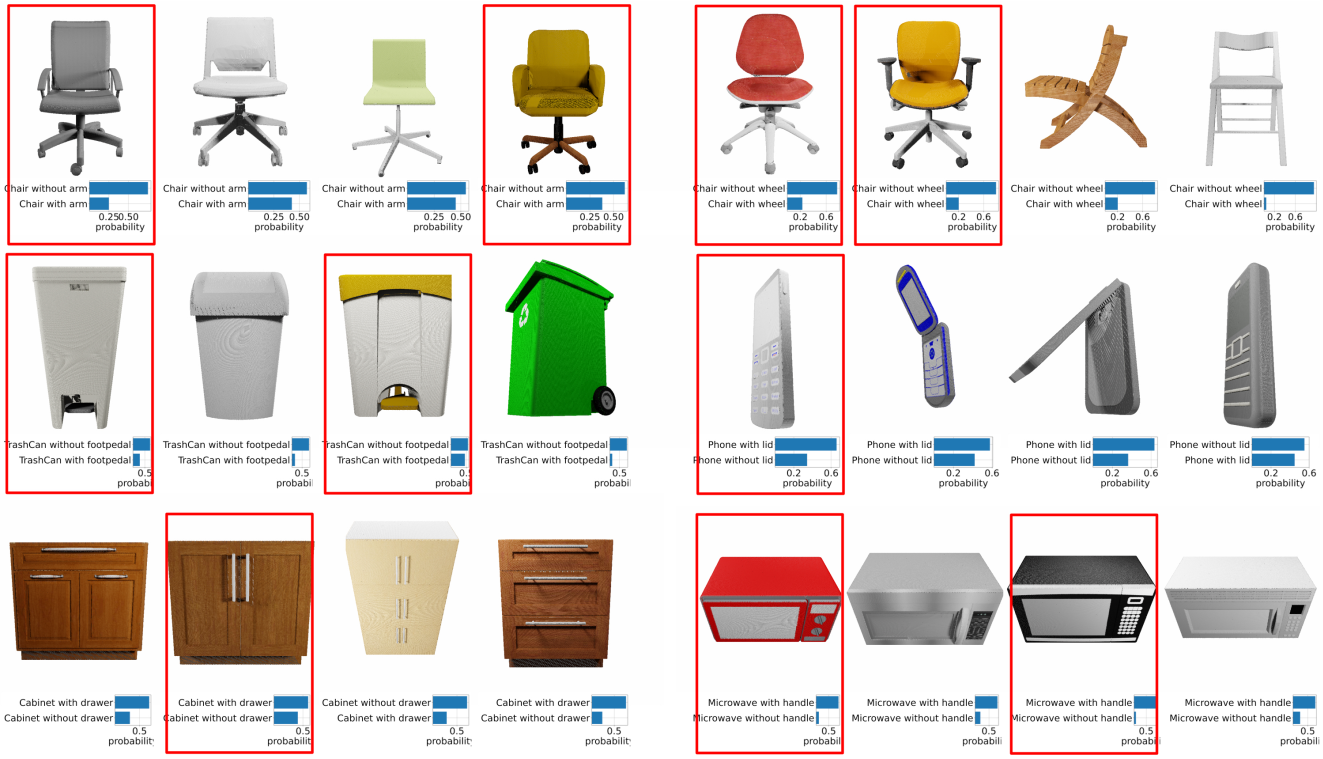

Figure S5. We perform binary classification using CLIP [8]. CLIP fails to identify whether an object has a part. Incorrect predictions are highlighted with red rectangles.

from multiple RGB-D images and cannot handle interior structures. This makes direct comparison with numbers reported in previous papers difficult. To ensure a fair comparison, previous methods would need to be re-run using the same input format (without interior points). Therefore, we focus on a consistent and extensive comparison on PartNetE, a more suitable benchmark for evaluating the generalizable part segmentation. PartNetE is larger, with 45 (vs. 16) object categories and finer-grained parts, and covers most of the categories in ShapeNetSeg. We re-run all baseline methods using a consistent setting on this more challenging benchmark to ensure a fair comparison.

S.8. Details of Baselines

We train baseline approaches on our PartNetE dataset.

PointNet++ and PointNext We use PointNext's official code base to train PointNet++ and PointNext for semantic segmentation under both the "45x8" and "45x8+28k" settings, as described in the main paper. Specifically, we adapt the configurations[1] provided by PointNext and randomly sample 10,000 points per shape as the network input. We use 148-class segmentation heads for both baselines, including 103 part classes and 45 background classes (one for each object category). For PointNext, we utilize a c32 model and take point positions, normals and heights as input. For PointNet++, the model takes point positions and normals as input.

PointGroup and SoftGroup We use SoftGroup's official code base to train PointGroup and SoftGroup for instance segmentation under both the "45x8" and "45x8+28k" settings, as described in the main paper. Specifically, the training includes two stages: 1) training a backbone module for semantic and offset prediction; 2) training the rest modules while freezing the backbone from stage 1. We randomly sample (up to) 50k points for each shape and utilize the point positions and normals as the network input.

For the first stage, there are 104 classes (including 103 part classes and one background class), and points are highly unbalanced across the classes. To avoid losses being dominated by several common part classes, we apply frequency-based class weights, calculated as the inverse square root of point frequency [4], to cross-entropy and offset losses. We also disable data augmentations (e.g., elastic transform) designed for scene-scale datasets. The voxel scale for voxelization is set to 100, and the backbone network is initialized with pretrained checkpoint `hais_ckpt_spconv2.pth`. We train the backbone for

[1] PointNext: https://github.com/guochengqian/PointNeXt/blob/master/cfgs/shapenetpart/pointnext-s.yaml, PointNet++: https://github.com/guochengqian/PointNeXt/tree/master/cfgs/scannet/pointnet++_original.yaml

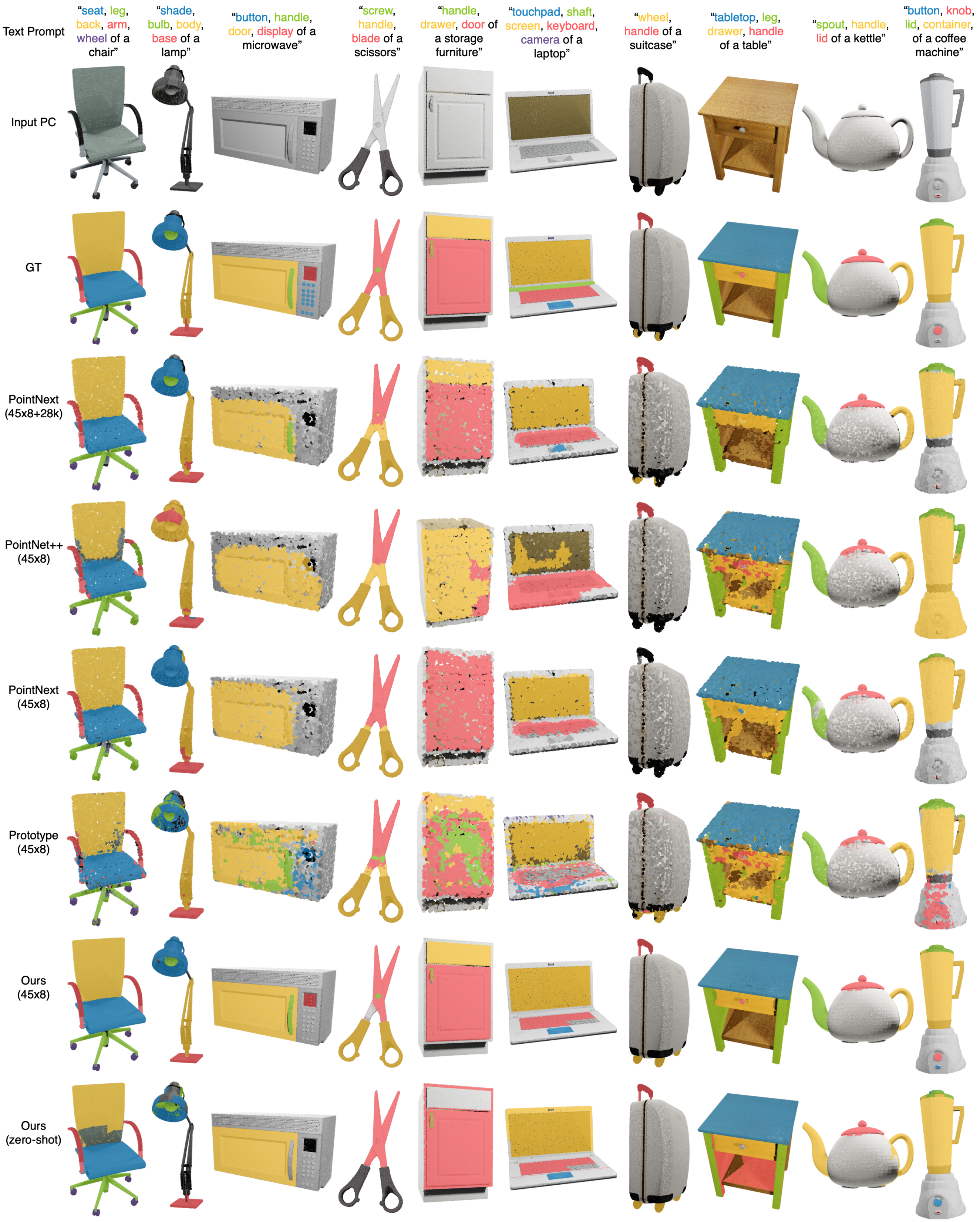
Text Prompt
"seat, leg, back, arm, wheel of a chair"
"shade, bulb, body, base of a lamp"
"button, handle, door, display of a microwave"
"screw, handle, blade of a scissors"
"handle, drawer, door of a storage furniture"
"touchpad, shaft, screen, keyboard, camera of a laptop"
"wheel, handle of a suitcase"
"tabletop, leg, drawer, handle of a table"
"spout, handle, lid of a kettle"
"button, knob, lid, container, of a coffee machine"
Input PC
GT
PointNext (45x8+28k)
PointNet++ (45x8)
PointNext (45x8)
Prototype (45x8)
Ours (45x8)
Ours (zero-shot)

Figure S6. Qualitative comparison between our method and baseline approaches on the PartNetE dataset. Semantic segmentation results are shown. For baseline approaches, we randomly sample 10,000 points as input. "45x8" indicates the few-shot setting, where the model is trained with 8 shapes per object category. "45x8+28k" indicates the setting where the additional 28k shapes are used for training .

Table S2. Full table (1/2) of semantic segmentation results on the PartNetE dataset. Category mIoUs are shown. For 17 overlapping object categories, baseline models leverage additional 28k training shapes in the 45x8+28k setting. For the other 28 non-overlapping object categories, there are only 8 shapes per object category during training.

			few-shot w/ additional data (45x8+28k)			few-shot (45x8)					zero-shot
	category	part	PointNet++ [6]	PointNext [7]	SoftGroup [10]	PointNet++ [6]	PointNext [7]	SoftGroup [10]	ACD [2]	Prototype [14]	Ours	Ours
Overlapping Categories (17)	Bottle	lid	48.8	68.4	41.4	27.0	67.6	20.8	22.4	60.1	**83.4**	76.3
	Chair	arm	83.5	88.6	**89.7**	29.5	68.6	67.8	27.6	58.7	74.1	34.6
		back	89.0	**93.4**	92.2	59.7	89.5	86.5	60.6	83.7	89.7	25.3
		leg	85.5	**94.0**	83.5	51.7	70.0	84.9	42.8	73.0	89.0	76.3
		seat	85.7	**90.5**	81.8	61.0	80.8	76.6	53.4	70.9	81.4	75.3
		wheel	79.7	92.6	**94.4**	9.0	16.7	86.6	10.7	67.9	92.6	92.2
	Clock	hand	19.2	28.4	2.5	0.0	0.0	6.0	0.0	10.5	**37.6**	26.7
	Dishwasher	door	59.3	**81.5**	50.7	55.6	73.9	54.2	50.6	68.6	71.2	20.5
		handle	39.6	**56.8**	55.3	0.0	0.0	30.1	0.0	28.0	53.8	0.0
	Display	base	88.1	**97.1**	94.5	48.9	82.3	50.5	36.9	76.9	97.0	70.1
		screen	80.4	**87.6**	49.6	40.1	78.8	46.1	42.1	73.6	73.9	61.2
		support	66.5	**83.4**	42.3	1.5	0.0	22.6	8.4	51.5	83.4	0.0
	Door	frame	48.2	50.0	42.6	22.6	**65.6**	23.4	23.5	49.1	20.9	1.0
		door	60.2	**75.7**	65.7	38.9	73.3	16.6	33.1	50.1	70.8	7.1
		handle	28.6	5.7	**51.0**	0.0	0.0	8.9	0.0	1.2	30.7	0.0
	Faucet	spout	80.1	**90.4**	82.6	31.2	67.2	50.4	31.4	62.1	79.0	12.7
		switch	54.3	**79.5**	54.1	10.8	33.3	18.5	16.9	29.9	63.8	0.9
	Keyboard	cord	82.3	6.1	78.0	0.0	0.0	57.1	0.0	31.2	**83.9**	74.6
		key	66.7	**83.8**	39.8	31.5	69.2	50.2	52.2	58.5	23.3	0.0
	Knife	blade	35.4	58.7	31.3	22.2	59.7	38.3	39.6	50.4	**65.2**	46.8
	Lamp	base	77.5	72.8	**92.8**	20.5	82.0	48.7	6.0	56.2	90.3	84.5
		body	64.5	65.8	78.2	17.5	64.4	40.5	27.3	59.0	**79.2**	0.0
		bulb	51.4	35.2	**66.3**	0.0	0.0	12.2	0.0	4.4	10.2	12.6
		shade	78.5	85.7	**91.5**	4.1	75.1	52.0	21.5	33.1	84.5	51.3
	Laptop	keyboard	66.4	**70.4**	25.1	22.0	40.6	41.9	20.0	48.3	60.1	48.0
		screen	79.0	**83.0**	33.9	28.4	79.9	42.6	35.5	68.2	62.8	71.2
		shaft	**27.7**	0.0	19.6	0.0	0.0	13.4	0.0	8.7	3.0	0.0
		touchpad	**27.3**	9.1	9.4	0.0	0.0	7.8	0.0	13.6	20.6	11.4
		camera	**76.6**	0.0	4.1	0.0	0.0	0.9	0.0	0.7	2.1	4.5
	Microwave	display	**25.0**	0.0	12.9	0.0	0.0	0.4	0.0	3.3	14.5	5.2
		door	63.6	**75.4**	44.9	25.0	63.9	51.8	26.5	62.0	45.2	39.9
		handle	73.1	86.6	84.8	0.0	0.0	33.2	0.0	37.7	**95.2**	0.0
		button	12.5	0.0	10.4	0.0	0.0	5.3	0.0	4.8	15.9	**21.3**
	Refrigerator	door	56.5	**87.8**	43.3	39.2	83.6	39.7	21.5	72.1	58.4	26.3
		handle	30.3	**64.5**	50.4	0.0	0.0	31.0	0.0	13.6	53.1	14.1
	Scissors	blade	59.0	82.1	**85.2**	44.5	72.7	74.0	52.6	45.4	76.8	65.4
		handle	78.1	89.8	**90.8**	65.2	83.4	79.0	64.7	79.7	86.8	0.0
		screw	12.8	0.0	**52.0**	0.0	0.0	14.0	0.0	3.9	17.4	0.0
	StorageFurniture	door	64.2	**71.9**	69.1	25.2	61.9	21.6	22.5	54.7	56.4	45.8
		drawer	65.6	**80.8**	43.9	0.0	0.0	17.0	0.3	26.7	33.0	26.4
		handle	10.9	52.8	67.6	0.0	0.0	18.0	0.0	9.2	**71.4**	16.2
	Table	door	**71.7**	14.5	33.6	0.0	0.0	0.0	0.0	0.0	0.0	24.7
		drawer	42.3	**55.6**	41.0	8.3	35.0	29.1	22.0	24.9	35.3	35.0
		leg	67.3	**85.0**	64.4	15.8	15.4	45.7	17.7	53.7	66.4	56.4
		tabletop	80.2	**93.8**	74.7	19.7	82.2	55.0	41.1	74.5	79.7	77.7
		wheel	80.0	51.8	58.9	0.0	0.0	0.0	0.0	0.0	61.0	**87.1**
		handle	40.9	11.8	**56.3**	0.0	0.0	19.4	0.0	1.2	12.3	5.2
	TrashCan	footpedal	**82.3**	0.0	1.4	0.0	0.0	0.9	0.0	37.7	0.0	2.4
		lid	55.5	**68.5**	49.7	4.0	59.6	26.9	0.0	60.9	64.8	63.5
		door	**77.4**	0.0	0.0	0.9	0.0	0.0	0.0	0.0	2.1	24.5
	Overall (17)		55.6	**58.5**	50.2	18.1	39.2	32.8	19.2	41.1	**56.3**	31.8

200 epochs with a batch size of 16. We apply cosine learning rate attenuation starting from epoch 45 with an initial learning rate of 0.001.

In the second stage, we train the remaining modules for instance segmentation, while freezing the trained backbone from the first stage. We train the networks with a batch size of 4 and an initial learning rate of 0.004. Since the original code is evaluated on indoor segmentation, we empirically tuned the parameters. Specifically, for the "45x8" setting, the grouping radius, mean active, and classification score threshold are set to 0.02, 50, and 0.001, respectively. For the "45x8+28k" setting, the grouping radius, mean active, and classification score threshold are set to 0.01, 300, and 0.01, respectively. In the "45x8+28k" setting, the few-shot

Table S3. Full table (2/2) of semantic segmentation results on the PartNetE dataset. Category mIoUs are shown. For 17 overlapping object categories, baseline models leverage additional 28k training shapes in the 45x8+28k setting. For the other 28 non-overlapping object categories, there are only 8 shapes per object category during training.

			few-shot w/ additional data (45x8+28k)			few-shot (45x8)						zero-shot
	category	part	PointNet++ [6]	PointNext [7]	SoftGroup [10]	PointNet++ [6]	PointNext [7]	SoftGroup [10]	ACD [2]	Prototype [14]	Ours	Ours
Non-Overlapping Categories (27)	Box	lid	18.6	84.2	8.8	24.5	69.4	24.1	21.1	68.8	**84.5**	57.5
	Bucket	handle	0.0	4.1	25.0	0.0	0.0	18.9	0.0	31.3	**36.5**	2.0
	Camera	button	0.0	0.0	12.6	0.0	0.0	13.9	0.0	6.0	**43.2**	14.2
		lens	13.0	66.4	34.6	19.4	51.9	43.3	20.2	58.0	**73.4**	28.6
	Cart	wheel	6.4	36.3	23.9	11.6	47.7	40.8	31.5	36.8	**88.1**	87.7
	CoffeeMachine	button	**32.6**	0.0	2.4	0.0	0.0	4.3	0.0	0.7	6.4	6.3
		container	29.0	25.8	4.6	7.6	23.0	25.5	2.8	25.9	**51.1**	27.3
		knob	**32.6**	3.6	8.2	0.0	0.0	1.3	0.0	7.8	32.6	17.5
		lid	44.0	42.3	17.8	11.2	45.0	27.6	0.0	45.7	**61.2**	50.3
	Dispenser	head	18.0	20.7	18.3	6.9	34.1	42.8	22.0	45.2	**60.4**	25.0
		lid	6.1	31.2	19.5	7.0	11.0	43.0	16.7	61.6	**87.1**	7.9
	Eyeglasses	body	77.2	93.0	77.8	85.8	**94.1**	74.5	82.6	81.7	84.8	0.6
		leg	75.1	83.2	67.0	71.8	84.6	70.9	73.7	74.0	**91.7**	3.0
	FoldingChair	seat	10.9	**96.4**	14.7	63.4	94.9	89.0	74.2	91.2	86.3	91.7
	Globe	sphere	46.5	92.3	59.0	51.4	88.8	85.1	69.8	88.3	**95.7**	34.8
	Kettle	lid	16.2	24.5	46.9	21.4	54.7	60.2	22.9	58.9	**78.8**	30.9
		handle	16.2	71.3	56.8	33.8	73.1	60.1	43.7	**73.6**	73.5	31.4
		spout	30.2	39.6	68.5	30.5	53.7	61.8	54.0	55.5	**78.6**	0.0
	KitchenPot	lid	25.9	79.6	49.1	44.1	**80.1**	66.8	69.9	76.1	77.7	4.8
		handle	5.7	34.3	41.9	19.3	51.8	42.7	33.8	50.5	**61.5**	4.6
	Lighter	lid	52.4	38.4	32.0	33.6	39.9	40.5	32.3	42.8	**69.9**	69.1
		wheel	15.0	10.5	24.3	0.8	0.0	35.3	0.0	15.4	**57.9**	27.8
		button	37.6	0.0	34.2	0.0	0.0	43.7	0.0	34.0	**66.3**	9.3
	Mouse	button	3.0	0.8	**20.2**	0.0	2.7	4.8	0.0	0.1	16.2	1.6
		cord	33.3	65.0	41.0	0.0	0.0	53.2	0.0	40.7	**66.5**	65.4
		wheel	0.0	0.0	**70.8**	0.0	0.0	31.9	0.0	19.4	49.4	14.0
	Oven	door	32.3	**75.6**	17.2	38.9	73.5	49.7	17.8	68.3	73.1	66.1
		knob	36.4	0.0	10.1	0.0	0.0	21.5	0.0	4.7	**73.9**	0.0
	Pen	cap	42.7	53.3	26.3	8.8	45.4	40.5	10.8	34.0	**68.4**	29.2
		button	50.3	25.6	31.4	0.0	21.0	52.1	0.0	61.0	**74.6**	0.0
	Phone	lid	40.0	**78.7**	0.3	10.3	66.7	2.0	19.7	68.3	74.0	48.5
		button	0.0	0.2	4.4	0.0	0.0	8.2	0.0	2.6	22.8	**23.7**
	Pliers	leg	57.7	**99.6**	74.2	99.3	**99.6**	91.2	83.5	91.0	33.2	5.4
	Printer	button	0.0	0.0	1.2	0.0	0.0	1.6	0.0	0.2	**4.3**	0.8
	Remote	button	3.6	**57.8**	37.1	0.0	0.5	37.5	0.0	29.6	38.3	11.5
	Safe	door	14.0	**76.7**	9.8	32.7	67.0	24.8	28.0	51.9	64.5	34.5
		switch	13.6	0.0	5.8	0.0	0.0	21.7	0.0	5.8	**27.9**	4.3
		button	**68.2**	0.0	0.4	0.0	0.0	0.0	0.0	2.7	4.1	28.4
	Stapler	body	58.3	91.4	83.4	30.4	91.1	83.9	49.8	83.0	**93.6**	2.1
		lid	44.9	**85.7**	76.8	45.7	83.3	80.5	50.2	78.4	76.0	39.6
	Suitcase	handle	6.3	9.3	30.0	6.7	28.9	30.7	26.4	38.9	**84.1**	23.4
		wheel	**75.0**	17.8	6.6	0.0	0.0	28.9	0.0	32.1	56.7	57.0
	Switch	switch	1.8	39.7	21.0	9.3	42.9	31.8	10.3	40.9	**59.4**	9.5
	Toaster	button	23.5	2.7	36.6	0.0	0.0	17.7	0.0	9.0	**58.7**	27.6
		slider	5.9	14.0	16.2	0.0	0.0	11.8	0.0	11.2	**61.3**	0.0
	Toilet	lid	19.5	49.4	12.7	9.4	68.5	27.9	53.4	56.8	**72.6**	35.0
		seat	**62.3**	0.0	2.9	0.0	0.0	6.2	0.0	0.1	21.3	15.4
		button	16.4	0.0	23.2	0.0	0.0	7.6	0.0	1.6	**67.6**	11.4
	USB	cap	54.9	67.2	61.6	21.1	**79.7**	73.9	11.4	72.6	58.1	21.7
		rotation	49.8	**68.6**	26.6	35.7	61.7	38.1	38.9	58.1	50.7	0.0
	WashingMachine	door	1.1	54.5	25.8	8.9	37.9	40.0	20.2	55.4	**63.3**	19.3
		button	0.0	0.0	22.4	0.0	0.0	5.0	0.0	6.7	**43.6**	5.6
	Window	window	26.3	**83.3**	39.2	62.6	83.2	66.4	66.8	76.6	75.4	5.2
	Overall (28)		25.4	**45.1**	30.7	21.8	41.5	41.1	25.6	46.3	**61.3**	24.4
Overall (45)			36.8	**50.2**	38.1	20.4	40.6	38.0	23.2	44.3	**59.4**	27.2

shapes are repeated 50 times in each epoch to mitigate the unbalanced data issue. The PointGroup is trained using a similar pipeline to SoftGroup, except using one-hot semantic results from the first-stage backbone instead of softmax results.

ACD Inspired by [2], we utilize an auxiliary self-supervised task to aid few-shot learning. Specifically, in

Table S4. The full table of instance segmentation results on the PartNetE dataset. Category mAP50s (%) are shown. For 17 overlapping object categories, baseline approaches leverage additional 28k training shapes in the 45x8+28k setting. For the other 28 non-overlapping object categories, there are only 8 shapes per object category during training.

	category	part	45x8+28k		few-shot (45x8)			zero-shot
			Point Group [3]	Soft Group [10]	Point Group [3]	Soft Group [10]	Ours	Ours
Overlapping Categories	Bottle	lid	38.2	43.9	8.0	22.4	**79.4**	75.5
	Chair	arm	94.6	**95.1**	35.9	71.0	67.7	23.9
		back	82.0	73.2	83.8	93.7	**95.4**	30.0
		leg	88.6	**93.6**	92.2	89.9	78.1	30.3
		seat	75.0	85.9	81.4	88.1	85.5	**88.9**
		wheel	98.0	97.7	92.8	95.9	95.5	**99.3**
	Clock	hand	1.0	1.0	1.0	1.0	**14.9**	4.2
	Dishwasher	door	**76.7**	75.0	50.6	55.6	57.4	22.5
		handle	55.6	**56.4**	1.0	26.4	32.9	0.0
	Display	base	95.2	**97.4**	13.2	22.1	94.2	58.3
		screen	46.0	55.4	32.9	49.2	**70.7**	40.5
		support	54.0	53.2	4.1	11.1	**84.0**	0.0
	Door	frame	**36.8**	28.3	2.7	9.8	2.8	1.0
		door	32.4	**34.3**	7.5	5.9	30.7	3.0
		handle	1.0	1.0	1.0	1.0	**20.3**	0.0
	Faucet	spout	85.4	**86.3**	50.7	52.4	61.7	3.1
		switch	**74.5**	72.5	11.2	22.2	47.6	1.5
	Keyboard	cord	42.6	39.7	34.3	21.3	**68.6**	25.0
		key	37.2	**37.7**	16.1	1.0	12.3	1.0
	Knife	blade	19.3	27.2	15.6	10.3	**43.9**	22.1
	Lamp	base	64.3	71.1	8.5	17.9	**89.9**	87.2
		body	48.6	36.5	4.3	11.0	**87.4**	1.0
		bulb	54.5	**59.2**	7.1	1.9	5.9	5.9
		shade	83.5	86.4	19.4	47.0	**90.1**	49.0
	Laptop	keyboard	0.0	0.0	40.1	**53.8**	53.4	42.5
		screen	1.0	1.0	36.3	**61.5**	48.5	59.5
		shaft	1.2	**3.5**	1.0	0.0	2.0	0.0
		touchpad	0.0	0.0	0.0	0.0	**19.7**	9.9
		camera	0.0	0.0	0.0	0.0	**1.0**	0.0
	Microwave	display	4.2	1.0	0.0	1.0	**6.3**	1.0
		door	**62.6**	57.1	0.0	31.0	34.4	19.3
		handle	1.0	1.0	0.0	0.0	**60.4**	0.0
		button	**100.0**	**100.0**	0.0	22.8	3.2	4.0
	Refrigerator	door	**57.1**	54.2	0.0	23.2	31.3	14.3
		handle	19.3	17.2	0.0	9.7	**39.7**	8.6
	Scissors	blade	6.2	6.5	4.5	3.0	**14.1**	4.2
		handle	82.0	**82.9**	41.9	34.5	58.4	0.0
		screw	27.2	**28.4**	8.9	4.6	4.3	0.0
	StorageFurniture	door	**86.9**	85.6	0.0	28.8	24.9	13.5
		drawer	3.9	4.2	0.0	1.5	6.1	**8.0**
		handle	56.4	57.5	0.0	4.6	**67.5**	11.2
	Table	door	44.4	**49.3**	0.0	0.0	0.0	8.2
		drawer	35.7	**36.5**	0.0	0.0	11.3	8.9
		leg	33.8	27.4	0.0	7.7	**45.9**	38.7
		tabletop	81.2	**82.0**	0.0	30.0	64.1	65.7
		wheel	1.0	1.3	0.0	1.1	64.7	**92.6**
		handle	**81.9**	80.8	0.0	46.4	7.6	5.5
	TrashCan	footpedal	34.8	**35.3**	0.0	15.3	0.0	2.3
		lid	0.0	0.0	0.0	1.0	37.8	**38.9**
		door	0.0	0.0	0.0	1.0	1.0	**1.8**
	Overall (17)		41.7	**42.4**	14.6	21.3	**42.5**	20.9

	category	part	45x8+28k		few-shot (45x8)			zero-shot
			Point Group [3]	Soft Group [10]	Point Group [3]	Soft Group [10]	Ours	Ours
Non-Overlapping Categories	Box	lid	7.2	8.6	15.8	19.7	**77.2**	24.2
	Bucket	handle	1.5	1.6	1.0	1.1	**18.2**	5.9
	Camera	button	1.0	1.5	4.5	6.1	**33.8**	11.9
		lens	16.1	0.0	5.0	16.4	**39.9**	4.9
	Cart	wheel	29.2	28.4	28.5	29.8	**83.3**	79.3
	CoffeeMachine	button	1.0	1.0	1.1	0.0	**2.2**	1.8
		container	2.5	4.0	13.6	19.7	**32.8**	7.1
		knob	5.6	5.0	3.3	1.5	**13.5**	7.2
		lid	3.3	1.4	8.9	22.6	**27.6**	19.5
	Dispenser	head	27.5	29.2	39.1	45.4	**46.4**	13.7
		lid	20.5	23.6	22.4	30.2	**80.6**	5.0
	Eyeglasses	body	31.7	39.5	28.1	34.7	**79.5**	1.0
		leg	68.0	62.7	50.3	56.3	**84.9**	1.2
	FoldingChair	seat	16.8	16.8	86.4	79.0	76.7	**87.0**
	Globe	sphere	63.1	63.1	80.2	75.7	**81.0**	18.3
	Kettle	lid	64.0	64.4	65.8	70.0	**76.1**	30.9
		handle	51.4	54.3	45.0	59.0	**78.1**	22.9
		spout	68.5	**72.6**	45.4	61.3	71.9	1.0
	KitchenPot	lid	68.3	68.5	81.4	87.1	**91.5**	1.0
		handle	**50.6**	50.1	32.5	44.3	49.5	1.3
	Lighter	lid	30.7	30.7	0.0	40.6	**45.8**	24.1
		wheel	6.0	5.3	0.0	**47.9**	34.3	16.6
		button	64.1	**67.8**	0.0	63.2	23.6	1.8
	Mouse	button	1.0	1.0	0.0	0.0	**1.7**	**1.7**
		cord	1.0	1.0	0.0	1.0	**66.3**	**66.3**
		wheel	**83.2**	**83.2**	0.0	53.7	50.5	8.9
	Oven	door	26.5	31.9	0.0	19.1	**54.9**	36.4
		knob	1.0	1.0	0.0	1.6	**74.1**	15.4
	Pen	cap	48.2	44.4	0.0	44.3	**51.6**	7.8
		button	16.9	16.9	0.0	10.9	**37.9**	1.0
	Phone	lid	1.0	1.1	0.0	1.2	**37.8**	12.0
		button	1.0	1.0	0.0	1.0	**26.6**	2.8
	Pliers	leg	28.2	**40.4**	6.8	14.5	4.7	5.9
	Printer	button	1.0	1.0	0.0	0.0	**1.3**	1.0
	Remote	button	**23.4**	22.5	0.0	6.2	23.1	3.5
	Safe	door	11.0	12.3	0.0	19.4	**68.4**	28.7
		switch	4.8	5.4	0.0	23.3	**27.4**	3.3
		button	**1.0**	**1.0**	0.0	**1.0**	**1.0**	**1.0**
	Stapler	body	86.6	96.7	52.4	88.0	**100.0**	1.0
		lid	90.0	**91.8**	69.8	78.2	89.7	36.0
	Suitcase	handle	25.5	24.2	0.0	12.9	**64.1**	40.8
		wheel	5.7	2.9	0.0	3.1	25.7	**27.5**
	Switch	switch	7.5	5.6	0.0	21.2	**35.1**	5.6
	Toaster	button	9.0	10.1	0.0	4.5	**31.4**	9.0
		slider	5.0	5.0	0.0	16.9	**45.4**	0.0
	Toilet	lid	5.5	6.1	0.0	37.5	**62.3**	11.0
		seat	0.0	0.0	0.0	1.0	**4.2**	1.9
		button	1.0	1.0	0.0	1.5	**70.3**	18.8
	USB	cap	67.3	**75.7**	0.0	69.0	26.0	23.4
		rotation	16.3	15.0	0.0	**33.3**	29.7	0.0
	WashingMachine	door	25.0	34.3	0.0	41.5	**46.4**	10.9
		button	0.0	0.0	0.0	1.0	**14.1**	3.0
	Window	window	21.2	**26.4**	0.0	4.3	15.6	1.3
	Overall (28)		24.6	**25.6**	16.8	28.4	**46.2**	16.2
Overall (45)			31.0	**31.9**	16.0	25.7	**44.8**	18.0

addition to the 45 × 8 labeled training shapes, we also utilized 1,906 unlabeled shapes for the semi-supervised learning. We use CoACD [11] to decompose the mesh of each 3D shape into approximate convex components using a concavity threshold of 0.05, which results in a median of 18 components per shape. Using the decomposition results, we add an auxiliary contrastive loss to the pipeline of PointNet++ as [2]. The auxiliary contrastive loss encourages points within each convex component to have similar features, while points in different components have different features. For the unlabeled shapes, only the ACD-based contrastive loss is used. For the limited labeled shapes (45×8), both contrastive and original segmentation losses are calculated. To compute the contrastive loss efficiently, we randomly sample 2.5k out of 10k points when calculating pairwise contrastive losses.

Prototype Inspired by [14], we also utilize prototype learning to build a few-shot baseline. Specifically, we construct prototype features using the learned point features (by the PointNext backbone, 96 dim) of 360 few-shot shapes. For each part category, we first sample up to 100 point features as the seed features using the furthest point sampling (FPS) in the feature space. We then group the point features into clusters according to their distances to the seed features. We take the average point features of each group to serve as prototype features, which results in 100 prototype features for each part category. For each test shape, we classify each point by finding the nearest prototype features. Note that we only consider prototype features of parts that the object category may have.

S.9. Full Table of Quantitative Comparison

Table S2 and S3 show the full tables of semantic segmentation results. Table S4 shows the full table of instance segmentation results.

References

[1] Maximilian Denninger, Martin Sundermeyer, Dominik Winkelbauer, Youssef Zidan, Dmitry Olefir, Mohamad Elbadrawy, Ahsan Lodhi, and Harinandan Katam. Blenderproc. *arXiv preprint arXiv:1911.01911*, 2019. 4

[2] Matheus Gadelha, Aruni RoyChowdhury, Gopal Sharma, Evangelos Kalogerakis, Liangliang Cao, Erik Learned-Miller, Rui Wang, and Subhransu Maji. Label-efficient learning on point clouds using approximate convex decompositions. In *European Conference on Computer Vision*, pages 473–491. Springer, 2020. 7, 8, 9

[3] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 9

[4] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European conference on computer vision (ECCV)*, pages 181–196, 2018. 5

[5] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 1

[6] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 7, 8

[7] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *arXiv:2206.04670*, 2022. 1, 7, 8

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 3, 5

[9] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv preprint arXiv:2007.08501*, 2020. 4

[10] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 7, 8, 9

[11] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *arXiv preprint arXiv:2205.02961*, 2022. 9

[12] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 1

[13] 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. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016. 4

[14] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8873–8882, 2021. 7, 8, 10