Supplementary Material: 3D Compositional Zero-shot Learning with DeCompositional Consensus

Muhammad Ferjad Naeem*[1], Evin Pınar Örnek*[2], Yongqin Xian[1], Luc Van Gool[1], and Federico Tombari[2,3]

[1] ETH Zürich, [2] TUM, [3] Google

This document provides additional results and qualitatives in Section 1, further details about C-PartNet dataset in Section 2, additional explanations for the benchmark baselines in Section 3, and a list of assets that are used in this work in Section 4.

1 Additional Results

1.1 Qualitative Results on Seen Object Classes

We report qualitative results on the 16 seen object classes in Figure 1 for Direct Segmentation and top-3 prediction results from the proposed DCC model. We confirm that under supervised segmentation, Direct Segmentation is able to give reasonable results for all seen object classes. This is because the model implicitly learns the part prior for these objects. We see that DCC also achieves comparable results for all of these objects. Since our model explicitly uses the part prior during the inference, it however performs better for small parts such as handle in Microwave and Table. Moreover, we see the the top-2 and top-3 results for DCC are usually classes with very similar parts (*e.g.,* Chair, Clock, Microwave).

1.2 Part Label Agnostic Comparison

Part discovery (segmentation) models aim to discover parts in a label agnostic way and compute recall with the ground-truth part labels as their metric. Different than these, *e.g.,* Learning-to-Group [2], our model learns semantic segmentation, which deals with the task of point-wise semantic part labeling. The recall metric measures the percentage of ground truth parts covered by predicted parts at an Intersection over Union larger than a threshold, where a range of threshold from 0.5 to 0.95 is set and the average is taken. We compare with current state of the art part discovery method Learning-to-Group [2] on C-Partnet in Table 1 with their recall metric. We show that we achieve superior performance in this setting too.

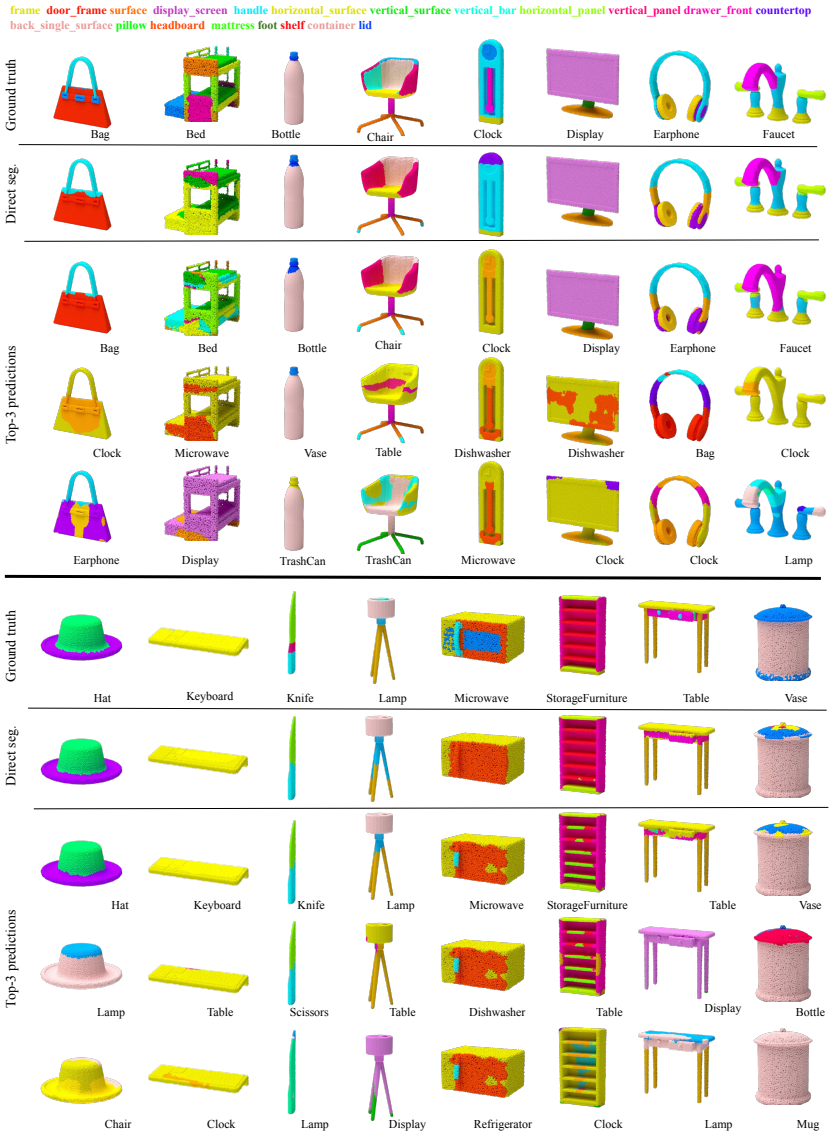

Fig. 1: **Qualitative results for seen object classes** for the Direct Segmentation, and top-3 predictions of the proposed DCC method are provided. We see that Direct Segmentation can give meaningful segmentation for seen object classes because it implicitly learns the part prior for these objects. However, it cannot generalize over the unseen object classes as shown in the main paper. Contrarily, DCC can predict both meaningful classification and segmentation results for both seen and unseen object classes.

Method	Avg	Bowl	Dish	Door	Lap	Mug	Refr	Scis	Trash
L2G [2]	29.6	62.8	7.5	18.0	71.0	50.1	4.5	10.8	12.3
DCC(ours)	**43.7**	**80.3**	**32.0**	**32.8**	**72.9**	**57.5**	**30.8**	**17.8**	**25.8**

Table 1: **Part-label agnostic segmentation result** with recall(%) on C-PartNet.

Method				**Unseen Objects**							
	HM	S	U	Bowl	Dish	Door	Lap	Mug	Refr	Scis	Trash
Without Object Prior	4.4	9.4	2.9	4.3	0.0	0.2	1.2	11.2	0.1	0.1	6.5
With Object Prior	24.8	23.4	26.3	58.2	20.4	**20.8**	18.0	46.8	10.0	**22.0**	14.4
DCC(ours)	**55.9**	**73.2**	**45.2**	**79.8**	**57.1**	5.3	**55.4**	**71.9**	**55.6**	0.0	36.8

Table 2: **3D-PointCapsNet** [8] zero-shot segmentation evaluated with object prior (ground truth object class) gives more reasonable results compared to without it. However, even in Object Prior setting, 3D-PointCapsNet perform worse than DCC.

1.3 3D Point Capsule Network and Object Prior

We present SOTA comparison in Table 1a of the main paper and see that 3D Point Capsule Network [8] perform poorly when predicting segmentation over all part classes. In the supplementary Table 2, we report an additional result of evaluating the capsule with an object prior. This involves only evaluating the segmentation over the parts that are present in the ground truth object class analogous to the Equation 1 of the main paper. We observe that 3D PointCapsNet is unable to provide a reasonable segmentation in the absence of the ground truth object class. When this prior is available, we see more reasonable results. We also see that capsules perform worse in Direct Segmentation than simple models like PointNet[6]. This surprising insight can be an avenue for further research into the capsule based models.

1.4 Door and Scissors oracle and failure case

We provide the oracle (*i.e.,* object prior of ground truth object class) results for the two main failure classes in Figure 2. Even in the oracle setting, the segmentation is poor for Door and Scissors. The performance is even worse when object prior is removed in Direct Seg and DCC. These objects have a large variation with respect to parts from the seen object classes in scale, the number of instances (two blades in Scissors vs one in Knife), and orientation. We hope to inspire future research in PointCloud processing aimed at solving these limitations.

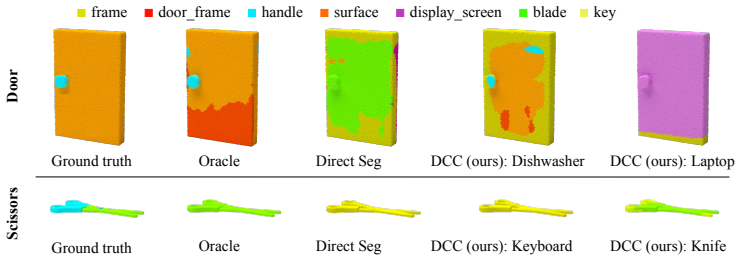

Fig. 2: **Failure cases.** Door and Scissors represent challenging objects with large variations in parts from the seen objects. We compare ground truth part segmentation, oracle (*i.e.,* ground truth object class is given as object prior), direct seg. (*i.e.,* prediction over $\mathcal{P}$ classes without priors) and top-2 predictions from our method. The segmentation model fails even in the oracle object prior setting, indicating that these object classes are already difficult to segment due to large part variation from seen object classes.

1.5 Part Granularity Ablation

In order to measure the effect of number of object categories and part labels, both in terms of seen and unseen classes, we conduct a part-granularity ablation. First, to measure how many seen object categories is needed, we use either 8/16 and 11/16 seen objects classes (in Fig. 3 rows 1,2). Furthermore, to study the impact of number of shared parts, we sample 25,50,75% data from each object class(16/16) equally (in Fig. 3 rows 3,4,5). We observe that dropping half the object classes(r2) results worse than dropping half of the samples per class(r4). With only 75% of data across seen classes, DCC maintains a competitive mIoU. We conclude that the method benefits more from diversity across number of object classes as it sees more variation of parts.

Setup	Avg	Bow	Dish	Door	Lap	Mug	Refg	Sci	Tr
8/16 Seen	23.1	61.0	25.6	0.2	1.8	39.5	23.2	0	33.6
11/16 Seen	25.0	64.0	28.4	0.6	14.6	33.0	25.8	0	33.8
25% Data	25.3	63.8	20.8	2.8	37.6	38.2	6.2	0	32.7
50% Data	27.8	64.6	25.9	10.0	32.6	34.5	22.0	0	32.9
75% Data	30.2	65.2	30.0	**10.2**	38.7	38.4	26.3	0	33.2
Full	**32.7**	**66.1**	**30.9**	5.3	**56.3**	**40.4**	**28.4**	0	**34.2**

Table 3: Granularity study (1) number of seen objects (2) number of parts.

Categories	Classes
Appliances	Microwave, Dishwasher, Refrigerator
Electronics	Display, Keyboard, Laptop
Furniture	Bed, Chair, Clock, Lamp, Table, Storage Furniture, Door
Containers	Bottle, Vase, Bowl, TrashCan, Mug
Cutters	Knife, Scissors
Misc.	Bag, Earphone, Faucet, Hat

(a) **Object class categories.**

	Train		Val		Test		Total
	Obj	Samp	Obj	Samp	Obj	Samp	Samp
Seen	16	16875	16	2426	16	4804	24105
Unseen	-	-	2	895	8	900	1795
Total	16	16875	18	2619	24	5169	25900

(b) **Dataset splits.**

Table 4: **Compositional PartNet** refines the labels of PartNet dataset to maximize shared parts across different object classes, and enables studying 3D-CZSL task. The available 24 object classes are divided into 16 seen classes for training and 8 unseen classes for inference in zero-shot. Object class categories (a) and C-PartNet dataset statistics in train/val/test splits (b) are shown.

2 C-PartNet Dataset Details

We provide further detail about the proposed C-PartNet dataset and its mapping to the PartNet[4].

2.1 Selecting Unseen Object Classes

We divide PartNet objects into functional categories in Table 4a and select the test time unseen object classes accordingly. This ensures that the compositional knowledge between seen and unseen object classes exists. We then select the train, val and test splits accordingly, resulting with the sample statistics as shown in Table 4b. The dataset overall has 25900 samples as PartNet, dividing the total 24 object classes into 18 seen and 8 unseen classes.

2.2 Label Mapping

We provide the full label mapping between C-PartNet and the original Partnet across all 24 object classes in Tables 6, 7, 8, 9. As explained in the main paper, we process the labels of PartNet to maximize compositional overlap between objects. We merge several parts to make them consistent across objects. In total, the part label space of C-PartNet consists of 96 unique parts compared to 128 of the original PartNet.

2.3 Part Statistics for Unseen Object Classes

Unseen object classes in C-PartNet are fully composable from parts of seen object classes. We report the number of instances of these shared parts across

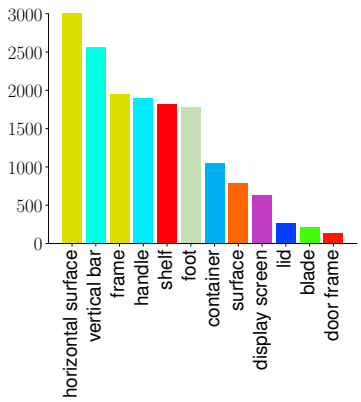

Fig. 3(a): Seen object classes

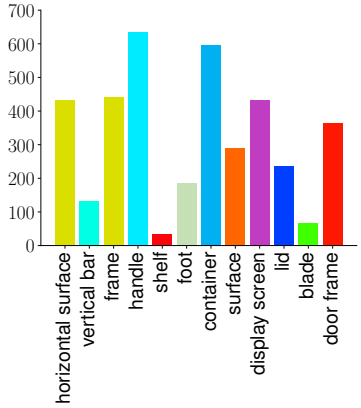

Fig. 3(b): Unseen object classes

Fig. 3: **Part label statistics.** The number of shared parts that compose the unseen object classes in (a) training set of seen object classes and (b) test set of unseen object classes are reported. We see that the part instances have a long tail distribution among training samples. We see a distribution shift from the training instances to part instances among unseen object classes.

the training set and among unseen object classes in the test set in Figure 3. We see in Figure 3a that these parts follow a long tail distribution in the training set. This makes it challenging to recognize parts that are at the tail of the distribution in both seen and unseen object classes. Moreover, when we compare this to the statistics of unseen object classes in Figure 3b, we see that there is a distribution shift between the seen and unseen object classes. These statistics also relate to our per part performance as reported in Figure 5 of the main paper where we saw that some parts have poor performance in unseen object classes. In particular, the failure cases for handle and foot among all unseen object classes is related to the challenge of generalization to new objects while having significant supervision from training set. Blade and door frame in unseen objects Scissors and Door have an additional challenge of having only few samples for supervision. This adds another dimension to the failure case shown in Figure 2.

2.4 Similarity of Unseen Objects to Seen Objects

We train a PointNet classification model on the seen object classes. From this model, we use the average feature representation of each seen and unseen object class to compute pairwise cosine similarity and report the top-3 nearest neighbor seen objects to unseen objects in Table 5. We reaffirm our observations from Figure 4 of the main paper regarding Direct Segmentation. We saw in the main paper that for unseen object classes, Direct Segmentation uses parts of the most geometrically similar seen object classes. In Table 5, among the Container categories, nearest neighbors for Bowl, Mug and Trashcan are Vase and

Unseen Object	Nearest Neighbor Seen Objects
Bowl	Vase, Bottle, Lamp
Dishwasher	Storage Furniture, Microwave, Vase
Door	Clock, Display, Storage Furniture
Laptop	Bed, Chair, Table
Mug	Vase, Lamp, Bottle
Refrigerator	Storage Furniture, Microwave, Vase
Scissors	Keyboard, Table, Storage Furniture
Trashcan	Vase, Bottle, Storage Furniture

Table 5: **Nearest neighbor similarities.** We compute the 3 nearest neighbors between seen object classes for unseen object classes.

Bottle, which share parts with corresponding unseen classes. The harder unseen objects Dishwasher, Laptop and Refrigerator, have nearest neighbors Storage Furniture, which do not share large part similarities but instead share geometric similarities. Finally, for the most challenging Door and Scissors, the nearest neighbors are the classes that do not share neither part labels or geometric similarities with them (*e.g.,* Keyboard and Table for Scissors). Current point cloud models are designed to capture structures based on geometric similarities, which also explain our observations. We advice future works to exploit part relations among instances of the same and different object classes as a potential avenue for increased compositionality in unseen objects.

3 Baselines

We benchmark several baselines against the proposed model, DeCompositional Consensus (DCC), in the main paper. These methods are adapted from the original works to fit our 3D-CZSL task. We provide additional details about the baselines in this section.

PartPred[1] trains a part prediction network from the global feature of each point cloud. The model outputs a binary vector of length $|\mathcal{P}|$ of what parts are found for the input point cloud. The part prediction network consists of an MLP with a similar configuration to DCC trained with a Binary Cross Entropy loss. The labels for this network are extracted according to if a part exists in the ground truth segmentation or not. The predicted parts are used to generate a segmentation mask during inference. This baseline represents a point cloud segmentation model with a separate part prediction branch.

SPNet[7] learns classification by projecting the global features learned by the network on a pretrained distribution where both seen and unseen objects lie. We use Word2Vec[3] to initialize the model. The classification model is trained with Cross Entropy Loss over $\mathcal{O}_s$.

CGE[5] proposes to model compositional relations in a graph. We reformulate CGE as a multitask compositional problem. The graph consists of nodes representing all parts $\mathcal{P}$ and object classes $\mathcal{O}$. A part is connected to an ob-

ject node if that part is contained in the part prior $\mathcal{P}_o$ of that object. Object nodes are also connected with each other if they share parts. The input graph is initialized with a Word2Vec[3] model and processed by a Graph Convolutional Network (GCN). From the output graph, the object nodes are used as classification weights and part nodes are used as the weights for part segmentation. This allows for information propagation from the seen object classes to the unseen object classes through the dependency structure defined in the graph. The classification task is trained with Cross Entropy Loss over $\mathcal{O}_s$ and the segmentation task is trained with our Compositional Part Segmentation framework.

PartPred DCC uses the part prediction model from PartPred and computes the DCC score for classification. This is done by using the Parts found in each segmentation hypothesis in the Hypothesis Bank and computing the consensus score using the part wise scores from PartPred.

3D Point Capsule Network[8] is designed as an alternative point cloud processing method especially to capture the part-whole relationships. The latent capsules are learned within an auto-encoder model through the dynamic agreement-by-routing algorithm. The final segmentation prediction is done by training a single layer MLP with the learned latent capsules. The authors train the model with a one-hot categorical vector representing the ground truth object as an input as introduced in the method section. We remove this input of the ground truth object prior and train the model for seen objects over all parts. The 3D PointCapsNet is trained with 128 latent capsules with 128 feature dimension per each capsule.

4 Assets Used

The following open source assets contributed to this work.

Dataset.

- https://partnet.cs.stanford.edu/

Open source code repositories.

- https://github.com/fxia22/pointnet.pytorch
- https://github.com/tiangeluo/Learning-to-Group
- https://github.com/mutianxu/GDANet
- https://github.com/aboulch/ConvPoint
- https://github.com/ExplainableML/czsl
- https://github.com/subhc/SPNet
- https://github.com/yongheng1991/3D-point-capsule-networks
- https://github.com/facebookresearch/pytorch3d
- https://github.com/isl-org/Open3D

Rendering tool.

- https://github.com/mitsuba-renderer/mitsuba2

References

1. Lampert, C.H., Nickisch, H., Harmeling, S.: Learning to detect unseen object classes by between-class attribute transfer. In: CVPR (2009) 7
2. Luo, T., Mo, K., Huang, Z., Xu, J., Hu, S., Wang, L., Su, H.: Learning to group: A bottom-up framework for 3d part discovery in unseen categories. ICLR (2020) 1, 3
3. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: NeurIPS (2013) 7, 8
4. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: CVPR (2019) 5
5. Naeem, M.F., Xian, Y., Tombari, F., Akata, Z.: Learning graph embeddings for compositional zero-shot learning. In: CVPR (2021) 7
6. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. CVPR (2017) 3
7. Xian, Y., Choudhury, S., He, Y., Schiele, B., Akata, Z.: Semantic projection network for zero-and few-label semantic segmentation. In: CVPR (2019) 7
8. Zhao, Y., Birdal, T., Deng, H., Tombari, F.: 3d point capsule networks. In: CVPR (2019) 3, 8

Object class	Part Labels	Changed from
Bag	bag body	-
	handle	-
	shoulder strap	-
Bed	bed_post	-
	frame_horizontal_hard_surface	-
	headboard	-
	horizontal_bar	frame_horizontal_surface_bar bar_stretcher bed_side_surface_horizontal_bar rung
	horizontal_surface	surface_base
	leg	-
	mattress	-
	pillow	-
	vertical_bar	bed_side_surface_vertical_bar ladder_vertical_bar
	vertical_panel	bed_side_surface_panel
Bottle	closure	-
	container	jug_body normal_bottle_body
	handle	bottle_handle jug_handle
	lid	-
	mouth	-
	neck	-
Bowl	container	container containing_things
	foot	bottom
Chair	arm_sofa_style	-
	back_connector	-
	back_single_surface	-
	back_support	-
	caster_stem	-
	central_support	star_leg_central_support pedestal_base_central_support
	chair_base	-
	connector	arm_connector
	foot	-
	frame	back_holistic_frame arm_holistic_frame seat_holistic_frame seat_frame_bar seat_surface_bar
	head_connector	-
	headrest	-
	horizontal_bar	back_surface_horizontal_bar back_frame_horizontal_bar arm_horizontal_bar bar_stretcher
	horizontal_surface	arm_writing_table seat_single_surface seat_surface
	leg	-
	pedestal	-
	rocker	-
	runner	-
	seat_support	-
	star_leg_base	star_leg_base_leg star_leg_base_knob star_leg_base_lever
	vertical_bar	back_surface_vertical_bar back_frame_vertical_bar arm_near_vertical_bar
	wheel	-

Table 6: Label mapping from C-PartNet to PartNet [1/4]

Object class	**Part Labels**	**Changed from**
Clock	box	-
	chain	-
	foot	-
	frame	-
	horizontal_surface	base_surface
	pendulum_body	-
	pendulum_clock_top	-
	surface	-
Dishwasher	door_frame	-
	foot	-
	frame	-
	handle	-
	surface	foot_base_surface surface_base
Display	display_screen	-
	foot	base_support
	surface	-
Door	door_frame	outside_frame
	handle	handle_fixed_part handle_movable_part
	surface	surface_board
Earphone	connector_wire	-
	earbud_connector	-
	earbud_connector_wire	-
	earbud_frame	-
	earbud_pad	-
	earcup_connector	-
	earcup_frame	-
	earcup_pad	-
	top_band	-
Faucet	horizontal_support	-
	horizontal_surface	surface_base
	hose	-
	switch	-
	tube	-
	vertical_support	-
Hat	bill	-
	brim	-
	button	-
	crown	-
	panel	-
Keyboard	frame	-
	key	-
Knife	blade	-
	bolster	-
	butt	-
	guard	-
	handle	-

Table 7: Label mapping from C-PartNet to PartNet [2/4]

Object class	Part Labels	Changed from
Lamp	body	lamp_pole lamp_body_solid lamp_post
	chain	-
	connector	-
	cord	-
	lamp_arm	-
	lamp_arm_curved_bar	-
	lamp_arm_straight_bar	-
	lamp_base_part	lamp_base_part street_lamp_base
	lamp_body	-
	lamp_body_vertical_panel	-
	lamp_cover	lamp_cover_frame_top lamp_cover_frame_bottom lamp_cover_frame_bar lamp_cover_holder
	lamp_finial	-
	lamp_shade	-
	lamp_wireframe_fitter	-
	leg	-
	light_bulb	-
Laptop	display_screen	screen_side
	horizontal_surface	base_side
Microwave	door_frame	-
	foot	-
	frame	-
	handle	-
	tray	-
Mug	container	body containing_things
	handle	-
Refrigerator	door_frame	-
	foot	-
	frame	-
	handle	-
	shelf	-
	surface	-
Scissors	blade	-
	handle	-
Storage Furniture	caster_stem	-
	countertop	-
	drawer_back	-
	drawer_front	-
	drawer_side	-
	foot	-
	handle	-
	hinge	-
	horizontal_bar	frame_horizontal_bar
	horizontal_panel	top_panel bottom_panel bottom_panel
	horizontal_surface	drawer_bottom base_side_panel
	shelf	-
	vertical_bar	frame_vertical_bar
	vertical_panel	back_panel vertical_side_panel vertical_front_panel vertical_divider_panel
	vertical_surface	cabinet_door_surface
	wheel	-

Table 8: Label mapping from C-PartNet to PartNet [3/4]

Object class	**Part Labels**	**Changed from**
Table	bar	-
	caster_stem	-
	central_support	-
	circular_bar	circular_stretcher
	drawer_back	-
	drawer_front	-
	drawer_side	-
	foot	-
	handle	-
	horizontal_bar	bar_stretcher
	horizontal_panel	bottom_panel
	horizontal_surface	ping_pong_net tabletop_surface pool_ball tabletop_surface tabletop_surface glass bar board drawer_bottom
	keyboard_tray_surface	-
	leg	-
	pedestal	-
	runner	-
	seat_support	bench_connector bench
	shelf	-
	star_leg_base	leg
	tabletop_connector	-
	tabletop_dropleaf	-
	vertical_panel	back_panel vertical_side_panel vertical_front_panel vertical_divider_panel
	vertical_surface	cabinet_door_surface
	wheel	-
TrashCan	container	container_bottom container_box container_neck
	foot	-
	frame	frame_horizontal_circle frame_bottom frame_holistic
	lid	cover_support cover_lid
	vertical_bar	frame_vertical_bar
Vase	container	-
	foot	-
	lid	-
	liquid_or_soil	-
	plant	-

Table 9: Label mapping from C-PartNet to PartNet [4/4]