Supplementary material for 3D Part Segmentation via Geometric Aggregation of 2D Visual Features

Marco Garosi
University of Trento
marco.garosi@unitn.it

Riccardo Tedoldi
University of Trento
riccardo.tedoldi@unitn.it

Davide Boscaini
Fondazione Bruno Kessler
dboscaini@fbk.eu

Massimiliano Mancini
University of Trento
massimiliano.mancini@unitn.it

Nicu Sebe
University of Trento
sebe@disi.unitn.it

Fabio Poiesi
Fondazione Bruno Kessler
poiesi@fbk.eu

A. Introduction

We provide additional material in support of our main paper. This document is organised as follows:

- In Appendix B, we describe the steps involved in COPS and we show them qualitatively on some point clouds.
- In Appendix C, we provide implementation details of COPS and specify the hyper-parameter used in our experiments to facilitate reproducibility.
- In Appendix D, we ablate the role of the specific layer of the DINOv2-Base architecture from which we perform feature extraction.
- In Appendix E, we investigate the role of the number of spatial and semantic nearest neighbours used in the Geometric Feature Aggregation (GFA) module.
- In Appendix F, we provide additional qualitative results on ScanObjectNN [9] and FAUST [2] datasets.
- In Appendix G, we provide details on the computational resources utilised.

B. Pipeline visualisation

In Fig. 1, we provide a detailed visualisation of the different steps required by COPS on four point clouds from ShapeNetPart [11]. The first three columns illustrate the feature extractor Φ, which processes the input point cloud depicted in the first column, extracts features using DINOv2 [6] (second column), and modifies them using the Geometric Feature Aggregation (GFA) module (third column). The next three columns illustrate the segmenter Ψ, which decomposes the object into parts via feature clustering (fourth column, colours are not informative) and assigns each part a semantic label (sixth column) by leveraging PointCLIPv2 [12] predictions (fifth column) as semantic anchors. The last column displays the ground-truth segmentation. The minimal difference between the last two columns suggests that COPS produces very accurate segmentations, despite the low quality of the PointCLIPv2 predictions shown in the fifth column.

C. Implementation details

In this section, we discuss implementation details and the hyper-parameters of COPS.

Point cloud processing. We perform rendering at the original point cloud resolution to retain finer details. Then, we randomly sample 10,000 points from each object, we update the pixel-to-point mappings utilised to back-project features to 3D, and we project them back. Next, GFA performs farthest point sampling (FPS) to find super points for feature aggregation. Subsequently, we randomly sample 2,048 points to obtain semantic labels via PointCLIPv2. Lastly, we perform clustering on these sampled points and we assign each cluster a semantic label via Hungarian with PointCLIPv2's predictions.

Rendering. We utilise PyTorch3D [8] for rendering. Notably, we set: (i) camera orientations, (ii) point size, and (iii) rendering canvas size. We have defined three camera settings: 6 orthogonal cameras, facing front, back, left, right, top, and bottom in Fig. 2(a); 10 cameras, following PointCLIPv2 [12] in Fig. 2(b); 48 cameras in Fig. 2(c). We set the point size to small values for datasets whose point clouds are *dense*, *i.e.*, containing many points. We enlarge the point size for *sparse* datasets, such as ShapeNetPart, to obtain smooth renders. Lastly, we set the canvas size to the input size of DINOv2 of 224×224 pixels, ensuring no scaling and/or cropping is required. When photometric (RGB) information is not available, we render depth maps. We utilise both the depth maps produced by PyTorch3D, where light pixels correspond to close points, and the depth maps in PointCLIPv2's style, where the dark pixels correspond to points close to the camera. We found COPS to perform the best with PyTorch3D's depth maps.

Feature extraction. DINOv2 is based on the vision trans-

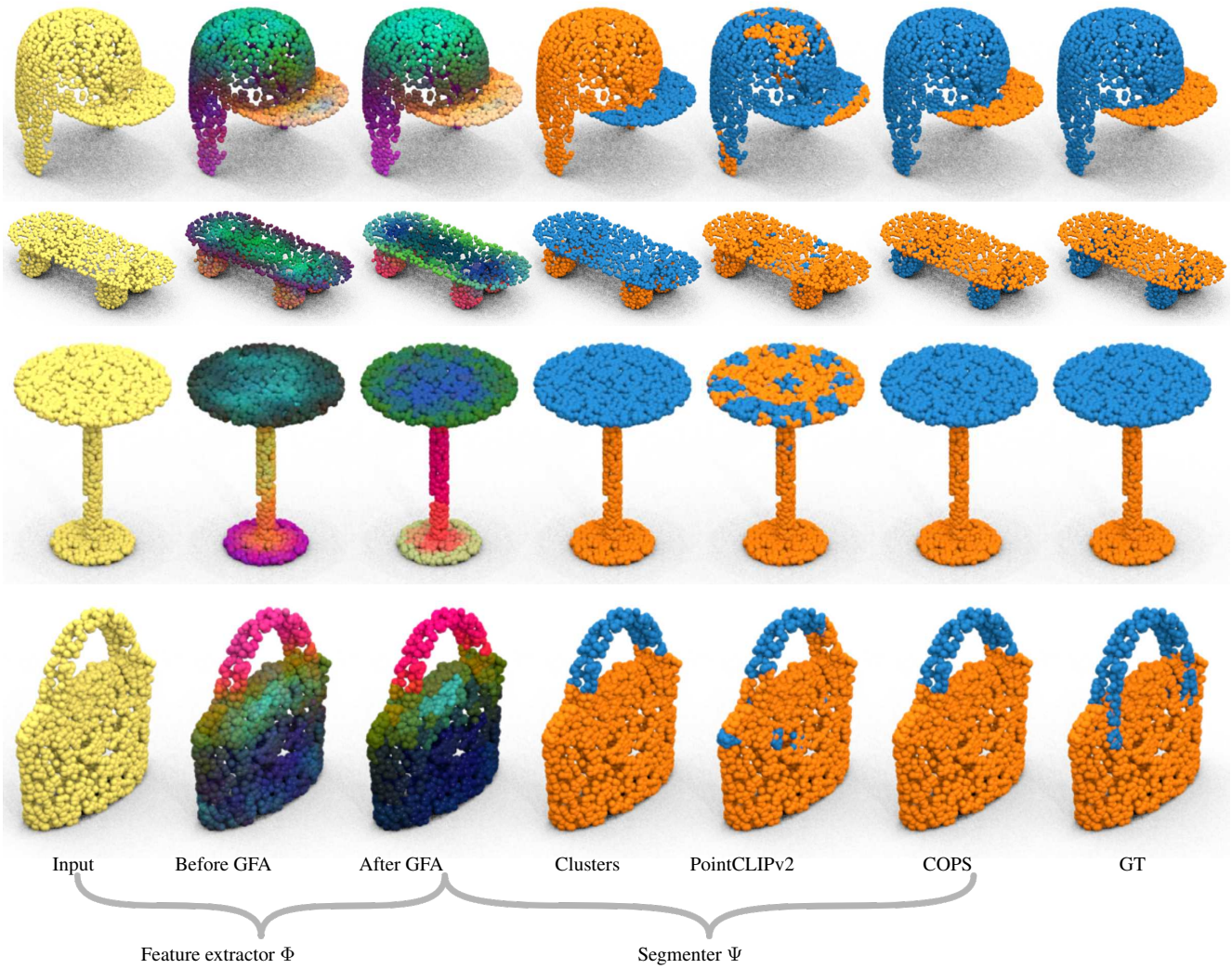
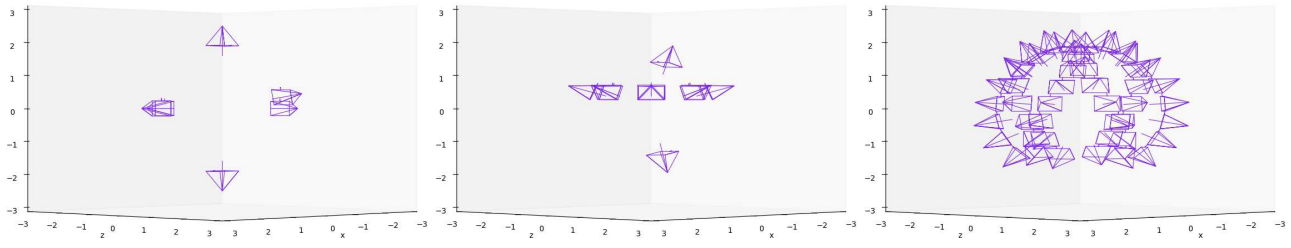


Figure 1. Detailed visualisation of the steps required by COPS. From left to right: input point cloud, intermediate features obtained by 3D-lifting DINOv2 features, final features obtained with GFA, part decomposition obtained via feature clustering (colours are not informative because cluster labels are not semantic), PointCLIPv2 predictions, COPS predictions, and ground-truth segmentation. By disentangling part decomposition (fourth column) from semantic label assignment, COPS can leverage noisy PointCLIPv2 predictions (fifth column) to produce accurate segmentations (sixth column).



(a) 6 views (b) 10 views (c) 48 views

Figure 2. Cameras visualised in 3D space. (a) shows the 6-camera setting, where the cameras are orthogonal to one another. (b) shows the 10 cameras adapted from PointCLIPv2 [12]. They capture the object from more sides, while the top and bottom views are similar to those in (a). (c) shows the largest setting, with 48 cameras sampled bottom to top on eight arches around the object. While this configuration provides more views than (a) and (b), there are many redundant views which contribute little to the overall performance, as we have shown in the corresponding ablation study.

former architecture [3, 10]. It splits the image into patches of 14×14 pixels and outputs a feature vector for each patch. However, we need pixel-level, or dense, features to perform lifting to 3D. Therefore, we upsample the feature maps to the input image size of 224×224 pixels using bicubic interpolation.

Geometric feature aggregation (GFA). GFA works in three steps: (i) it samples super points, (ii) it aggregates features via either spatial or semantic attention, and (iii) it upsamples super point features to the whole point cloud. GFA has two hyper-parameters: the number of super points and the number of neighbours considered in the aggregation step. For the first, too many super points may lead to reduced spatial/semantic consistency, while too few can make the features collapse, not accounting for the local variability of the point cloud. By default, we sample 256 super points. For the second, the more the points and the larger the context window used to compute the super point's feature. By default, we set it to 10 for spatial attention and to 90 for semantic attention. In Sec. E, we conduct an ablation study on these hyper-parameters.

D. Ablation study on DINOv2-Base layers

Following FoundPose [7], in Tab. 1 we evaluate COPS with different patch descriptors. We utilise DINOv2-Base (ViT-B) and sample patch-level features at different levels, showing how performance changes. We observe an increment in performance as we utilise increasingly higher-level patch descriptors, which encode more abstract semantic information, *e.g.*, about parts [6]. We note that performance improves in certain categories, such as "airplane", as we move towards higher-level representations. However, we find that for other categories, such as "mug" or "knife", lower-level representations may provide the optimal level of abstraction for achieving accurate part segmentation results. This highlights the need for further exploration in future work, assessing how to consider representations at different scales to achieve consistent improvements.

E. Ablation study on GFA

In Tab. 2(a), we conduct an additional ablation study on the GFA module. We evaluate 13 distinct configurations, including the standard GFA employed in the main paper. Specifically, we vary the number of sampled super points, the number of neighbours taken into account during the attention operation, *i.e.*, the "context window", and weighting by distance. In Tab. 2(b) we repeat these experiments by swapping the order of spatially- and semantically-consistent feature aggregation. The results show that our default configuration performs the best. We observe that increasing the number of neighbours reduces performance, thus suggesting that a smaller "context window" allows GFA to aggregate only the most relevant features. Raising the number of super points leads to a decrease in performance because it limits the effect of GFA. If all the points are kept as super points, GFA has no effect, while if too few super points are sampled, features can collapse. Lastly, performing spatially-before semantically-consistent feature aggregation makes GFA capture geometric knowledge better, thus achieving higher performance.

F. Additional qualitative results

Following our main paper, we show additional qualitative results on other datasets. Fig. 4 shows qualitative results on FAUST [2], using annotations from SATR [1]. Fig. 3 shows qualitative results on ScanObjectNN [9] in the most challenging OBJ-BG setting. Objects such as "bed" or "sofa" pose challenges in distinguishing between the individual parts due to overlapping geometry or intricate designs. Moreover, the real-world point clouds in ScanObjectNN [9] are noisy and can contain several occlusions, making it difficult to separate them into distinct parts.

G. Resources used

Our training-free method does not require extensive computational resources and is designed to be computationally efficient. We run all our experiments on a consumer desktop NVIDIA RTX 3060 GPU with 12GB of VRAM and a laptop NVIDIA RTX 2070 Super Max-q with 8GB of VRAM. Evaluation time on the NVIDIA RTX 3060 GPU took approximately: 40 seconds on FAUST [1]; 1 hour on ScanObjectNN [9]; 3 hours on PartNetE [4]; 9 hours on ShapeNetPart [11]; 12 hours on PartNet [5]. All the reported inference times are doubled if running the inference on the consumer laptop with the NVIDIA RTX 2070 GPU with 8GB of VRAM. Further optimisations, such as pre-rendering views for all objects, can be introduced to lower test times. However, they can take up a large amount of storage space.

References

[1] Ahmed Abdelreheem, Ivan Skorokhodov, Maks Ovsjanikov, and Peter Wonka. SATR: Zero-Shot Semantic Segmentation of 3D Shapes. In *ICCV*, 2023. 3, 6

[2] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. FAUST: Dataset and evaluation for 3D mesh registration. In *CVPR*, 2014. 1, 3, 6

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 3

[4] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. Partslip: Low-shot part seg-

	ViT-B layer	mIoU_I	mIoU_C	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skate	Table
1	0	49.7	46.4	31.1	53.0	45.7	27.9	47.5	50.8	60.7	66.6	45.5	71.3	21.6	42.9	35.6	31.0	48.8	61.8
2	1	55.9	52.1	31.5	72.9	65.2	29.9	60.6	56.0	47.7	62.9	45.9	90.0	22.6	58.4	39.8	29.5	50.0	70.3
3	2	58.4	51.6	32.3	65.2	49.2	32.2	66.0	55.7	50.9	68.9	46.8	90.6	23.9	43.9	42.7	31.4	52.5	72.7
4	3	58.8	52.5	33.1	64.4	54.5	32.9	66.4	55.3	48.7	72.6	47.2	92.0	23.7	53.4	40.1	31.2	52.1	72.7
5	6	60.5	55.2	36.8	53.8	65.4	32.2	68.0	60.8	66.6	78.5	48.4	91.1	24.6	36.5	48.9	45.7	53.3	71.8
6	7	61.0	56.4	41.1	59.9	64.4	31.8	68.2	63.5	67.2	80.1	48.4	88.9	24.8	41.1	49.6	47.0	55.6	71.0
7	8	61.4	56.3	43.2	60.7	61.6	31.8	68.4	64.7	67.1	77.5	48.2	90.1	25.1	40.3	49.8	45.8	55.1	71.7
8	10	62.6	59.0	50.7	64.7	71.1	30.5	69.0	66.2	70.2	81.6	48.7	77.9	25.3	60.1	52.2	47.6	56.3	71.2
9	11	63.3	60.1	50.6	70.7	69.8	29.7	70.0	66.0	66.2	79.7	49.0	86.8	25.9	68.2	53.5	46.0	57.5	72.1
10	12 (w/o norm.)	64.0	61.1	50.9	75.8	69.5	29.7	71.2	66.4	68.8	76.5	48.8	89.2	25.1	72.4	56.4	47.6	56.8	72.8
11	12 (w/ norm.)	64.4	60.9	51.3	71.0	69.7	29.9	71.7	66.4	72.6	80.2	48.6	91.1	26.1	62.5	54.3	46.7	59.4	72.9

Table 1. Part segmentation performance on ShapeNetPart [11] obtained when extracting DINOv2 base features from different layers, *i.e.*, at different depths. DINOv2 base is based on ViT-B, and has 13 layers. The highlighted row corresponds to the results shown in the main paper.

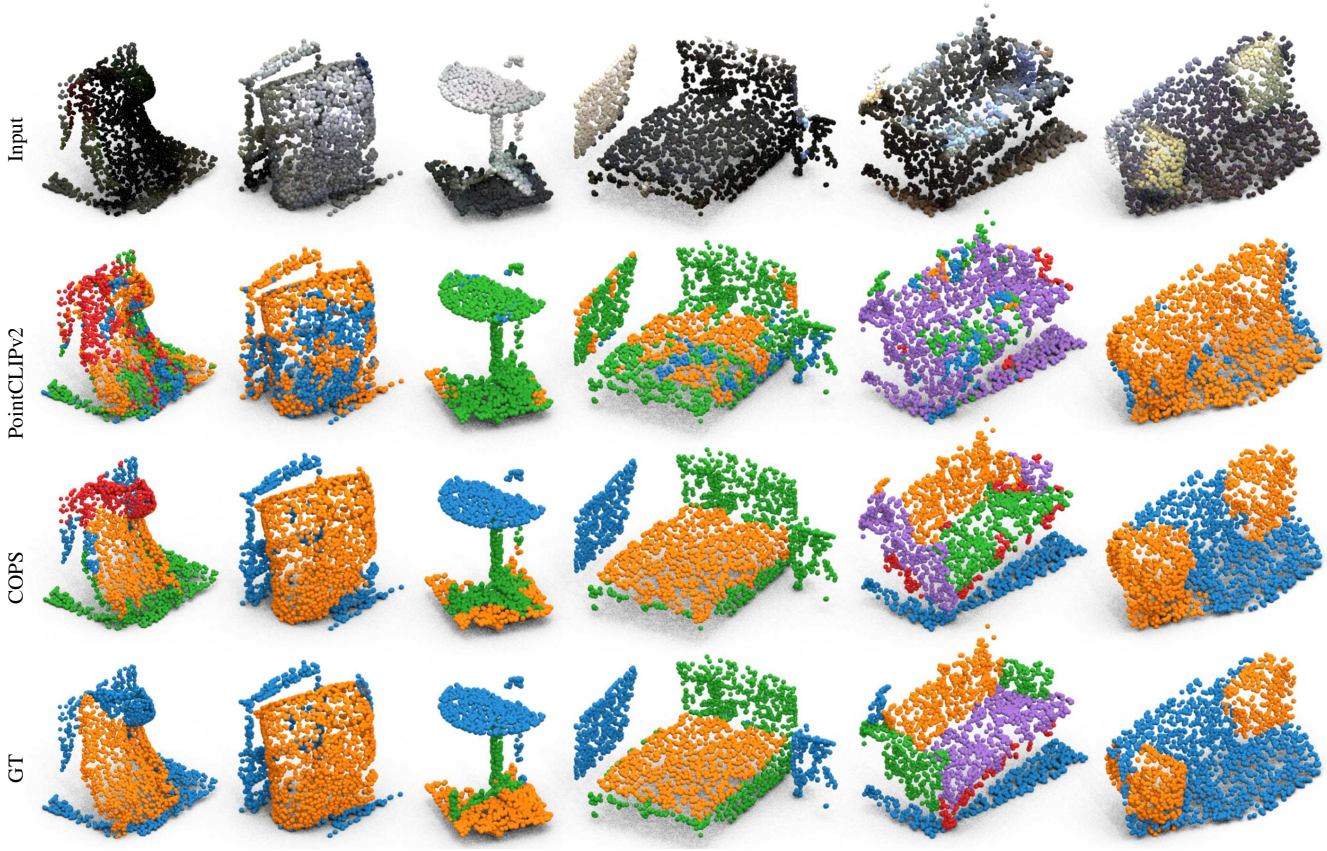


Figure 3. Qualitative results on ScanObjectNN [9]. Top to bottom: input point cloud with RGB colours, PointCLIPv2 predictions, COPS predictions, and ground-truth segmentation.

mentation for 3d point clouds via pretrained image-language models. In *CVPR*, 2023. 3

[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 *CVPR*, June 2019. 3

[6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel

		Sup. spat.	Sup. sem.	Nei. spat.	Nei. sem.	$mIoU_I$	$mIoU_C$	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skate	Table
Superpoints	1	512	256	10	90	64.2	60.0	50.6	68.3	69.2	30.1	71.6	65.4	74.2	78.6	48.9	90.9	26.3	58.1	52.9	45.5	57.3	72.7
	2	256	512			64.3	60.7	51.0	70.6	70.0	29.7	71.5	64.8	72.0	79.0	48.9	91.6	26.1	63.3	53.8	46.4	59.5	73.0
	3	512	512			64.3	60.8	50.9	71.4	70.1	29.7	71.2	65.8	73.7	78.6	48.5	91.2	26.2	62.9	52.3	48.0	59.6	73.1
	4	128	256			64.3	61.0	51.0	68.9	72.1	30.0	71.7	66.0	70.5	79.0	48.9	91.4	25.5	64.3	55.9	49.0	59.4	72.8
	5	256	128			64.2	60.3	51.4	66.0	72.0	29.9	71.7	65.2	70.9	78.9	48.9	91.0	25.8	62.9	53.1	48.1	57.2	72.6
	6	128	128			64.3	60.5	50.7	73.3	67.0	29.6	71.5	66.5	72.1	80.3	49.1	91.0	25.8	55.7	56.2	47.7	59.4	73.0
Neighbours	7	256	256	10	90	64.4	60.9	51.3	71.0	69.7	29.9	71.7	66.4	72.6	80.2	48.6	91.1	26.1	62.5	54.3	46.7	59.4	72.9
	8			90	90	61.9	58.6	49.6	62.8	68.9	28.0	67.8	64.0	73.3	80.3	48.5	89.0	24.9	58.9	54.9	42.1	55.0	69.5
	9			170	256	58.6	54.9	49.0	59.5	69.3	25.3	62.8	60.4	70.5	79.5	48.3	86.6	23.9	32.1	53.5	46.4	46.3	65.6
	10			90	10	61.2	58.5	49.1	68.0	66.7	27.3	68.2	64.7	69.6	79.3	48.4	90.7	24.6	54.5	53.8	46.5	55.0	69.2

(a) Spatially-consistent aggregation followed by semantically-consistent aggregation

		Sup. sem.	Sup. spat.	Nei. sem.	Nei. spat.	$mIoU_I$	$mIoU_C$	Airplane	Bag	Cap	Car	Chair	Earphone	Guitar	Knife	Lamp	Laptop	Motorbike	Mug	Pistol	Rocket	Skate	Table
Superpoints	1	256	512	90	10	63.9	60.3	50.6	74.1	68.3	29.2	71.3	65.5	71.4	78.4	48.5	90.9	26.3	61.0	51.2	46.4	58.9	72.6
	2	512	256			64.2	60.3	50.8	70.5	68.2	29.9	71.1	65.6	74.1	80.1	48.7	91.3	26.4	54.9	53.6	48.6	58.6	72.9
	3	512	512			63.9	60.1	50.2	71.5	71.0	29.2	71.0	65.6	72.0	79.2	48.6	91.0	26.4	55.1	53.7	46.7	57.9	73.0
	4	256	128			64.2	60.4	50.6	69.7	66.7	30.1	71.6	65.5	72.5	80.4	48.6	91.1	25.8	63.3	54.9	46.4	56.1	72.6
	5	128	256			64.1	60.3	50.7	71.4	68.1	29.6	71.3	65.3	72.1	79.7	48.6	91.0	26.2	59.3	53.9	46.7	58.6	72.7
	6	128	128			64.0	59.9	50.2	67.5	65.8	29.6	71.7	65.2	71.8	80.4	48.9	90.8	26.8	60.1	53.5	44.7	58.7	72.5
Neighbours	7	256	256	90	10	64.1	60.4	50.7	64.7	68.2	30.0	71.4	65.2	72.2	79.2	48.4	91.1	26.0	67.0	54.0	47.8	58.3	72.7
	8			90	90	61.8	58.8	49.3	69.7	69.2	27.4	67.9	63.9	74.4	79.7	48.5	89.9	25.0	52.6	52.2	47.9	53.7	69.4
	9			256	170	59.0	55.3	47.6	59.8	67.3	25.1	64.8	60.9	69.2	80.3	48.2	89.2	24.0	36.5	51.4	46.7	48.8	65.7
	10			10	90	61.6	58.1	49.3	66.4	66.9	27.1	68.3	64.1	72.1	79.7	48.3	90.5	26.3	46.5	53.1	48.7	54.2	68.9

(b) Semantically-consistent aggregation followed by spatially-consistent aggregation

Table 2. Ablation on the geometric feature extractor (GFA) module on ShapeNetPart [11]. Results were obtained using 10 rendered (depth only) views. All experiments use a two-stage GFA. Panels (a) and (b) show how performance changes when swapping the two stages. Each of the first two groups of columns reports the configuration for the two stages. "Sup." is the number of super points sampled from the input point cloud. "Nei." is the number of neighbours considered during the aggregation phase. The highlighted row corresponds to the results shown in the main paper.

Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1, 3

[7] Evin Pınar Örnek, Yann Labbé, Bugra Tekin, Lingni Ma, Cem Keskin, Christian Forster, and Tomas Hodan. Foundpose: Unseen object pose estimation with foundation features. *arXiv preprint arXiv:2311.18809*, 2023. 3

[8] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 1

[9] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *ICCV*, 2019. 1, 3, 4

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 3

[11] 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*, 35(6):1–12, 2016. 1, 3, 4, 5

[12] Xiangyang Zhu, Renrui Zhang, Bowei He, Ziyu Guo, Ziyao Zeng, Zipeng Qin, Shanghang Zhang, and Peng Gao. Pointclip v2: Prompting clip and gpt for powerful 3d open-world learning. In *ICCV*, 2023. 1, 2

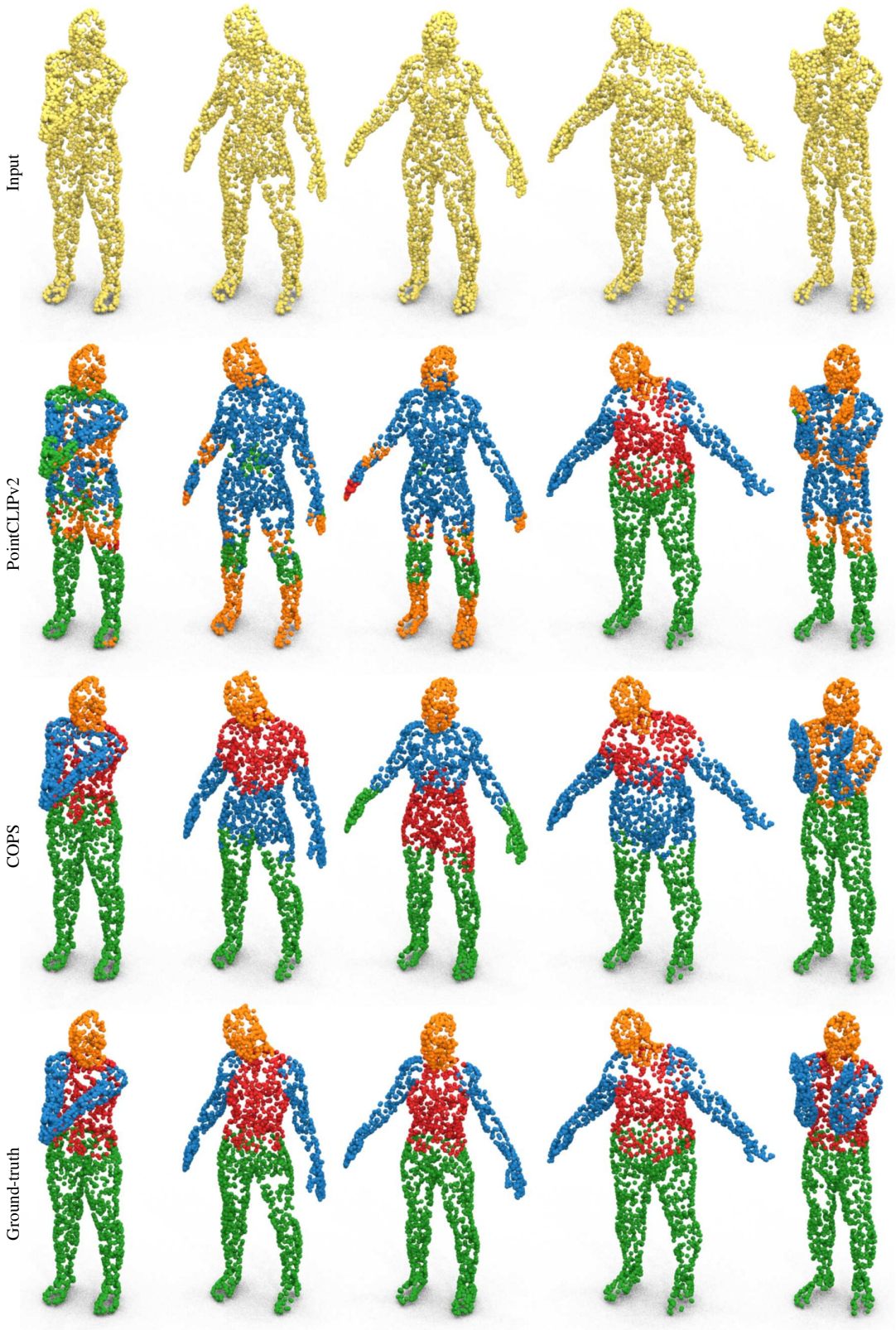


Figure 4. Qualitative results on FAUST [2]. Top to bottom: input texture-less point cloud (coloured in yellow for visualisation purposes), PointCLIPv2 predictions, COPS predictions, and ground-truth segmentation provided by SATR [1].