COARSE3D: Class-Prototypes for Contrastive Learning in Weakly-Supervised 3D Point Cloud Segmentation – Supplementary Material

Rong Li[1]
selirong@mail.scut.edu.cn
Anh-Quan Cao[2]
anh-quan.cao@inria.fr
Raoul de Charette[2]
raoul.de-charette@inria.fr

[1] South China University of Technology
Guangzhou, China
[2] Inria
Paris, France

In this document we study the effect of annotation sampling in Sec. 1 and report implementation details with new ablation in Sec. 2 and more qualitative results in Sec. 3.

1 Effect of annotations

In weak supervision settings, changing the set of labelled points impact performance. We now study more in depth the effect of sampled labels by randomly resampling labels 3 times on 2 datasets. As it appears in Tab. 1, our method is relatively stable (*i.e.*, std<1). We also measure the gap between random annotations and those of human by asking 2 operators to annotate the entire SemanticPOSS, labeling roughly 0.01% points per frame. Again, from Tab. 1 our 'human' labels are within 3 std of the mean 0.01% performance (*i.e.*, 29.27 vs 31.48±0.43).

	SemanticKITTI [8]		**SemanticPOSS [8]**		
Anno.	0.10%	0.01%	0.10%	0.01%	human (≈ 0.01%)
run 1	57.57	47.35	43.00	31.10	29.27
run 2	56.54	47.28	42.88	31.95	-
run 3	55.71	46.76	42.47	31.38	-
all	56.61±0.93	47.13±0.32	42.78±0.28	31.48±0.43	-

Table 1: Effect of annotation sampling on SemanticKITTI and SemanticPOSS.

2 Implementation details

2.1 Label voxel propagation

We replicate SQN [4] and apply their random grid downsampling with 0.06 voxel size. Our trivial scheme simply propagates existing labels to all points within the same voxel – thus densifying the labels at no extra labelling cost. In the extremely rare case of conflicting labels within a single voxel (e.g. a voxel having two labelled points with different classes), the voxel label will be randomly assigned.

We evaluate the effect of this voxel propagation scheme using the SalsaNext backbone on SemanticKITTI val. set in the 0.1% annotation setting. With/without our scheme we get **57.57**/56.26 mIoU. Since baselines do not use our label propagation, it is important to note that the mIoU gap obtained (+1.31 mIoU) is smaller than the gap with the original SalsaNext (+5.14, cf. main paper Tab. 3f) in the same 0.1% setting. This advocates that our method only *partly* benefits from our voxel propagation scheme and performs best thanks to our overall contrastive learning strategy.

2.2 Segmentation backbones

SalsaNext [3]. We use the official implementation[1] for the SemanticKITTI dataset[1] and applied our best effort to fairly re-implement it for nuScenes [2] and SemanticPOSS [6]. Specific to SemanticPOSS[6], we used an input padding to get a compatible size. When trained with our method, we finetune our contrastive learning hyperparameters.

SqueezeSegV3 [7]. We use the lighter *SqueezeSegV3-21* from the official implementation[2]. To boost performance on weakly supervised tasks, we replaced the multi-layer cross-entropy loss – improper for sparse weak labels due to its downsampling –, with normal cross-entropy loss. When trained with our method, we simply use the contrastive learning hyperparameter found with SalsaNext.

RangeNet++ [5]. We use the ligher *RangeNet-21* from the official implementation[3]. Inputs are pad to get compatible size. When trained with our method, we simply use the contrastive learning hyperparameter found with SalsaNext.

3 Additional results

We report additional qualitative results using SalsaNext on SemanticKITTI, SemanticPOSS, and nuScenes in Figs. 1, 2 and 3, respectively, for both 0.1% and 0.01% settings. Overall, our method surpasses SalsaNext, especially in ambiguous and cluttered regions, illustrated in Fig. 1 (sidewalk/parking - row 1; vegetation/building - row 3), in Fig. 2 (car - row 1; rider - row 3; pole/plants - row 4), and in Fig. 3 (terrain/other flat - row 1). Also, SalsaNext makes more mistake regarding classes with close semantical meaning as in Fig. 1 (truck/car - row 1), in Fig. 2 (building/fence - row 5), and in Fig. 3 (bus/trailer/truck - row 2, 3, 4, 5). Furthermore, our method shows superiority in predicting small objects with similar structure or spatial position, demonstrated in Fig. 1 (fence/vegetation - row 4, 5; trunk/traffic sign - row

[1] https://github.com/TiagoCortinhal/SalsaNext
[2] https://github.com/chenfengxu714/SqueezeSegV3
[3] https://github.com/PRBonn/lidar-bonnetal

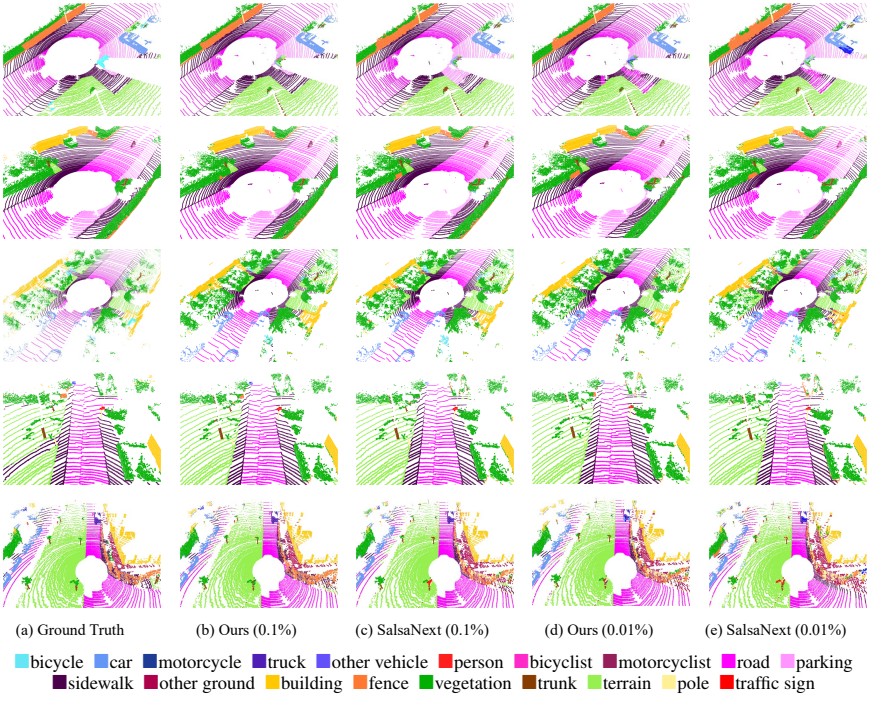

Figure 1: Additional qualitative results on SemanticKITTI [1]

5), in Fig. 2 (rider/people - row 2, 3, 5; pole/plants - row 4), and in Fig. 3 (terrain/other flat - row 1). Additionally, our method infers better far away, low density regions e.g. Fig. 2 (car - row 1), and Fig. 3 (vegetation/barrier - row 2).

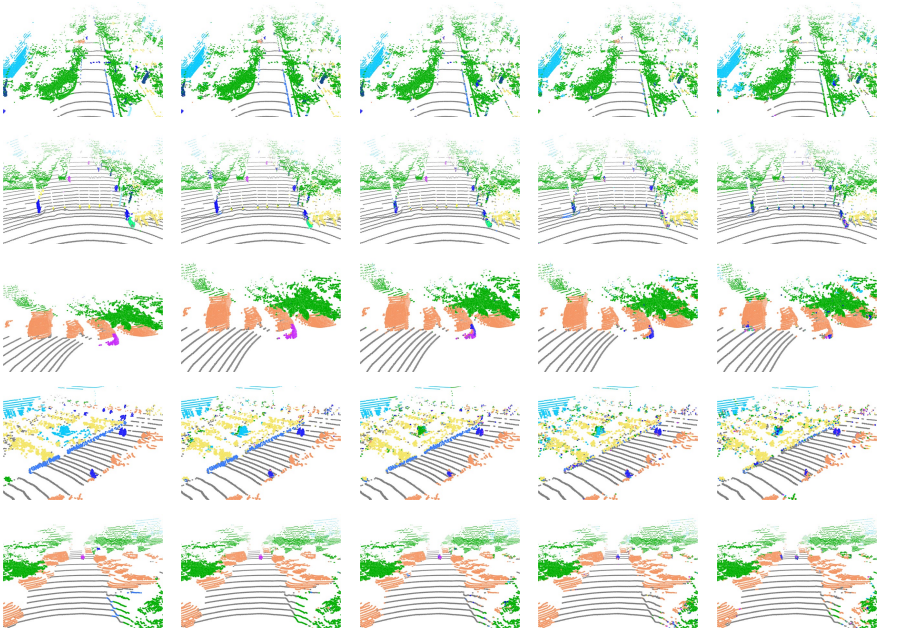

Figure 2: Qualitative results on SemanticPOSS [6]

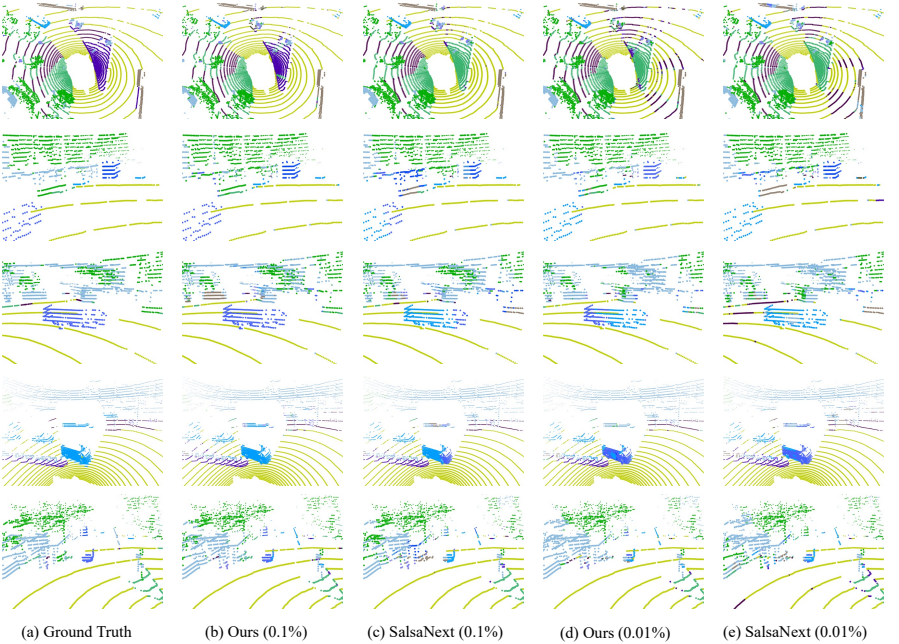

Figure 3: Qualitative results on nuScenes [2]

References

[1] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, C. Stachniss, and Juergen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, 2019.

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[3] Tiago Cortinhal, George Tzelepis, and Eren Erdal Aksoy. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *ISVC*, 2020.

[4] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Ales Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds with 1000x fewer labels. In *ECCV*, 2022.

[5] Andres Milioto, Ignacio Vizzo, Jens Behley, and C. Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *IROS*, 2019.

[6] Yancheng Pan, Biao Gao, Jilin Mei, Sibo Geng, Chengkun Li, and Huijing Zhao. Semanticposs: A point cloud dataset with large quantity of dynamic instances. In *IV*, 2020.

[7] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Péter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *ECCV*, 2020.