ALSO: Automotive Lidar Self-supervision by Occupancy estimation – Supplementary material –

Alexandre Boulch[1] Corentin Sautier[1,2] Björn Michele[1,3] Gilles Puy[1] Renaud Marlet[1,2]

[1]Valeo.ai, Paris, France [2]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, Marne-la-Vallée, France
[3]CNRS, IRISA, Univ. Bretagne Sud, Vannes, France

Appendix

Table of Contents

A. Experimental settings

In this section, we give additional information for reproducibility purpose. Additionally, the code is publicly available at github.com/valeoai/ALSO under an open source license.

A.1. Self-supervision

A.1.1 Occupancy decoder

The occupancy decoder, presented in Figure A1, is a four-layer MLP. It takes as input the concatenation of the latent vector and the local coordinates of the query point q with respect to the support point s. We use ReLU activations, and the hidden size of the MLP is set to 128, which is the size of the latent space.

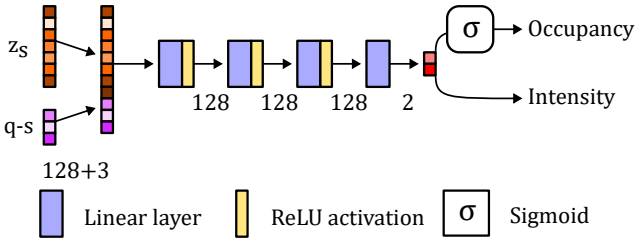


Figure A1. Decoder architecture.

We also provide a code sample for the decoder in Listing 1. The code is written in Python, using PyTorch [6] and PyTorch Geometric [3] for neighborhood computation.

A.1.2 Data transformation.

We detail here the transformations of the data used at pre-training.

Semantic segmentation pre-training. The voxel-size used in the sparse convolution backbone is set to 0.1 m for nuScenes/LivoxSimu and 0.05 m for SemanticKITTI/SemanticPOSS.

Detection pre-training. The voxel sizes are those used in OpenPCDet [9] and ONCE [4] for the different backbones. On KITTI, the voxel size is 0.05 m on the horizontal plane and 0.1 m in the vertical direction. On ONCE, the voxel size is 0.1 m on the horizontal plane and 0.2 m in the vertical direction.

Data augmentations. We apply classical point cloud data augmentations: random rotation around the z-axis as well as random flipping of the other axes.

Hardware configuration. For all our pre-traininings, we use a single Nvidia V100 16GB GPU.

```python
import torch
from torch.nn as Linear, ReLU
from torch.nn.functional import
    binary_cross_entropy_with_logits as bce_loss,
     l1_loss
from functools import partial
from torch_geometric.nn import radius

# l_size: latent_size
# o_size: output size
# r: neighborhood radius

class OccupancyDecoder(torch.nn.Module):

  def __init__(self, l_size=128, o_size=2, r=1):
    super().__init__()

    mlp = []
    for i in range(3):
        mlp.append(Linear(l_size, l_size))
    mlp.append(Linear(l_size, o_size))
    self.mlp = torch.nn.Sequential(*mlp)
    self.ball_search = partial(radius, r=r)
    self.r = r

  def forward(self, data):

    # get data from input dictionary
    pos_support = data["pos_support"]
    batch_support = data["batch_support"]
    pos_query = data["pos_query"]
    batch_query = data["batch_query"]
    latent = data["latent"]

    ## NEIGHBORHOOD SEARCH - LOCAL COORDINATES
    row, col = self.ball_search(x=pos_support,
        y=pos_query, batch_x=batch_support,
        batch_y=batch_query)
    pos_local = pos_query[row] - pos_support[col]
    l_local = latent[col]

    ## OCCUPANCY ESTIMATION
    x = torch.cat([l_local, pos_local], dim=1)
    x = self.mlp(x)
    occ_preds, i_preds = x[:, 0], x[:, 1]

    ## LOSS COMPUTATION
    occ_gt = data["query_occupancy"][row]
    occ_loss = bce_loss(occ_preds, occ_gt)

    i_gt = data["query_intensity"][row]
    i_mask = (i_gt >= 0)
    i_loss = l1_loss(i_preds[i_mask],
                     i_gt[i_mask])

    return occ_loss + i_loss
```

Listing 1. Pseudo-code of the decoder code using PyTorch syntax.

A.2. Details for finetuning on downstream tasks

Weight initialization. For semantic segmentation, we initialize all the backbone's weights with the pre-trained weights, except for the last layer (used for point-wise classification) which is randomly initialized. Then we finetune all the layers with the same learning rate, as described in the main paper.

For detection, we initialize the network's weights using the pre-trained weights, both for the sparse backbone and the dense BEV layers. These backbone and dense layers are finetuned along with the SECOND/PV-RCNN detectors on the task of supervised 3D object detection.

Hardware configuration. Semantic segmentation and KITTI detection downstream experiments are done on a single Nvidia RTX2080Ti 11GB GPU. For ONCE detection, we used 8 Nvidia V100 16GB GPUs.

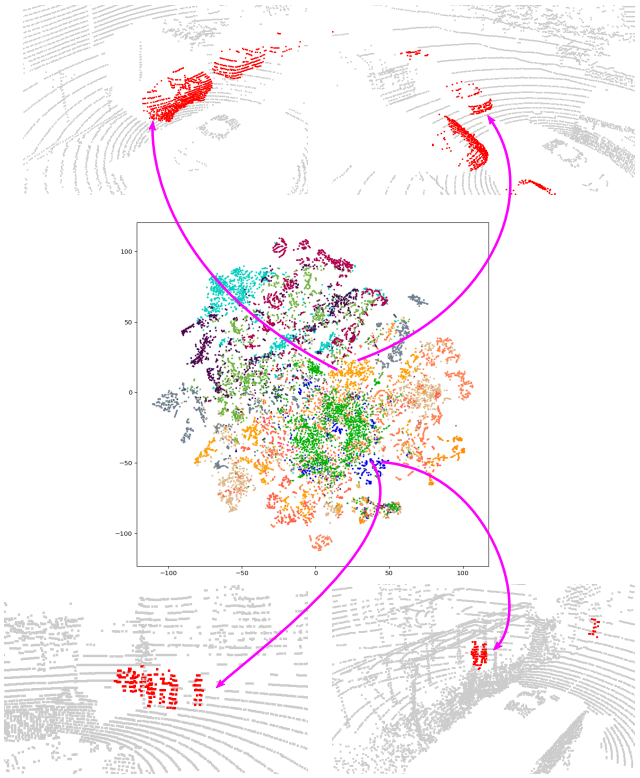


Figure A2. Latent space visualization.

B. Analysis of the latent space

In the main paper, we showed that using a self-supervised geometric pretext task is powerful to pre-train a backbone for both semantic segmentation and object detection. In this section, we look at the structure of the underlying latent space, learned using ALSO.

B.1. Natural clusters in latent space

We randomly select 15 nuScenes point clouds from which we select at most 2000 points of each semantic class. We gather the self-supervised latent vectors corresponding to

these points and embed them in a 2-dimensional space using t-SNE [10].

Figure A2 presents the result of this t-SNE embedding, where the colors encode the semantic classes. We notice that points belonging to the same class tends to be clustered together in this representation. When restricting the analysis to the car class and selecting neighbors in this representation, we notice that the corresponding cars are captured from the same point-of-view (rear right from the ego vehicle), indicating that the latent space probably tends to group together objects sharing the same apparent geometry.

B.2. Expected geometric properties visible in the latent space.

Surface orientation. In Figure A3 (a), we differentiate points belonging to horizontal flat surfaces (driveable surface, sidewalk, terrain and other flat surfaces) as opposed to points on objects usually sampled from the side, i.e., where surface is vertical (buildings, pedestrian...). We remark we can almost linearly separate these two sets of points in the t-SNE representation of the latent space, showing that the network rely on low-level geometric features, e.g., rough normal estimation, to solve the pretext task.

(a) Flat horizontal surfaces (yellow) vs vertical objects (purple)

(b) Side w.r.t. ego orientation: right (yellow) and left (purple)

Figure A3. Geometric structure of the latent space (nuScenes).

Symmetric occupancy map. In Figure A3 (b), we differentiate points with positive and negative x-coordinate, i.e., point on the right and on the left of the ego-vehicle. Again, we notice that these two sets of points are quite well separated in the t-SNE representation. This is explained by the fact the occupancy reconstruction task produces oriented surfaces. Two similar objects but located on different side of the road exhibit symmetric occupancy maps w.r.t. the (y, z) plane (see Figure A4), yielding different representations in the latent space.

C. Additional parameter and alternative studies

δ parameter. In our approach, a location at a random distance in $[0, \delta]$ behind an observed point is deemed occupied.

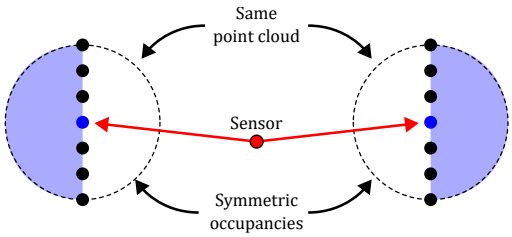


Figure A4. Symmetry of occupancy.

(a) δ parameter study

δ (m)	0.05	0.1	0.2	0.4	0.8
mIoU	38.3	38.4	38.7	38.7	38.1

(b) Decoder head alternatives

Decoder head	POCO (Knn + Att.)	Ball + Avg.	Ball + Max.	ALSO Ball per point
mIoU%	33.8	35.7	35.8	38.4

Table A1. Parameter study for δ (a) and alternative study for the decoder head (b) during the pre-training. Experiments are evaluated on the ablation-val set of nuScenes.

We argue this heuristics, also successfully used in [8], is simple and correct often enough although, as any heuristics, it can occasionally be wrong, just as random negatives in contrastive learning are also sometimes wrong. Importantly, our heuristics is stable across a significant range of values for δ, as shown in Table A1 (a). In our experiments, we uniformly chose $\delta = 0.1$ m as a kind of minimal thickness of the sort of objects we want to perceive. But, as visible in this table, there may be slightly more beneficial values depending on the dataset, e.g., 0.2 m for nuScenes.

Decoder head. Local information is good for accurate surface reconstruction [1, 7]: it allows each point feature to focus on local geometric details, as the decoder aggregates contextual information to predict occupancy. Instead, in ALSO, we force each individual point to know how to reconstruct its entire neighborhood on its own. Doing so yields single point features that are more context-aware, hence with a more semantic flavor, at the possible cost of a less-accurate surface reconstruction. In Table A1 (b), we provide an additional study to compare different reconstruction heads: POCO head [1], ball search + average/maximum pooling. It shows that limiting the expressiveness of the decoder to a per-point MLP (ALSO head) helps self-supervision.

D. Additional visualizations

We also provide additional visualizations on Figure A5. They are produced similarly to Figure 1 in the main paper. To produce these aggregated views, we compute the occupancy

in a 1-meter radius ball from the input points by randomly picking query points in this ball. Each inside point is then labeled with the estimated class of the closest input point. These estimated classes are obtained with the downstream model finetuned for semantic segmentation. Visualization are provided for SemanticKITTI, nuScenes and LivoxSimu.

E. Semantic segmentation: experiment scores details

All the scores for the ablation study and the semantic segmentation experiments are averaged over 5 runs in the main paper. We provide here all the individual score: in Table A2 for the ablation study, in Table A3 for the experiments on nuScenes, in Table A4 for the experiments on SemanticKITTI, in Table A5 for the experiments on SemanticPOSS and in Table A6 for the experiments on LivoxSimu. In each table, we highlight in bold the best run, and report the average score (same as in the paper) as well as the the standard deviation.

F. 3D detection: experiment scores details

Last, we provide more detailed scores for the experiments on 3D object detection.

On KITTI-3D detection, we provide in Table A7 scores for the different official metrics (2D object detection, bird's-eye-view detection, 3D object detection, orientation similarity). We also report the scores for the easy and hard categories (in the main paper, the reported scores correspond to the moderate difficulty category). Cells are colored according to the gain obtained using self-supervised weight initialization.

ALSO offers a performance boost on pedestrian and cyclists no matter what the metric is. However, on the car class, the gain is reduced, probably because the car class is already performing well.

As in [4], we report in Table A8 the per-distance performance for each of the three classes of interest. Our approach performs on par with the proposed baselines, including DeepCluster [2]. The performance boost is mainly due to the good detection performance of pedestrians between 0 and 50 meters.

References

[1] Alexandre Boulch and Renaud Marlet. POCO: Point convolution for surface reconstruction. In *CVPR*, pages 6302–6314, 2022. 3

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018. 4

[3] Matthias Fey and Jan E. Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. 1

[4] Jiageng Mao, Minzhe Niu, Chenhan Jiang, hanxue liang, Jingheng Chen, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu, Hang Xu, and Chunjing Xu. One million scenes for autonomous driving: ONCE dataset. In *NeurIPS Datasets and Benchmarks Track (Round 1)*, 2021. 1, 4

[5] Lucas Nunes, Rodrigo Marcuzzi, Xieyuanli Chen, Jens Behley, and Cyrill Stachniss. Segcontrast: 3D point cloud feature representation learning through self-supervised segment discrimination. *IEEE RA-L*, 7(2):2116–2123, 2022. 7, 8

[6] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 1

[7] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 3

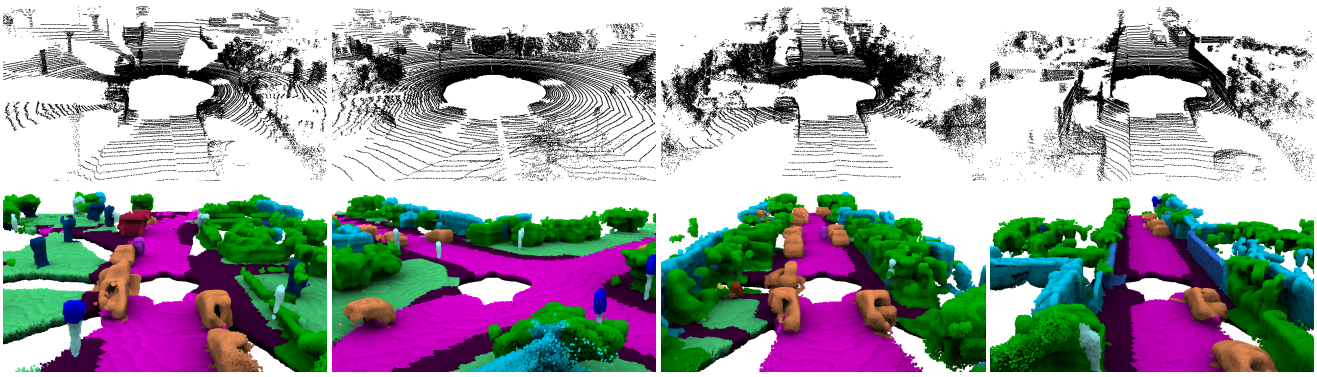
[8] Raphael Sulzer, Loic Landrieu, Alexandre Boulch, Renaud Marlet, and Bruno Vallet. Deep surface reconstruction from point clouds with visibility information. In *ICPR*, 2022. 3

[9] OpenPCDet Development Team. OpenPCDet: An open-source toolbox for 3D object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 1

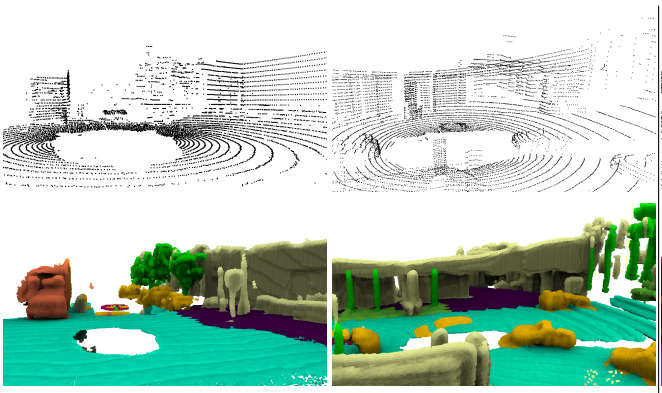
[10] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 3

[11] Saining Xie, Jiatao Gu, Demi Guo, Charles R. Qi, Leonidas J. Guibas, and Or Litany. PointContrast: Unsupervised pre-training for 3D point cloud understanding. In *ECCV*, 2020. 6, 7, 8

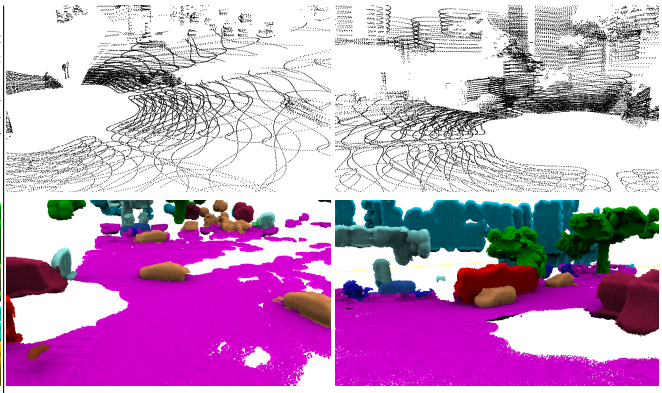
[12] Zaiwei Zhang, Rohit Girdhar, Armand Joulin, and Ishan Misra. Self-supervised pretraining of 3D features on any point-cloud. In *ICCV*, 2021. 6, 7, 8



(a) SemanticKITTI



(b) nuScenes



(c) LivoxSimu

Figure A5. Semantic segmentation predictions and occupancies on various datasets: (a) SemanticKITTI, (b) nuScenes, (c) LivoxSimu.

Input Intensity	Loss $\mathcal{L}_I$	Search radius (m)	Num. epochs	Runs					Average and std dev
✗	✗	1.0	100	36.45	**36.68**	35.94	36.38	36.64	36.42 ±0.30
✓	✗	1.0	100	**38.38**	38.17	38.12	38.23	38.14	38.21 ±0.10
✓	✓	1.0	100	38.48	38.54	38.31	**38.85**	38.01	**38.44 ±0.31**
✓	✓	0.5	100	37.60	**37.77**	37.74	37.25	37.60	37.59 ±0.21
✓	✓	1.0	100	38.48	38.54	38.31	**38.85**	38.01	**38.44 ±0.31**
✓	✓	2.0	100	38.06	37.93	**38.68**	38.45	37.68	38.16 ±0.40
✓	✓	4.0	100	36.18	36.89	36.40	36.38	35.88	36.35 ±0.37

Table A2. Ablation study on nuScenes custom ablation validation set, 1%. Details of experiments.

%	Backbone	Method	Runs					Average and std
0.1%	MinkUNet	No pre-training	21.88	21.21	**22.05**	21.08	21.96	21.64 ±0.45
		PointContrast [11]	26.39	27.35	**27.82**	26.95	26.89	27.08 ±0.54
		DepthContrast [12]	**21.89**	21.88	21.63	21.87	21.27	21.71 ±0.27
		ALSO	26.62	**26.86**	25.99	25.59	26.08	26.23 ±0.51
	SPVCNN	No pre-training	21.97	22.30	22.09	22.18	**22.45**	22.20 ±0.19
		ALSO	24.40	24.16	**25.86**	25.73	23.93	24.82 ±0.91
1%	MinkUNet	No pre-training	34.86	35.09	34.72	34.72	**35.55**	34.99 ±0.35
		PointContrast [11]	37.24	37.24	36.25	36.76	**37.36**	36.97 ±0.46
		DepthContrast [12]	34.51	34.74	**35.38**	34.23	34.07	34.59 ±0.51
		ALSO	37.42	37.52	37.15	37.11	**37.94**	37.43 ±0.34
	SPVCNN	No pre-training	34.27	**34.94**	34.26	34.10	34.37	34.39 ±0.32
		ALSO	37.24	37.14	37.55	37.24	**37.74**	37.38 ±0.25
10%	MinkUNet	No pre-training	57.62	**57.66**	57.31	56.70	57.19	57.30 ±0.39
		PointContrast [11]	59.00	58.73	58.66	58.96	**59.05**	58.88 ±0.17
		DepthContrast [12]	**58.03**	57.00	57.36	57.56	56.90	57.37 ±0.46
		ALSO	58.63	58.62	59.11	59.28	**59.35**	59.00 ±0.35
	SPVCNN	No pre-training	**57.37**	56.97	57.34	56.75	57.18	57.12 ±0.26
		ALSO	58.15	58.56	58.42	58.48	**58.60**	58.44 ±0.18
50%	MinkUNet	No pre-training	68.80	68.90	68.94	**69.31**	69.01	68.99 ±0.19
		PointContrast [11]	69.15	69.09	69.39	69.42	**69.75**	69.36 ±0.26
		DepthContrast [12]	69.12	69.04	69.38	**69.57**	68.66	69.15 ±0.35
		ALSO	69.69	69.58	69.93	69.66	**70.17**	69.81 ±0.24
	SPVCNN	No pre-training	**69.24**	69.06	68.68	68.74	69.09	68.96 ±0.24
		ALSO	69.55	**69.77**	69.47	69.24	69.65	69.54 ±0.20
100 %	MinkUNet	No pre-training	71.21	**71.35**	71.20	70.93	71.32	71.20 ±0.17
		PointContrast [11]	71.12	71.27	70.90	70.94	**71.31**	71.11 ±0.19
		DepthContrast [12]	71.31	71.20	71.30	70.81	**71.36**	71.20 ±0.22
		ALSO	**71.95**	71.92	71.60	71.88	71.39	71.75 ±0.24
	SPVCNN	No pre-training	70.82	70.79	70.56	**70.86**	70.41	70.69 ±0.19
		ALSO	71.41	71.18	70.99	71.20	**71.48**	71.25 ±0.20

Table A3. NuScenes. Details of experiments.

%	Backbone	Method	from [5]	Runs					Average and std
0.1%	MinkUNet	No pre-training	25.59	**30.22**	29.99	29.74	30.15	29.77	29.97 ±0.22
		PointContrast [11]	28.52	32.79	31.84	31.88	**32.96**	32.60	32.41 ±0.52
		DepthContrast [12]	33.51	32.43	32.09	**33.01**	32.24	32.80	32.51 ±0.38
		SegContrast [5]	34.78	**32.65**	32.38	32.48	32.18	31.83	32.30 ±0.31
		ALSO	N/A	**34.97**	34.83	34.81	35.10	35.11	34.96 ±0.14
	SPVCNN	No pre-training	N/A	**30.94**	30.81	30.66	30.47	30.81	30.74 ±0.18
		ALSO	N/A	35.35	34.78	34.71	34.93	**35.43**	35.04 ±0.33
1%	MinkUNet	No pre-training	41.70	45.1	46.32	46.59	**46.68**	46.49	46.24 ±0.65
		PointContrast [11]	43.40	47.71	47.97	48.22	47.25	**48.43**	47.92 ±0.46
		DepthContrast [12]	46.41	**49.62**	49.12	48.59	48.94	48.74	49.00 ±0.40
		SegContrast [5]	47.41	48.91	**49.35**	48.87	48.81	48.54	48.90 ±0.29
		ALSO	N/A	50.04	50.28	49.43	**50.41**	50.02	50.04 ±0.38
	SPVCNN	No pre-training	N/A	46.24	**47.23**	46.53	46.26	46.67	46.59 ±0.40
		ALSO	N/A	49.34	**49.75**	49.05	48.90	48.32	49.07 ±0.53
10%	MinkUNet	No pre-training	53.87	57.04	**58.74**	57.71	56.27	58.01	57.55 ±0.94
		PointContrast [11]	53.79	59.48	59.78	**60.44**	59.26	59.56	59.70 ±0.45
		DepthContrast [12]	56.29	59.49	60.74	60.27	60.46	**60.75**	60.34 ±0.52
		SegContrast [5]	55.21	**59.63**	58.57	58.45	58.78	58.29	58.74 ±0.53
		ALSO	N/A	60.41	60.45	60.47	**60.54**	60.43	60.46 ±0.05
	SPVCNN	No pre-training	N/A	58.8	58.95	59.21	**59.47**	57.85	58.86 ±0.62
		ALSO	N/A	60.71	60.32	**60.97**	60.32	60.66	60.60 ±0.28
50%	MinkUNet	No pre-training	58.34	61.48	**62.33**	61.88	61.80	61.31	61.76 ±0.39
		PointContrast [11]	57.30	62.68	62.91	62.54	62.35	**63.19**	62.73 ±0.33
		DepthContrast [12]	58.54	63.24	**63.31**	63.16	62.44	62.37	62.90 ±0.46
		SegContrast [5]	58.33	**62.58**	62.20	61.61	61.74	62.46	62.12 ±0.43
		ALSO	N/A	63.09	63.43	62.99	63.28	**64.15**	63.39 ±0.46
	SPVCNN	No pre-training	N/A	61.32	62.15	61.61	61.7	**62.39**	61.83 ±0.43
		ALSO	N/A	63.4	63.4	63.45	**64.08**	63.44	63.55 ±0.29
100%	MinkUNet	No pre-training	59.63	62.49	62.35	62.98	62.50	**63.06**	62.68 ±0.32
		PointContrast [11]	59.77	63.57	63.14	63.13	**63.95**	63.26	63.41 ±0.35
		DepthContrast [12]	59.88	63.76	**64.31**	63.52	63.54	64.12	63.85 ±0.35
		SegContrast [5]	60.53	**62.64**	61.57	62.53	62.24	62.45	62.29 ±0.43
		ALSO	N/A	**64.29**	63.75	63.75	63.34	63.07	63.64 ±0.46
	SPVCNN	No pre-training	N/A	62.39	62.86	62.33	**62.88**	62.82	62.66 ±0.27
		ALSO	N/A	63.60	**64.04**	63.59	63.93	63.76	63.78 ±0.20

Table A4. SemanticKITTI. Details of experiments.

%	Backbone	Method	Runs					Average and std
0.1%	MinkUNet	No pre-training	37.23	**37.53**	36.37	36.72	36.59	36.89 ±0.48
		PointContrast [11]	39.22	**40.60**	38.56	39.24	38.73	39.27 ±0.80
		DepthContrast [12]	38.69	39.35	**41.16**	39.87	39.25	39.66 ±0.94
		SegContrast [5]	41.72	**42.89**	41.45	41.74	40.68	41.70 ±0.79
		ALSO	40.04	41.23	**41.29**	40.88	39.83	40.65 ±0.68
1%	MinkUNet	No pre-training	**46.99**	46.23	46.09	46.33	46.47	46.42 ±0.35
		PointContrast [11]	**48.45**	48.26	48.40	48.43	47.11	48.13 ±0.58
		DepthContrast [12]	48.08	48.78	48.29	48.36	**48.96**	48.49 ±0.36
		SegContrast [5]	48.94	**50.02**	49.34	49.08	49.66	49.41 ±0.44
		ALSO	**50.55**	48.85	49.02	49.86	49.51	49.56 ±0.68
10%	MinkUNet	No pre-training	54.29	53.96	54.53	**54.95**	54.60	54.47 ±0.37
		PointContrast [11]	**55.91**	54.60	54.77	54.96	55.29	55.11 ±0.52
		DepthContrast [12]	**56.17**	55.81	55.17	55.82	55.96	55.79 ±0.37
		SegContrast [5]	55.48	55.41	55.39	**55.50**	54.97	55.35 ±0.22
		ALSO	56.16	**56.19**	55.43	55.15	56.14	55.81 ±0.49
50%	MinkUNet	No pre-training	**55.48**	55.19	55.40	55.27	55.04	55.28 ±0.17
		PointContrast [11]	55.84	56.46	55.62	55.90	**57.01**	56.17 ±0.56
		DepthContrast [12]	54.67	56.35	56.12	56.26	**56.73**	56.03 ±0.79
		SegContrast [5]	56.84	55.89	54.82	**56.89**	56.76	56.24 ±0.89
		ALSO	**57.07**	55.56	56.71	56.13	56.30	56.35 ±0.58
100%	MinkUNet	No pre-training	54.52	**55.52**	**55.52**	55.10	54.83	55.10 ±0.44
		PointContrast [11]	55.80	56.02	55.48	**57.13**	56.40	56.17 ±0.63
		DepthContrast [12]	**57.38**	56.36	56.40	56.29	56.00	56.49 ±0.52
		SegContrast [5]	56.24	56.46	56.22	**57.17**	55.83	56.38 ±0.49
		ALSO	56.88	**58.23**	55.45	56.01	56.88	56.69 ±1.05

Table A5. SemanticPOSS. Details of experiments.

%	Backbone	Method	Runs					Average and std
0.1%	MinkUNet	No pre-training	47.43	**48.38**	48.03	48.21	48.10	48.03 ±0.36
		ALSO	51.47	52.31	**54.35**	52.28	52.45	52.57 ±1.07
1%	MinkUNet	No pre-training	63.73	63.33	63.46	**64.38**	64.03	63.79 ±0.43
		ALSO	**65.86**	65.48	65.16	65.50	65.24	65.45 ±0.27
10%	MinkUNet	No pre-training	66.51	**66.84**	66.67	66.60	66.64	66.65 ±0.12
		ALSO	67.30	68.08	67.43	**68.10**	67.82	67.75 ±0.37
50%	MinkUNet	No pre-training	68.35	**68.87**	68.10	68.60	68.52	68.49 ±0.29
		ALSO	69.55	69.41	69.66	69.55	**69.73**	69.58 ±0.12
100%	MinkUNet	No pre-training	68.91	68.87	68.82	**69.25**	68.68	68.91 ±0.21
		ALSO	69.37	69.86	69.62	**70.14**	69.61	69.72 ±0.29

Table A6. Livox Synthetic Dataset. Details of experiments.

Backbone	Metric set	Method	Pre-training	Car			Pedestrian			Cyclist		
			Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
SECOND	2D object detection	Scratch†	-	95.84	94.49	92.00	68.27	64.69	61.15	91.02	78.88	76.00
		ALSO	KITTI	96.88	94.43	91.89	70.44	67.60	64.40	91.67	81.67	78.01
			KITTI-360	95.69	94.34	91.81	69.76	67.12	64.12	93.26	80.97	77.87
			nuScenes	97.48	94.70	93.56	72.39	68.69	65.86	91.34	81.64	78.46
	Bird's eye view	Scratch†	-	93.76	89.82	87.65	59.74	54.85	50.56	87.19	70.96	68.00
		ALSO	KITTI	93.82	90.16	88.00	60.38	56.01	52.58	87.90	73.39	69.15
			KITTI-360	92.38	89.60	87.79	61.75	57.36	53.91	89.26	73.74	69.48
			nuScenes	94.64	90.77	88.24	64.32	59.13	55.25	86.92	74.58	70.17
	3D object detection	Scratch†	-	90.20	81.50	78.61	53.89	48.82	44.56	82.59	65.72	62.99
		ALSO	KITTI	90.76	81.97	79.10	56.30	51.93	48.03	83.71	69.14	65.27
			KITTI-360	88.95	81.79	78.92	57.83	52.45	48.32	86.76	70.68	66.56
			nuScenes	90.21	81.78	78.97	59.56	54.24	50.27	81.12	68.19	64.10
	Orientation similarity	Scratch†	-	95.83	94.35	91.79	63.70	59.52	55.85	90.86	78.44	75.52
		ALSO	KITTI	96.86	94.30	91.65	66.44	62.82	59.17	91.36	81.07	77.35
			KITTI-360	95.68	94.24	91.63	65.58	62.25	58.52	92.35	78.70	75.64
			nuScenes	97.45	94.54	93.30	67.67	63.33	60.28	91.01	80.75	77.49
PV-RCNN	2D object detection	Scratch†	-	97.86	94.39	93.92	73.84	68.68	65.53	94.34	81.89	77.36
		ALSO	KITTI	98.26	94.42	94.07	76.05	70.89	67.50	95.32	83.41	80.42
			KITTI-360	98.04	94.42	94.11	76.75	71.15	67.40	95.18	83.59	78.70
			nuScenes	96.12	94.45	93.99	73.70	68.70	65.31	94.61	81.86	78.67
	Bird's eye view	Scratch†	-	94.65	90.61	88.56	68.28	60.62	55.95	92.52	75.03	70.40
		ALSO	KITTI	94.82	90.75	88.67	68.93	61.88	57.74	93.18	77.73	73.09
			KITTI-360	94.40	90.60	88.56	72.04	63.40	59.05	95.11	77.25	73.37
			nuScenes	93.10	90.64	88.53	68.72	60.92	56.96	93.11	76.74	73.06
	3D object detection	Scratch†	-	91.74	84.60	82.29	65.51	57.49	52.71	91.37	71.51	66.98
		ALSO	KITTI	91.90	84.72	82.55	65.57	58.49	53.75	92.52	75.06	70.48
			KITTI-360	92.13	84.68	82.58	68.72	60.16	54.87	92.86	74.04	69.30
			nuScenes	92.31	84.86	82.61	65.60	57.76	52.96	91.70	74.98	70.67
	Orientation similarity	Scratch†	-	97.84	94.25	93.70	69.73	63.89	60.31	94.20	81.00	76.47
		ALSO	KITTI	98.23	94.31	93.89	70.07	64.94	61.09	95.15	82.84	79.79
			KITTI-360	98.02	94.32	93.91	72.55	66.62	62.64	94.85	83.07	78.14
			nuScenes	96.09	94.29	93.76	67.66	62.74	59.37	94.23	81.07	77.86

†: retrained by ourselves, scores may vary from main paper.

Color scale: [-3,-2[[-2,-1[[-1,0[[0,1[[1,2[[2,3[[3,4[[5,6[[6,7[

Table A7. KITTI3D detection. Details of experiments. Cells are colored according to difference with from-scratch pre-training.

Method	Vehicle				Pedestrian				Cyclist				mAP
	overall	0-30	30-50	50-inf	overall	0-30	30-50	50-inf	overall	0-30	30-50	50-inf	
U_{small}													
baseline	71.19	84.04	63.02	47.25	26.44	29.33	24.05	18.05	58.04	69.96	52.43	34.61	51.89
BYOL	68.02	81.01	60.21	44.17	19.50	22.16	16.68	12.06	50.61	62.46	44.29	28.18	46.04 -5.85
PointContrast	71.07	83.31	64.90	49.34	22.52	23.73	21.81	16.06	56.36	68.11	50.35	34.06	49.98 -1.91
SwAV	72.71	83.68	65.91	50.10	25.13	27.77	22.77	16.36	58.05	69.99	52.23	34.86	51.96 +0.07
DeepCluster	73.19	84.25	66.86	50.47	24.00	26.36	21.73	16.79	58.99	70.80	53.66	36.17	52.06 +0.17
ALSO	71.73	84.30	65.21	48.30	28.16	31.45	25.19	16.29	58.13	70.04	52.76	33.88	52.68 +0.79

Table A8. ONCE detection. Detail of experiments.