S3PT: Scene Semantics and Structure Guided Clustering to Boost Self-Supervised Pre-Training for Autonomous Driving - Supplementary material

Maciej K. Wozniak[1,5*] Hariprasath Govindarajan[2*] Marvin Klingner[1]
Camille Maurice[1] B Ravi Kiran[3] Senthil Yogamani[4]
*co-first authors
{mwozniak, hargov, mklingne, cmaurice, ravkira, syogaman}@qti.qualcomm.com
[1]Automated Driving, Qualcomm Technologies International GmbH
[2]Arriver Software AB and Linköping University, Sweden
[3]Qualcomm SARL, France [4]Automated Driving, Qualcomm Technologies, Inc
[5]KTH Royal Institute of Technology, Sweden

1. Configuration details

1.1. Pre-training setup

Here, we provide the complete pre-training configuration for all the methods that we evaluate in Sec. 4.2 of the main paper. The DINO and CRiBo methods are pre-trained using their publicly available repositories and we follow their default configuration. For completeness, we provide the full configuration details for DINO in Tab. 1 and for CRiBo in Tab. 2. Our proposed S3PT is based on the CRiBo repository and we use a similar configuration to the defaults, except for our proposed modifications. The full configuration details of S3PT is provided in Tab. 3.

1.2. Evaluation protocols

In this section, we provide details about the evaluation protocols and configurations used for different downstream evaluation experiments, namely semantic segmentation, domain generalization and 3D object detection.

Hyper-parameter	ViT-Small/16	ViT-Base/16
training epochs	500	100
batch size	256	256
learning rate	5e−4	7.5e−4
warmup epochs	10	10
freeze last layer epochs	1	3
min. learning rate	1e−5	2e−6
weight decay	0.04 → 0.4	0.04 → 0.4
stochastic depth	0.1	0.1
gradient clip	-	0.3
optimizer	adamw	adamw
fp16	✓	✓
momentum	0.996 → 1.0	0.996 → 1.0
global crops	2	2
global crops scale	[0.25, 1.0]	[0.25, 1.0]
local crops	10	10
local crops scale	[0.05, 0.25]	[0.05, 0.25]
head mlp layers	3	3
head hidden dim.	2048	2048
head bottleneck dim.	256	256
norm last layer	✗	✓
num. prototypes	65536	65536
teacher temp.	0.04 → 0.07	0.04 → 0.07
temp. warmup epochs	30	50
student temp.	0.1	0.1

Table 1. Hyperparameter settings for DINO

1.2.1 Semantic segmentation

We obtain dense feature representations from frozen backbones and only train the head network for semantic segmentation on nuImages [1] and Cityscapes [4] datasets. We consider a similar evaluation protocol as in CRiBo [5] and use the linear decoder head from Segmenter [8]. This linear probing head maps the token features (16 × 16 patches) to class assignments and then uses a bilinear upsampling to transform the outputs to image pixel dimensions. We additionally, also evaluate the Mask Transformer decoder head, with the same default configuration proposed in Segmenter. We use the `mmsegmentation` 1.2.2 [3] library for evaluation and use the default 160K iterations training schedule. For each pre-training method and dataset, the results reported in the tables are the best results after considering the following set of learning rates: {`8e-4, 3e-4, 8e-5`}, similar to other works which perform such evaluations. For nuImages, we use the same dataset configuration setup as the publicly available `ade20k` dataset configuration in `mmsegmentation`. For Cityscapes, we use the `cityscapes_768x768` configuration. The configurations for the linear probing and Mask Transformer decoder heads in Segmenter are available in `mmsegmentation`. For domain generalization experiments, we use the same semantic segmentation models (frozen backbone and decoder

Hyper-parameter	ViT-Small/16	ViT-Base/16
training epochs	500	100
batch size	256	256
learning rate	5e−4	7.5e−4
warmup epochs	10	10
freeze last layer epochs	1	3
min. learning rate	1e−5	2e−6
weight decay	0.04 → 0.4	0.04 → 0.4
stochastic depth	0.1	0.1
gradient clip	-	0.3
optimizer	adamw	adamw
fp16	✓	✓
momentum	0.996 → 1.0	0.996 → 1.0
global crops	2	2
global crops scale	[0.25, 1.0]	[0.32, 1.0]
head mlp layers	3	3
head hidden dim.	2048	2048
head bottleneck dim.	256	256
norm last layer	✗	✓
num. prototypes	65536	65536
teacher temp.	0.04 → 0.07	0.04 → 0.07
temp. warmup epochs	30	50
student temp.	0.1	0.1
sinkhorn lambda	20.0	20.0
sinkhorn iterations	5	5
pos alpha	[1.0, 1.0]	[1.0, 1.0]
which features	last	last
num spatial clusters	32	32
queue size	25000	25000

Table 2. Hyperparameter settings for CRiBo

Hyper-parameter	ViT-Small/16	ViT-Base/16
training epochs	500	100
batch size	256	256
learning rate	5e−4	7.5e−4
warmup epochs	10	10
freeze last layer epochs	1	3
min. learning rate	1e−5	2e−6
weight decay	0.04 → 0.4	0.04 → 0.4
stochastic depth	0.1	0.1
gradient clip	-	0.3
optimizer	adamw	adamw
fp16	✓	✓
momentum	0.996 → 1.0	0.996 → 1.0
global crops	2	2
global crops scale	[0.25, 1.0]	[0.32, 1.0]
head mlp layers	3	3
head hidden dim.	2048	2048
head bottleneck dim.	256	256
norm last layer	✗	✗
num. prototypes	65536	65536
vmf normalization	✓	✓
centering	probability	probability
teacher temp.	0.04 → 0.07	0.04 → 0.07
temp. warmup epochs	30	50
student temp.	0.1	0.1
sinkhorn lambda	20.0	20.0
sinkhorn iterations	1	1
pos alpha	[1.0, 1.0]	[1.0, 1.0]
depth alpha	[4.0, 4.0]	[4.0, 4.0]
which features	last	last
num spatial clusters	128	128
queue size	2500	2500

Table 3. Hyperparameter settings for S3PT

head) trained using the above described setup.

1.2.2 3D object detection

For our test and training for 3D object detection task we use `mmdetection3D` 1.4.0 [2]. For benchmarking, we use camera-only 3D object detector PETR [7] (author's implementation in mmdetection3D library[1]). In short, PETR encodes the position information of 3D coordinates into image features, creating 3D position-aware features. This allows object queries to perceive these features and perform end-to-end object detection. Essentially, PETR transforms multi-view images into a unified 3D space by combining positional information directly with the image features. This approach enables the model to detect objects in 3D space using only camera data, without relying on LiDAR or other sensors.

The original backbone for PETR in `mmdetection3D` implementation is VoVNetCP, based on the VoVNet [6], is designed for efficient and effective feature extraction. It uses a unique One-Shot Aggregation (OSA) module, which concatenates features from multiple layers only once, reducing computational overhead and improving efficiency.

For original benchmark, denoted as *supervised* in Tab. 4 (of the main paper), we used the backbone weights provided by the authors[2]. This model is pre-trained on DDAD15M and then, trained on nuScenes train set in a supervised manner.

For self-supervised pre-training of image backbones with methods: DINO, CRiBo and S3PT (ours) we used weights obtained in pre-training, described in Sec. 4.2 of the main paper (detailed above in Sec. 1.1). Next, we trained PETR together with these backbones, however, with frozen image backbone weights, thus gradients does not flow through to the image backbone. Please, refer to Tab. 4 for detailed training setting and we use the `cyclic-20e` scheduler from `mmdetection3D`.

2. Additional results

2.1. ViT backbones with different patch sizes

In addition to the standard ViTs with patch size 16 evaluated in the main paper, we also evaluate the impact of using different patch sizes in ViT-Small model. Specifically, we evaluate patch sizes 14, 16 and 32. We avoid patch size 8, as it is extremely compute expensive to train. We pre-train these models using S3PT for 100 epochs with the same pre-training configuration. Only difference is that we reduce the number of spatial clusters to 32 for ViT-Small/32 due to the

[1] https://github.com/open-mmlab/mmdetection3d/tree/main/projects/PETR

[2] https://drive.google.com/file/d/1ABI5BoQCkCkP4B0pO5KBJ3Ni0tei0gZi/view

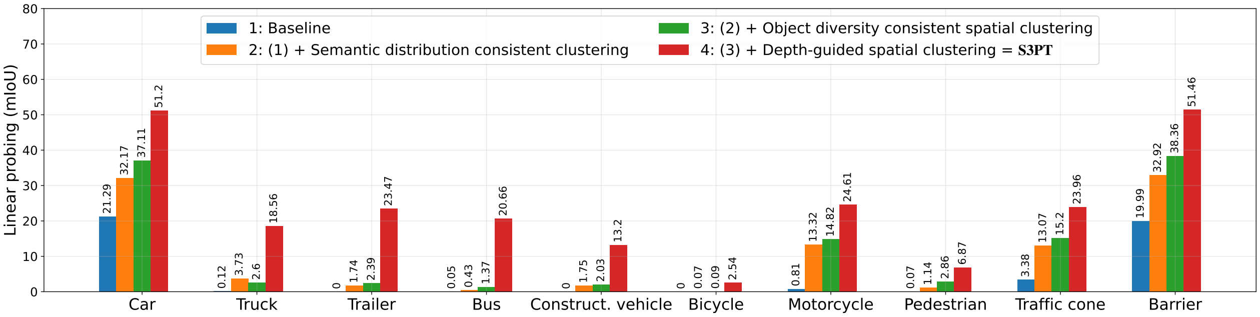

Figure 1. Object-wise segmentation performance of models from Tab. 1 using the linear probing head. The models are obtained by sequentially applying the proposed modifications to CrIBo baseline to finally achieve S3PT.

Hyper-parameter	PETR
img size	$(320, 800)$
grid size	$[512, 512, 1]$
voxel size	$[0.2, 0.2, 8]$
training epochs	20
batch size	2
learning rate	3e−4
warmup epochs	1
weight decay	0.01
stochastic depth	0.1
gradient clip	35
optimizer	Adamw
fp16	✗
momentum	$0.85 \rightarrow 1.0$
global crops	2
PETR head	
head hidden dim.	$384/784(ViT-S/B)$
num query	900
num layers	6
num heads	8
feedforward channels	2048

Table 4. Hyperparameter settings for 3D object detection

Class	mIoU		
	ViT-Small/14	ViT-Small/16	ViT-Small/32
Car	68.57	68.47	63.59
Truck	36.1	41.52	30.72
Trailer	33.43	30.13	24.83
Bus	42.59	34.9	32.12
Construction Vehicle	26.21	14.55	15.37
Bicycle	12.47	9.33	10.5
Motorcycle	48.18	47.09	42.65
Pedestrian	23.41	22.1	14.75
Traffic Cone	33.54	35.96	25.83
Barrier	65.82	66.82	61.74
Overall	**50.97**	**50.04**	**46.04**

Table 5. Semantic segmentation performance on nuImages (with Mask Transformer head) of ViT-Small models with different patch sizes, after pre-training with S3PT on nuScenes dataset

decreased number of patch tokens. We evaluate the models on nuImages semantic segmentation task with a Mask Transformer head and report the results in Tab. 5. We find smaller patch sizes to perform better, which is in agreement with other self-supervised pre-training methods. This is especially beneficial for dense prediction tasks, where smaller patch sizes enable more granular predictions.

2.2. Object-wise performance of our contributions

In Fig. 5 of the main paper, we showed the object-wise performance of sequentially adding our contributions on the nuImages semantic segmentation task with a Mask Transformer head. We show the linear probing performance of adding our contributions in Fig. 1 above.

2.3. Average distance to different objects in the dataset

In Fig. 2 can see that on average most of the objects are between 15-25m away from from the ego vehicle. However, some objects are far away, which makes them harder to detect. In Tab. 6 we show that adding more and more distant objects, substantially decrease the performance of PETR. Nevertheless, We observe that PETR with S3PT (our backbone) does not drop as much in performance as other models which features have not learnt any 3D cues. Additionally it performs better on long-tailed distribution objects.

2.4. Additional qualitative results

In Fig. 3 we present additioanl results for models used in hyper parameter search in Table 1 and Figure 5. We can observe how adding each component to our training schema influences final segmentation quality results. Note, these presented models are ViT-S pretrained with only 100 epochs with Linear Probing, we used this shorter trainig for hyperparameter search. Full results require 500 epochs for ViT-S and 100 for ViT-B.

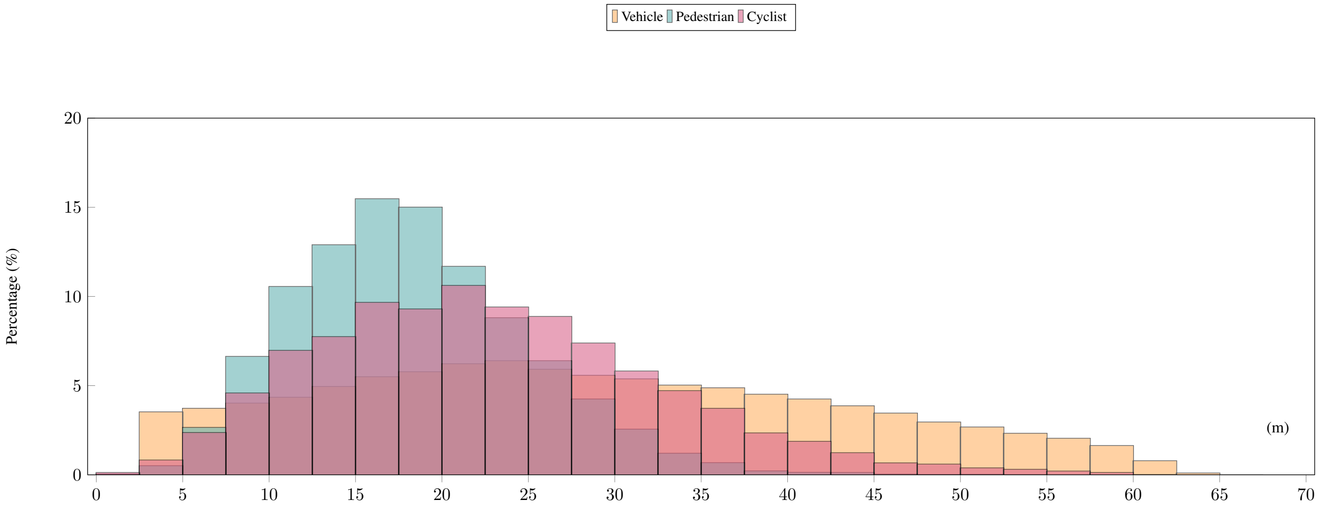

Figure 2. Average distance to different objects in the dataset cap at 65 meters

Table 6. Performance on 3D object detection with different backbones, depending on the distance/range. We can see that our method is more robust towards far away objects (note, we remove construction vehicle class).

Class Names	**0-20m**			**0-30m**			**0-60m**		
	Dino	**Cribo**	**Ours**	**Dino**	**Cribo**	**Ours**	**Dino**	**Cribo**	**Ours**
car	0.515	0.489	0.555	0.288	0.333	0.396	0.276	0.246	0.299
truck	0.300	0.344	0.368	0.159	0.229	0.281	0.166	0.151	0.195
bus	0.276	0.237	0.427	0.172	0.139	0.312	0.077	0.056	0.135
trailer	0.153	0.103	0.222	0.144	0.082	0.168	0.122	0.053	0.168
pedestrian	0.455	0.401	0.477	0.337	0.269	0.349	0.226	0.205	0.272
motorcycle	0.357	0.331	0.405	0.295	0.210	0.307	0.197	0.196	0.275
bicycle	0.033	0.02	0.014	0.016	0.027	0.028	0.006	0.005	0.022
traffic cone	0.352	0.335	0.370	0.285	0.287	0.297	0.263	0.263	0.285
barrier	0.592	0.452	0.421	0.418	0.465	0.430	0.369	0.343	0.368
mAP	**0.337**	**0.299**	**0.362**	**0.261**	**0.227**	**0.285**	**0.189**	**0.169**	**0.224**

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1

[2] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 2

[3] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 1

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1

[5] Tim Lebailly, Thomas Stegmüller, Behzad Bozorgtabar, Jean-Philippe Thiran, and Tinne Tuytelaars. Cribo: Self-supervised learning via cross-image object-level bootstrapping. In *Proceedings of the International Conference on Learning Representations*, 2024. 1

[6] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 2

[7] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *Proceedings of the European Conference on Computer Vision*, pages 531–548, 2022. 2

[8] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia

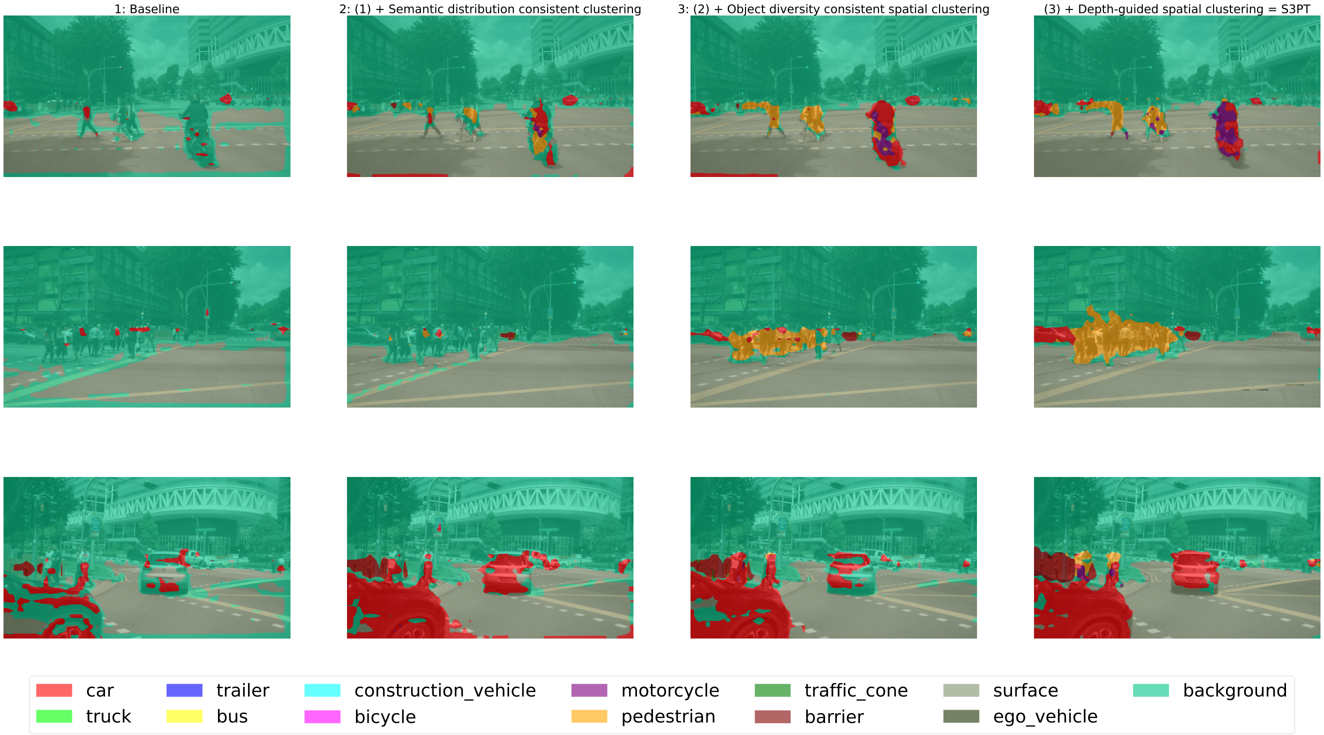
1: Baseline
2: (1) + Semantic distribution consistent clustering
3: (2) + Object diversity consistent spatial clustering
(3) + Depth-guided spatial clustering = S3PT
🟥 car
🟩 truck
🟦 trailer
🟨 bus
construction_vehicle
bicycle
motorcycle
pedestrian
traffic_cone
barrier
surface
ego_vehicle
background

Figure 3. Qualitative results obtained by adding each of our proposed components to the baseline CRiBo method. Note that these are visualizations corresponding to the ablation experiments shown in Tab. 1 and the object-wise performances shown in Fig. 5, where we pre-trained a ViT-S/16 backbone for only 100 epochs and then trained a linear probing Segmenter head. With a longer pre-training for 500 epochs using S3PT, we demonstrate improved further segmentation quality (see Fig. 6 and Tab. 2).

Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the International Conference on Computer Vision*, pages 7262–7272, 2021. 1