Supplementary Material: Cross-sensor Domain Adaptation for 3D Object Detection Using Stationary Object Aggregation Pseudo-labelling

A. Implementation Details

In this section, we include the implementation details for the experiments presented in the main text.

A.1. Point Cloud Input

Single- and few-frame input: As detailed in the main text, the nuScenes [1] and Waymo [4] datasets have different sensor ranges and point cloud features. In our experiments, we match the input point cloud format. Specifically, the input point cloud range is $[-75, 75]$ m for both x and y dimensions, and $[-2, 4]$ m for the z dimension. For few-frame models, we use five features (x, y, z, i, e, t) for each point, where (x, y, z) are the point location, i is the intensity normalized to $[0, 1]$, e is the elongation, and t is the timestamp offset in seconds. For the single-frame model, we exclude timestamp offset t.

Since the nuScenes dataset does not provide elongation information, we set $e = 0$ for all nuScenes point clouds. Moreover, following previous work [5, 7], we apply a $+1.8$ m offset to the z dimension to approximately transform the nuScenes point clouds from sensor frame to ego vehicle frame.

Full-sequence input: For full-sequence models, we only use the (x, y, z) channels. To reduce training time and memory consumption, we pre-compute the aggregated point cloud for each sequence and perform a voxel-downsampling step with $3.25\,\mathrm{cm}^3$ voxels. During training, the pre-computed aggregated point clouds are transformed using pose transformations and further uniformly downsampled to at most 1,000,000 points.

Similar to single and few-frame input, a $+1.8$ m offset is applied to nuScenes aggregated point clouds.

A.2. Architecture

We use CenterPoint [8] and VoxelNeXt [2] implemented in the open-source framework OpenPCDet[1] with minor modifications to make the models compatible to both nuScenes and Waymo datasets.

Voxelization: The point cloud is voxelized using a voxel size of $(7.5\,\mathrm{cm}, 7.5\,\mathrm{cm}, 15\,\mathrm{cm})$. For each point cloud, we use at most 500,000 voxels, with each voxel containing at most 10 points.

[1] https://github.com/open-mmlab/OpenPCDet

Backbone: We adopt the backbone used in nuScenes models for both datasets. Detailed configurations can be found in the OpenPCDet repository.

Detection heads: Since our models are trained for only Vehicle / Car class, We use a single detection head for both architectures. For few-frame models, an additional head with 2 convolution layers is added to regress the velocity (v_x, v_y).

A.3. Training

All models are trained with a total batch size of 32, over multiple GPUs. We use the Adam optimizer with a one cycle learning rate schedule.

Baseline: All single-frame and few-frame models are trained for 36 epochs. The learning rate is set to 0.001 for nuScenes models and 0.003 for Waymo models.

ST3D [7] For the nuScenes $\rightarrow$ Waymo direction, the models are fine-tuned for 12 epochs with a learning rate of 0.0001. The positive and negative confidence thresholds for pseudo-labelling are set to $(0.5, 0.3)$ for Direct pseudo-labels and $(0.1, 0.05)$ for SOAP pseudo-labels. For the Waymo $\rightarrow$ nuScenes direction, the models are fine-tuned for 6 epochs with a learning rate of 0.0003. The positive and negative confidence thresholds are set to $(0.6, 0.2)$ for Direct pseudo-labels and $(0.3, 0.2)$ for SOAP pseudo-labels. In all experiments, Direct pseudo-labels are updated every 2 epochs using the memory ensemble proposed in ST3D.

SSDA3D [5] Both stages in SSDA3D experiments follow the baseline training configurations. The CutMix and MixUp augmentation probabilities are set to 0.5. Predictions from the first stage models are filtered by a confidence threshold of 0.3 to construct the corresponding pseudo-labels for second stage training. When SOAP predictions are used, confidence thresholds of 0.15 and 0.25 are used to construct Waymo and nuScenes pseudo-labels, respectively.

SOAP The SOAP model is initialized with the weights from a corresponding few-frame model and trained for an additional 12 epochs with a learning rate of 0.001 for nuScenes and 0.003 for Waymo. As described in the main text, the annotations are constructed using QST. We set the QSS threshold ϵ to 0.7 and 0.85 for nuScenes and Waymo datasets, respectively.

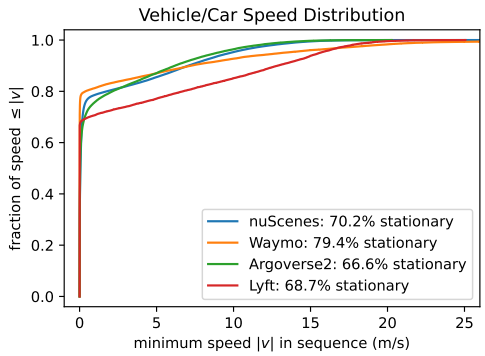


Figure 1. Cumulative distribution for Vehicle / Car speed in realistic self-driving datasets.

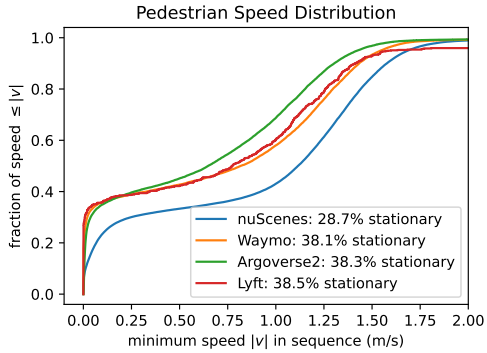


Figure 2. Cumulative distribution for Pedestrian speed in realistic self-driving datasets.

A.4. Post-processing

The post-processing for each dataset follows the implementation in OpenPCDet.

nuScenes: The predictions are filtered with a confidence threshold of 0.1 and a range of $[-61.2, 61.2]$ m for both x and y, and $[-10, 10]$ m for z. NMS is performed on the best 1000 predictions using an IoU threshold of 0.2, with at most 83 predictions retained.

Waymo: The predictions are filtered with a confidence threshold of 0.1 and a range of $[-75.2, 75.2]$ m for both x and y, and $[-2, 4]$ m for z. NMS is performed on the best 4096 predictions using an IoU threshold of 0.7, with at most 500 predictions retained.

SCP The SOAP preditions undergo the SCP step, which clusters and filters predictions in the global coordinate system. The cluster size threshold η depends on the frame rate of the dataset, so we use $\eta = 10$ for Waymo (10 Hz) and $\eta = 2$ for nuScenes (2 Hz). The cluster threshold μ for both SCP and WBF are set to 0.5.

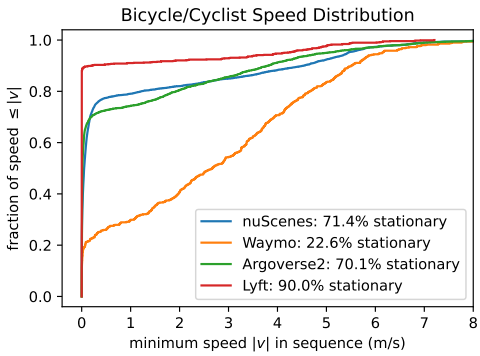


Figure 3. Cumulative distribution for Bicycle / Cyclist speed in realistic self-driving datasets.

B. Speed Statistics in Self-driving Datasets

As mentioned in the main text, we observe that stationary objects are a statistically important component of object detection. In Fig. 1, we present the cumulative distribution of speeds for the Vehicle / Car class in four realistic self-driving datasets: nuScenes [1], Waymo [4], Lyft [3], and Argoverse2 [6]. In all datasets, we observe a significant proportion of objects are stationary ($|v| < 0.2\,\mathrm{m/s}$) at some point in the sequence, ranging from 66.6% in Argoverse2 to 79.4% in Waymo.

The corresponding distribution for pedestrian and bicycle/cyclist are shown in Figs. 2 and 3. For bicycle/cyclist class, we observe similar statistics in nuScenes, Argoverse and Lyft dataset. Note that in the Waymo dataset, only bicycles with riders are labelled, hence the much lower percentage compared to other datasets.

For pedestrian, however, the percentage of objects that are stationary at some point in the sequence is significantly smaller. As mentioned in the limitation section in the main text, this may limit the effectiveness of our approach to these classes.

C. Additional Results

We include additional evaluation based on speed for nuScenes → Waymo CenterPoint models in Tables 1 and 2. First, we notice that in all cases, the stationary performance of SOAP pseudo-labels and models fine-tuned with SOAP pseudo-labels exceeds SOTA by a significant margin, highlighting the effectiveness of our proposed method. Second, interestingly, while the pseudo-label performance for dynamic objects is on par or worse than the few-frame baseline (Direct and Co-training), after fine-tuning with the SOAP pseudo-labels using ST3D or SSDA3D, the dynamic performance is consistently better than SOTA methods.

Method	Training Data	Stationary (<0.2 m/s) Level 1		Stationary (<0.2 m/s) Level 2		Slow (0.2-1 m/s) Level 1		Slow (0.2-1 m/s) Level 2		Medium (1-3 m/s) Level 1		Medium (1-3 m/s) Level 2		Fast (3-10 m/s) Level 1		Fast (3-10 m/s) Level 2	
Direct	$\{\mathcal{S}\}$	21.5		18.0		**24.3**		**22.3**		**27.5**		**25.5**		**27.1**		**25.4**	
SOAP (ours)		**56.2**	**+63.3%**	**49.5**	**+63.5%**	23.9	-0.9%	22.0	-0.7%	24.4	-7.4%	22.6	-7.4%	25.7	-2.8%	24.1	-2.7%
ST3D [7]	$\{\mathcal{S}, \mathcal{T}_P\}$	27.1	+10.2%	22.8	+9.7%	26.1	+3.9%	24.0	+4.0%	23.9	-8.6%	22.1	-8.6%	28.8	+3.4%	27.0	+3.4%
ST3D + SOAP (ours)		**48.6**	**+49.5%**	**41.7**	**+47.8%**	**32.6**	**+17.9%**	**30.0**	**+17.9%**	**32.2**	**+11.2%**	**29.9**	**+11.2%**	**34.4**	**+14.7%**	**32.3**	**+14.5%**
Oracle	$\{\mathcal{T}\}$	76.3		67.6		70.6		65.3		69.6		64.9		76.7		73.1	

$\mathcal{S}$: labelled source domain; $\mathcal{T}$: labelled target domain; $\mathcal{T}_\subset$: small subset of $\mathcal{T}$; $\mathcal{T}_P$: pseudo-labelled target domain

Table 1. Unsupervised domain adaptation results for nuScenes → Waymo, where Waymo dataset is unlabelled, split based on object speed. The percentages represent the amount of the Direct–Oracle domain gap closed.

Method	Training Data	Stationary (<0.2 m/s) Level 1		Stationary (<0.2 m/s) Level 2		Slow (0.2-1 m/s) Level 1		Slow (0.2-1 m/s) Level 2		Medium (1-3 m/s) Level 1		Medium (1-3 m/s) Level 2		Fast (3-10 m/s) Level 1		Fast (3-10 m/s) Level 2	
Direct	$\{\mathcal{S}\}$	21.5		18.0		24.3		22.3		27.5		25.5		27.1		25.4	
Co-training	$\{\mathcal{S}, \mathcal{T}_\subset\}$	58.1	+66.2%	50.0	+64.4%	50.1	+52.3%	46.2	+52.1%	44.5	+37.7%	41.3	+37.6%	51.5	+47.9%	48.7	+47.5%
CutMix [5]		57.4	+65.1%	49.6	+63.4%	**51.1**	**+54.3%**	**47.1**	**+54.1%**	**48.0**	**+45.4%**	**44.5**	**+45.2%**	**57.2**	**+59.2%**	**54.2**	**+58.7%**
SOAP (ours)		**71.9**	**+91.3%**	**64.6**	**+93.6%**	39.4	+30.6%	36.4	+30.8%	38.6	+24.6%	35.8	+24.5%	51.8	+48.5%	49.0	+48.2%
SSDA3D [5]	$\{\mathcal{S}, \mathcal{T}_\subset, \mathcal{T}_P\}$	66.5	+81.6%	58.2	+80.7%	56.3	+64.8%	52.0	+64.6%	53.8	+58.2%	50.0	+58.2%	62.6	+69.7%	59.4	+69.2%
SSDA3D + SOAP (ours)		**70.1**	**+88.0%**	**61.6**	**+87.5%**	**58.7**	**+69.7%**	**54.3**	**+69.6%**	**54.6**	**+60.0%**	**50.7**	**+59.9%**	**63.6**	**71.8%**	**60.3**	**+71.2%**
Oracle	$\{\mathcal{T}\}$	76.7		67.8		73.7		68.2		72.6		67.6		78.0		74.4	

$\mathcal{S}$: labelled source domain; $\mathcal{T}$: labelled target domain; $\mathcal{T}_\subset$: small subset of $\mathcal{T}$; $\mathcal{T}_P$: pseudo-labelled target domain

Table 2. Semi-supervised domain adaptation results for nuScenes → Waymo, where 1% of Waymo data is labelled, split based on object speed. The percentages represent the amount of the Direct–Oracle domain gap closed.

D. Qualitative Results

We present qualitative results of SOAP pseudo-labels for nuScenes → Waymo in Figs. 4 and 5. In both unsupervised and semi-supervised settings, we observe that SOAP pseudo-labels are more accurate compared to Direct, Co-training, and CutMix [5], especially for far objects.

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 *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020. 1, 2

[2] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. VoxelNeXt: Fully sparse voxelnet for 3D object detection and tracking. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pages 21674–21683, 2023. 1

[3] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Level 5 perception dataset 2020. https://level-5.global/level5/data/, 2019. 2

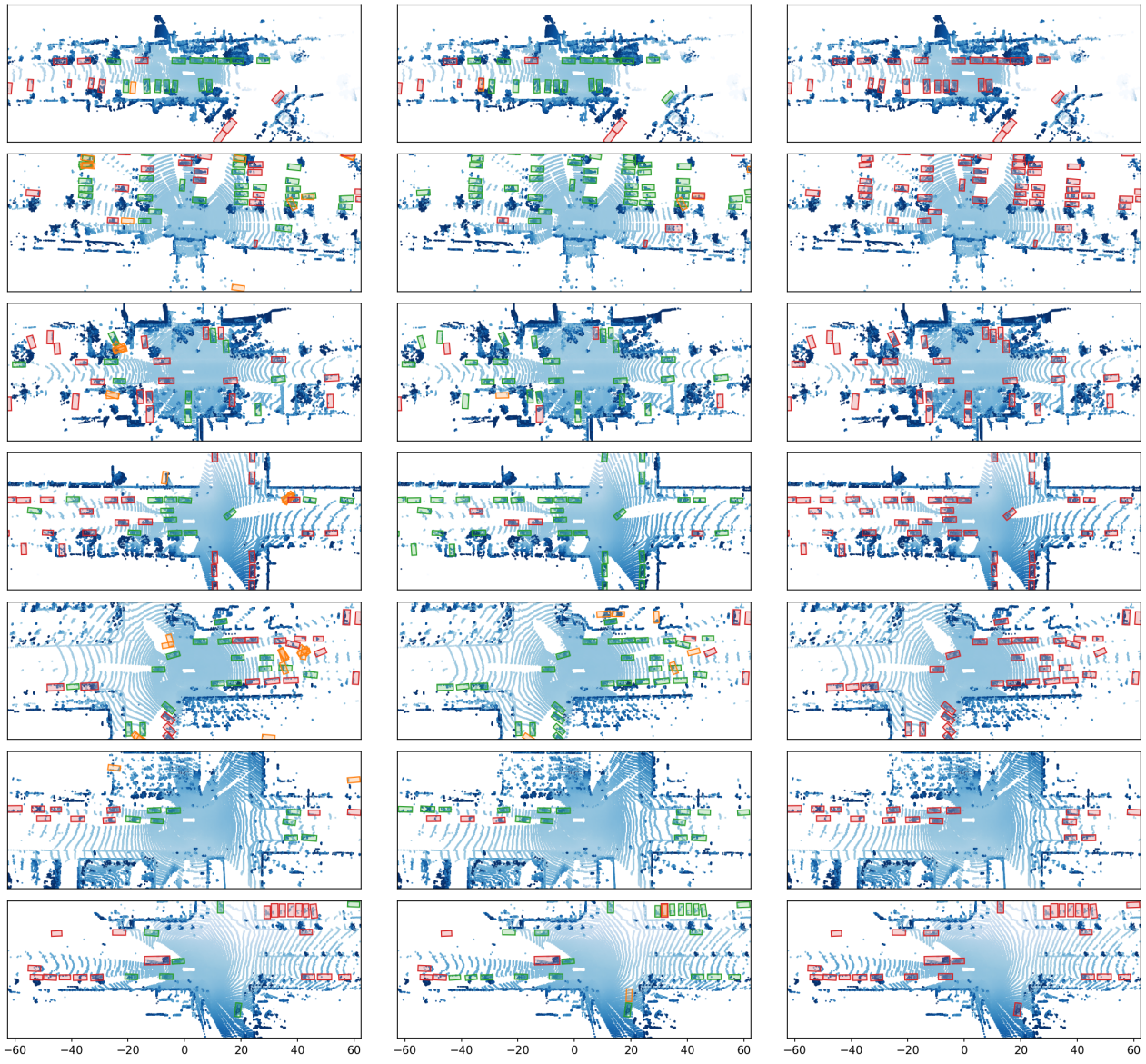
[4] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2020. 1, 2

[5] Yan Wang, Junbo Yin, Wei Li, Pascal Frossard, R. G. Yang, and Jianbing Shen. SSDA3D: Semi-supervised domain adaptation for 3D object detection from point cloud. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2023. 1, 3

[6] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2

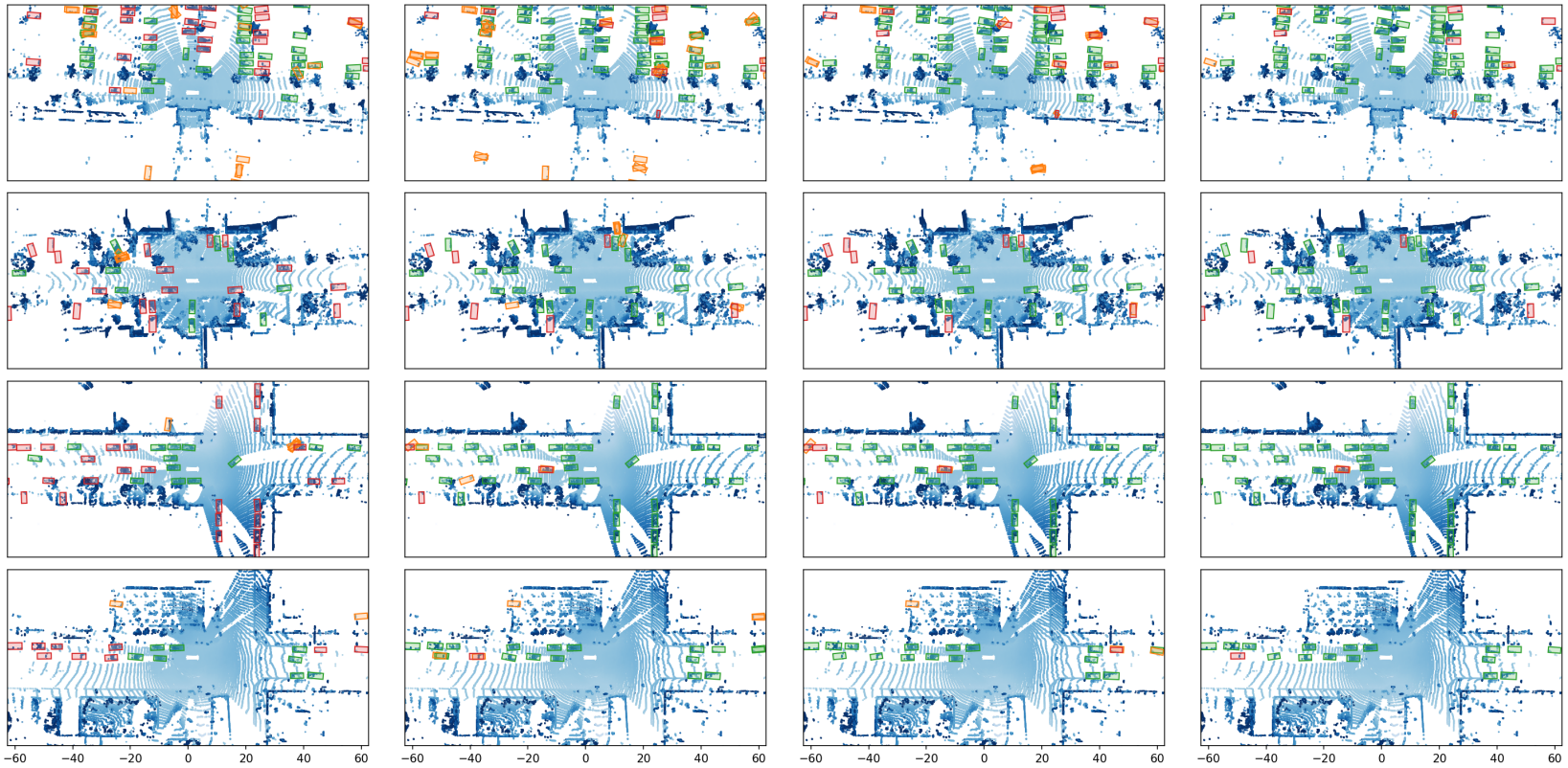
[7] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for unsupervised domain adaptation on 3D object detection. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2021. 1, 3

[8] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D object detection and tracking. In *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, pages 11784–11793, 2021. 1



(a) Direct (b) SOAP (c) Ground Truth

Figure 4. Examples of pseudo-labels generated by different methods in nuScenes → Waymo unsupervised domain adaptation setting, and the corresponding ground truth labels. Green represents true positive pseudo-labels, orange represents false positive pseudo-labels, and red represents false negative pseudo-labels.



(a) Direct (b) Co-training (c) CutMix (SSDA3D) (d) SOAP

Figure 5. Examples of pseudo-labels generated by different methods in nuScenes → Waymo semi-supervised domain adaptation setting. Green represents true positive pseudo-labels, orange represents false positive pseudo-labels, and red represents false negative pseudo-labels.