Supplementary Material for ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection

Jihan Yang[1]*, Shaoshuai Shi[2]*, Zhe Wang[3,4], Hongsheng Li[2,5], Xiaojuan Qi[1]†
[1]The University of Hong Kong [2]CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
[3]SenseTime Research [4]Shanghai AI Laboratory [5]School of CST, Xidian University
{jhyang, xjqi}@eee.hku.hk, shaoshuaics@gmail.com, wangzhe@sensetime.com, hsli@ee.cuhk.edu.hk

Outline

In this supplementary file, we provide more details and visualizations omitted in our main paper due to 8-pages limits on paper length:

- Sec. S1: Dataset details for our domain adaptation tasks.
- Sec. S2: Analysis of domain difference and systematic bias on pseudo labels .
- Sec. S3: Implementation details for SECOND-IoU and other memory ensemble variants.
- Sec. S4: More experimental results with IoU threshold at 0.5.
- Sec. S5: Additional ablation studies.
- Sec. S6: Qualitative results.
- Sec. S7: Experiments on other adaptation tasks.

S1. Dataset Overview

We compare four LiDAR 3D object detection datasets as shown in Table S1. They are different in LiDAR type, beam angles, point cloud density, size, and locations for data collection. Visual illustrations in Figure S1 obviously show the different patterns of LiDAR point clouds in terms of distribution and density. Even for data from LiDARs with same beams (Waymo, KITTI, and Lyft in Figure S1), point clouds are also different in the range, vertical, and horizontal distributions. For instance, Waymo not only utilizes a small horizontal azimuth of LiDAR, but also clusters LIDAR beams in the medium of vertical angles (see Figure S1). Both these LiDAR setups lead to denser point clouds in the collected data (see *# points per scene* in Table S1).

We conduct experiments on domain adaptations from the label-rich domain to label-insufficient domains (*i.e*. Waymo $\rightarrow$ KITTI, Waymo $\rightarrow$ Lyft, Waymo $\rightarrow$ nuScenes) and the more challenging domain adaptations across domains with the different number of LiDAR beams (*i.e*. Waymo $\rightarrow$ nuScenes and nuScenes $\rightarrow$ KITTI). On all evaluated settings, our approach improves the baseline method and outperforms the existing approach by a significant margin, demonstrating the efficacy of the proposed approach.

S2. Domain Difference and Systematic bias

S2.1. Lyft Annotation Discrepancies

The Lyft [6] dataset is constructed by a labeling protocol different from the other three datasets, *i.e*. the Lyft dataset does not annotate objects on both sides of the road. For instance, we observe that the objects on the main branch of the road (*w.r.t* the ego car) are most likely annotated, while many objects on both sides might not be annotated. Visual illustrations of the annotated bounding boxes are shown in Fig. S2 for the Waymo dataset (blue boxes) and Fig. S3 (blue boxes) for the Lyft dataset.

The differences in annotation protocols will have a negative influence on the evaluation of domain adaptation results. When we use the pre-trained model on the Waymo dataset to evaluate data from the Lyft scenes, our model correctly predicts the cars on two sides of the road (see green boxes in Fig S3), which, however, are not annotated by the Lyft dataset (see blue boxes in Fig. S3). This makes it hard to evaluate the actual performance boost with the proposed domain adaptation method. We believe that our method can obtain a further performance boost if the results are properly evaluated.

S2.2. Analysis of Domain Discrepancy

We conclude that the domain gap mainly lies in two folds: (i) content gap (*e.g*. object size) caused by different data-capture locations; (ii) point distribution gap caused by different LiDAR beams. Self-training explicitly closes the domain gap by reformulating the UDA problem as a target

*equal contribution
†corresponding author

Dataset	LiDAR Type	Beam Angles	# Points Per Scene†	# Training Frames	# Validation Frames	Location
Waymo [10]	64-beam	[-18.0°, 2.0°]*	160,139	158,081	39,987	USA
KITTI [3]	64-beam	[-23.6°, 3.2°]	118,624	3,712	3,769	Germany
Lyft [6]	64-beam	[-29.0°, 5.0°]*	69,175	18,900	3,780	USA
nuScenes [2]	32-beam	[-30.0°, 10.0°]	24,966	28,130	6,019	USA and Singapore

Table S1. Dataset overview. Notice that we use **version 1.0** of Waymo Open Dataset. * indicates we obtain the information from [11]. † means that we count this statistical information only on the validation set.

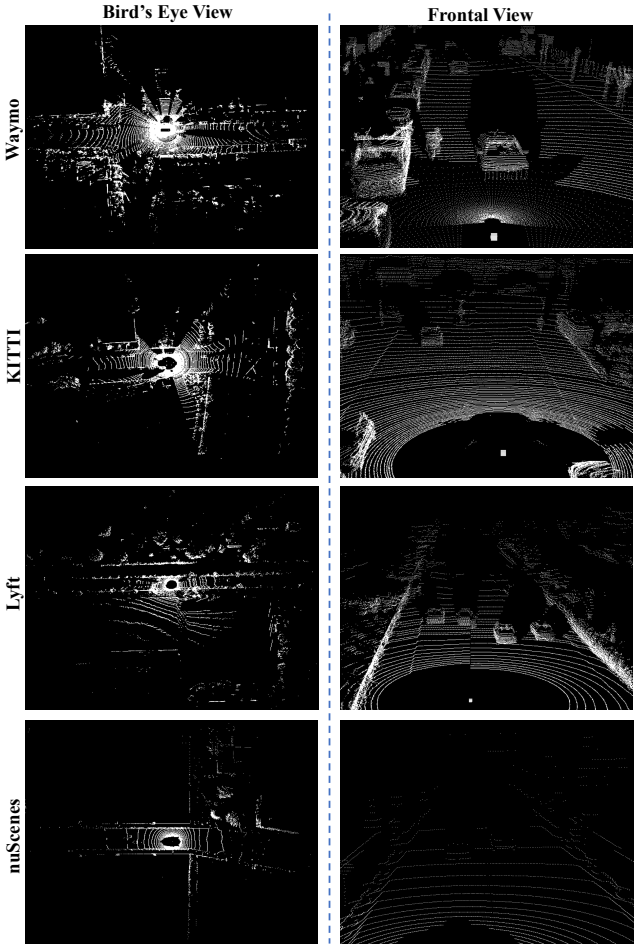


Figure S1. Visualization of bird's eye views (left) and frontal views (right) for different datasets: Waymo [10], KITTI [3], Lyft [6] and nuScenes [2]. nuScenes has obviously sparse point clouds than other three datasets since it is only collected by 32-beam LiDAR. Even Waymo, KITTI and Lyft all utilize 64-beam LiDARs, Waymo is denser than KITTI and Lyft and its beams are clustered in the medium of vertical angles.

domain supervised problem with pseudo labels, where better pseudo labels provide better performance.

S2.3. Systematic Bias on Pseudo Labels

An important systematic bias on pseudo labels is *Annotation style bias* due to different annotation rules such as how to annotate (tightness of bounding boxes) and which to annotate (See Sec. S2.1 in Suppl.). This will make pseudo labels biased toward the source domain labeling rules, different from target domain GT.

S3. Implementation details

In this section, we give more implementation details in constructing our adaptation tasks. Further, we illustrate the component selection of the oracle model, the IoU head of SECOND [12] as well as the other two memory ensemble variants: NMS ensemble and bipartite ensemble.

S3.1. Parameter setups

We typically pre-train the detector for 30 epochs on Waymo and then train 30 epochs for self-training to converge on Waymo $\rightarrow$ KITTI setting. Besides, we update pseudo labels every two epoch. The scaling range of ROS is [0.75, 1.1], ensuring a reasonable scaled car size. For the QTMB, the two thresholds T_{neg} and T_{pos} of triplet box partition are 0.25 and 0.6, respectively. As for CDA, we split the total self-training epochs into six stages (i.e., epochs [0, 5), [5, 10), [10, 15), [15, 20), [20, 25), [25, 30)). More detailed parameter setups could be found in our released code.

S3.2. Details of Voxel Size and GT Sampling for Oracle Model.

Here, we provide more details on the voxel size for SECOND-IOU and the GT sampling strategy for training.

Voxel Size. We derive our Oracle model with voxel size [0.10m, 0.10m, 0.15m] rather than [0.05m, 0.05m, 0.15m] To be noted, we adopt this setting in **all** experiments including our pre-trained model and self-training pipeline for a fair evaluation. The reason why we adopt this setting is that all our models are trained with the ring view (about $150m \times 150m$) which will take too much GPU memory if the voxel size is set to [0.05m, 0.05m, 0.15m] (we can only set batch size as 1 for SECOND-IoU and totally fail to run PV-RCNN with such voxel size). We use NVIDIA GTX 1080Ti with 11G GPU memory for all experiments and adopt voxel size [0.10m, 0.10m, 0.15m] to achieve the best trade-off between memory and realization in various settings as well as frameworks.

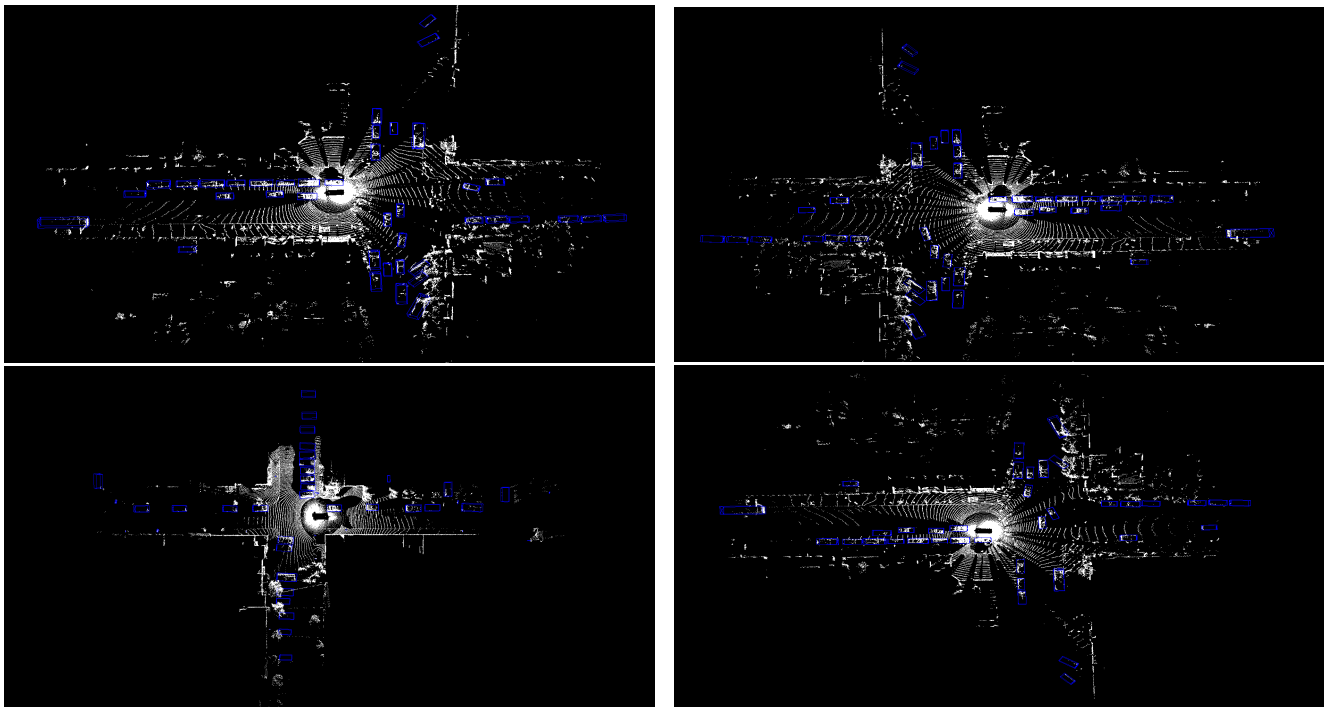


Figure S2. Examples of Waymo scenes. The blue boxes are ground-truth bounding boxes.

Method	Voxel Size	GT Sampling	AP_{BEV} / AP_{3D}
Oracle (Ours)	[0.10m, 0.10m, 0.15m]		83.29 / 73.45
	[0.05m, 0.05m, 0.15m]		85.99 / 76.53
	[0.10m, 0.10m, 0.15m]	√	88.08 / 81.52
	[0.05m, 0.05m, 0.15m]	√	**88.56 / 81.87**
Oracle (SN [11])	-	unknown	80.60 / 68.90

Table S2. Comparison of different setting (voxel size and GT sampling) for our Oracle model based on SECOND-IoU. We also compare them with the Oracle performance release in the SN [11] based on PointRCNN. The reported AP results are evaluated on the moderate difficulty of the car category of the KITTI validation set at IoU threshold 0.7.

GT Sampling. We do not adopt the GT sampling data augmentation for all settings for fair comparisons. The reason is that it is unaffordable for the iterative self-training pipeline to use GT sampling data augmentation since it requires frequently generating a new GT database with updated pseudo labels, which produces a large computation cost (leveraging GT sampling for self-training takes more than 3× training time).

More Analysis. Here, we show the oracle results trained with voxel size [0.05m, 0.05m, 0.15m] and GT sampling data augmentations. The results are listed in Table S2. Though our model performance presented in Table 1 in our paper is obtained using a sub-optimal setup for memory and computational efficiency, our adaptation results are still competitive in comparison with results in Table S2. Furthermore, employing PointRCNN as the framework, Oracle results in SN [11] even has 4.55% performance gap to our sub-optimal Oracle model. It is noteworthy that, the development of the ST3D model is orthogonal with the above modifications, and ST3D could also benefit from these training modifications and further boost the performance.

We would like to highlight that our focus in this paper is to demonstrate the effectiveness of ST3D without adopting various training tricks in 3D object detection. And we believe the presented comparisons in the main paper are fair and could assess the actual progress made by our ST3D pipeline.

S3.3. SECOND-IoU

Given the object proposals from the RPN head in the original SECOND network, we extract the proposal features from 2D BEV features using the rotated RoI-align operation [4]. Then, taking the extracted features as inputs, we adopt two fully connected layers with ReLU nonlinearity [1] and batch normalization [5] to regress the IoU between RoIs and their corresponding ground-truths (or pseudo boxes) with sigmoid nonlinearity. During training, we do not back-propagate the gradient from our IoU head $\mathcal{L}_{\text{iou}}$ to our backbone network. We observe the attached IoU branch could also boost the performance of the baseline SECOND model, namely SECOND-IoU, if the IoU prediction score is used for NMS.

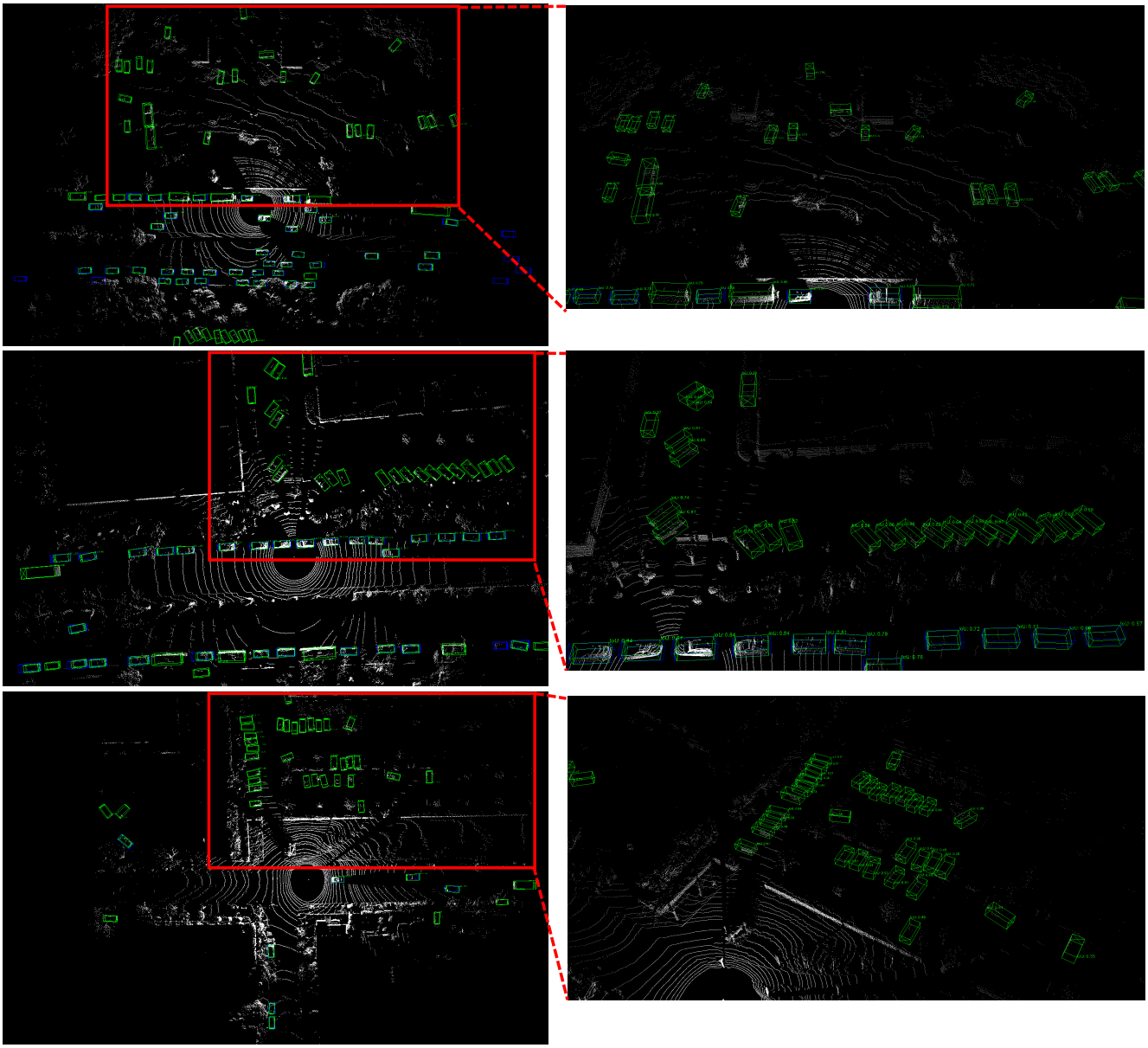


Figure S3. Examples to show the annotation gap between Lyft and Waymo. The green boxes are prediction results from the Waymo pre-trained model while the blue boxes are Lyft annotated boxes.

S3.4. Other Memory Ensemble Variants

NMS ensemble is an intuitive solution to match and merge boxes based on the IoU between two boxes. It directly removes matched boxes with lower confidence scores. Specifically, we concatenate historical pseudo labels and current proxy-pseudo labels to $[\tilde{M}_i^t]_k = \{[M_i^t]_{k-1}, [\hat{L}_i^t]_k\}$ as well as their corresponding confidence scores to $\tilde{u}_i^k = \{u_i^{k-n}, u_i^k\}$ for each target sample P_i^t. Then, we obtain the final pseudo boxes $[M_i^t]_k$ and corresponding confidence score u_i^k by applying NMS with a IoU threshold 0.1 as

$$[M_i^t]_k, u_i^k = \text{NMS}([\tilde{M}_i^t]_k,\ \tilde{u}_i^k). \tag{1}$$

Bipartite ensemble employs optimal bipartite matching to pair historical pseudo labels $[M_i^t]_{k-1}$ and current proxy-pseudo labels $[\hat{L}_i^t]_k$ and then follow consistency ensemble to process matched pairs. Concretely, we assume that there are n_m and n_l boxes for $[M_i^t]_{k-1}$ and $[\hat{L}_i^t]_k$ separately. Then, we search a permutation of n_m elements $\sigma \in \mathfrak{S}_{n_m}$

with the lowest cost as

$$\hat{\sigma} = \underset{\sigma \in \mathfrak{S}_{n_m}}{\arg\min} \sum_{j}^{n_m} \mathcal{L}_{\text{match}} \left(b_j, b_{\sigma(j)} \right), \tag{2}$$

where the matching cost $\mathcal{L}_{\text{match}}$ is the $-$IoU between the matched boxes. Notice that the matched box pairs with IoU lower than 0.1 would still be regarded as unmatched.

S4. Experimental Results with IoU = 0.5

In this section, we report the AP_{BEV} and AP_{3D} with the IoU threshold **0.5** as a supplement to the experimental results in our main submission. The results are shown in Table S5, S6, S7, S8 and S9, S10. To be noted, IoU threshold **0.7** is a more strict criterion and widely adopted to assess 3D object detection models for the "car" category [9, 12, 8, 11].

S5. Extra Ablation Studies

In this section, we present more ablation experiments and analysis. All experiments are conducted with the 3D detector SECOND-IoU on the adaptation setting of Waymo $\rightarrow$ KITTI. Our reported AP results are evaluated on the moderate difficulty of the car category of the KITTI dataset.

Method	Framework	Sequence	AP_{3D}	Closed Gap
Source Only	PointRCNN	unknown	21.9	-
SF-UDA^{3D} [7]	PointRCNN	$\surd$	**54.5**	56.0%
Oracle	PointRCNN		80.1	-
Source Only	SECOND-IoU		17.9	-
ST3D	SECOND-IoU		54.1	**65.1%**
Oracle	SECOND-IoU		73.5	-

Table S3. Comparison with SF-UDA^{3D} on nuScenes $\rightarrow$ KITTI.

Compared with the Contemporary SOTA. As shown in Table S3, SF-UDA^{3D} is a contemporary work that leverages the consistency of **temporal information** along with the point cloud sequences to address the domain shift on 3D object detection. By using **only the single-frame** point cloud as input, our ST3D achieves similar performance while being much closer to the fully-supervised oracle results.

Quality-aware Confidence Criterion. Here, we investigate the influence of the IoU confidence criterion on the pre-trained SN model, the self-training pipeline and the fully supervised oracle model, respectively. As illustrated in Table S4, the IoU score can bring performance improvements for all three settings in comparison with the classification score. Specifically, the IoU confidence yields a 2.12% gain for the SN model and a 0.44% gain for the fully supervised oracle model in terms of AP_{3D}. More importantly, our ST3D (w/ SN) self-training pipeline could benefit more from the IoU criterion, obtaining as much as 3.79% performance boost in items of AP_{3D}. This suggests that the IoU confidence criterion could facilitate the model to produce high-quality pseudo-labeled data, and ultimately lead to a much better 3D object detection model.

Method	Confidence	AP_{BEV} / AP_{3D}	Gain
SN	Classification	77.68 / 57.08	-
	IoU	78.96 / 59.20	1.28 / 2.12
ST3D (w/ SN)	Classification	82.21 / 69.58	-
	IoU	**85.83 / 73.37**	**3.62 / 3.79**
Oracle	Classification	84.48 / 73.01	
	IoU	83.29 / 73.45	-0.99 / 0.44

Table S4. Comparison of different confidence criteria.

Quality of Pseudo Labels. To directly investigate how each component contribute to the quality of pseudo labels, we utilize AP_{3D} and #TPs to assess the correctness of pseudo labels. Besides, **ATE**, **ASE** and **AOE** are to measure the translation, scale and orientation errors (refer to nuScenes toolkit [2]). As shown in Figure S4, ROS mitigates domain differences in object size distributions and hence largely reduces ASE; with Triplet, QAC and MEV, our method generates accurate and stable pseudo labels, localizing more #TPs with fewer errors; and CDA overcomes overfitting and reduces both ASE and AOE.

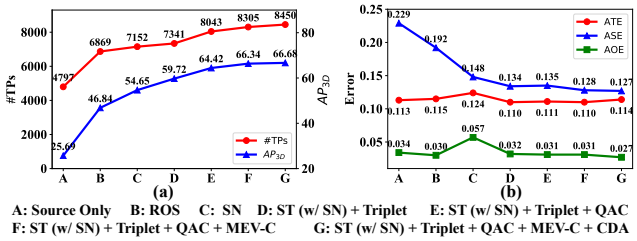
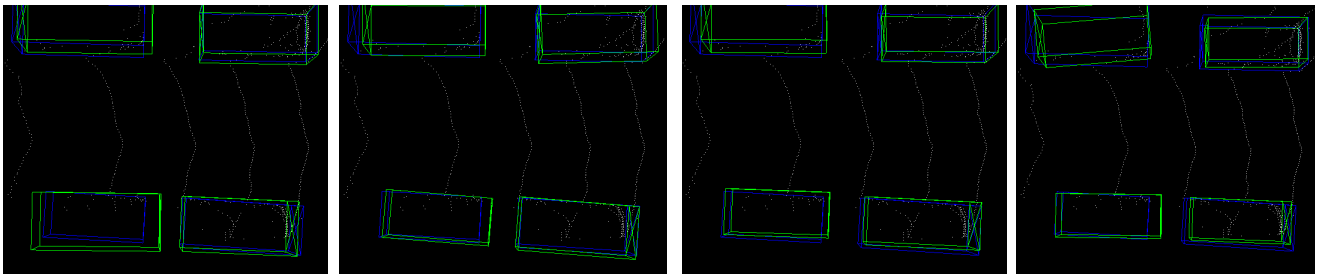


Figure S4. Quality of pseudo labels on KITTI training set.

S6. Qualitative Results

Qualitative Results of Random Object Scaling. We have compared the AP_{BEV} and AP_{3D} of our ROS with SN and Source Only model in the Table 2 of our main paper. Here, we provide qualitative results of the Source Only model, ROS, SN and Oracle for visual comparisons. As shown in Fig. S5, the zoom-in regions in the left bottom box in each sub-figure shows that both SN and ROS can largely improve the localization accuracy of the pre-trained model while our ROS does not leverage extra statistical information on the target domain.

Qualitative Results of ST3D. We provide some qualitative results of our proposed ST3D equipped with SN on the KITTI validation set as shown in Fig. S6. Our ST3D (w/ SN) could also predict high-quality object bounding boxes on various scenes with only adaptation and self-training manner.



(a) Source Only **(b) ROS** **(c) SN** **(d) Oracle**

Figure S5. Comparison of ROS and SN to close object-size level domain gap on Waymo → KITTI. The green and blue bounding boxes are detector predictions and GTs, respectively. (a) Source Only: The detector is trained on Waymo without SN or ROS. (b) The detector is trained with ROS on Waymo. (c) The detector is trained with SN [11] on Waymo. (d) The detector is trained on KITTI.

Method	AP_{BEV} / AP_{3D}
(a) Source Only	91.52 / 89.94
(b) Random Object Scale (ROS)	88.98 / 87.33
(c) SN	87.18 / 85.91
(d) Ours (w/o ROS)	**93.68 / 92.50**
(e) Ours (w/ ROS)	90.85 / 89.47
(f) Ours (w/ SN)	92.65 / 92.36

Table S5. Effectiveness analysis of Random Object Scaling (AP IoU threshold at 0.5).

Method	AP_{BEV} / AP_{3D}
SN (baseline)	87.18 / 85.91
ST (w/ SN)	86.17 / 85.86
ST (w/ SN) + Triplet	86.61 / 85.90
ST (w/ SN) + Triplet + QAC	91.76 / 90.79
ST (w/ SN) + Triplet + QAC + MEV-C	**93.57 / 92.95**
ST (w/ SN) + Triplet + QAC + MEV-C + CDA	92.65 / 92.36

Table S6. Component ablation studies (AP IoU threshold at 0.5). **ST** represents naive self-training. **Triplet** means the triplet box partition. **QAC** indicates the quality-aware criterion. **MEV-C** is consistency memory ensemble-and-voting. **CDA** means curriculum data augmentation.

T_{neg}	T_{pos}	AP_{BEV} / AP_{3D}	T_{neg}	T_{pos}	AP_{BEV} / AP_{3D}
0.20	0.60	**93.34 / 93.01**	0.25	0.25	91.48 / 90.93
0.25	0.60	92.65 / 92.36	0.25	0.30	91.17 / 90.70
0.30	0.60	93.16 / 92.00	0.25	0.40	92.05 / 91.63
0.40	0.60	92.97 / 90.96	0.25	0.50	**92.81** / 92.35
0.50	0.60	92.19 / 91.47	0.25	0.60	92.65 / **92.36**
0.60	0.60	92.16 / 90.40	0.25	0.70	83.08 / 82.90

Table S7. Sensitivity analysis for $[T_{neg}, T_{pos}]$ of triplet box partition (AP IoU threshold at 0.5).

S7. Experimental Result on More Tasks.

Our experiments in the main paper are designed to cover most practical scenarios (across different LiDAR beam ways and from label-rich domains to label insufficient domains), and we also rule out some ill-posed settings, such as we do not consider KITTI and Lyft as source domain since KITTI lacks of ring view annotations and Lyft has very difference annotations in our main paper (see Sec. S2.1 in supplementary materials). However, to validate the effectiveness of our method, we further conduct 5 extra experiments. Tab. S11 shows that, without tuning hyper-parameters, ST3D still achieves promising results on these five adaptation tasks.

Method	Memory Voting	Merge	AP_{BEV} / AP_{3D}
ST3D (w/ ME-N)	√	Max	**92.72 / 92.40**
ST3D (w/ ME-B)	√	Max	92.65 / 92.03
ST3D (w/ ME-C)	√	Max	92.65 / 92.36
	√	Avg	91.48 / 90.57
	×	Max	92.66 / 92.22
	×	Avg	90.80 / 90.50

Table S8. Ablation studies of memory ensemble (different variants and merge strategies for matched boxes) and memory voting (AP IoU threshold at 0.5). We denote three memory ensemble variants: consistency, NMS and bipartite as ME-C, ME-N, ME-B separately.

Method	World	Object	Intensity	AP_{BEV} / AP_{3D}
ST3D	×	×	-	83.31 / 66.73
	√	×	Normal	**93.62 / 93.21**
	×	√	Normal	91.36 / 89.85
	√	√	Normal	93.57 / 92.95
	√	√	Strong	92.42 / 91.49
	√	√	Curriculum	92.65 / 92.36

Table S9. Analysis of data augmentation types and intensities (AP IoU threshold at 0.5).

References

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. 3

[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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 5

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, 2012. 2

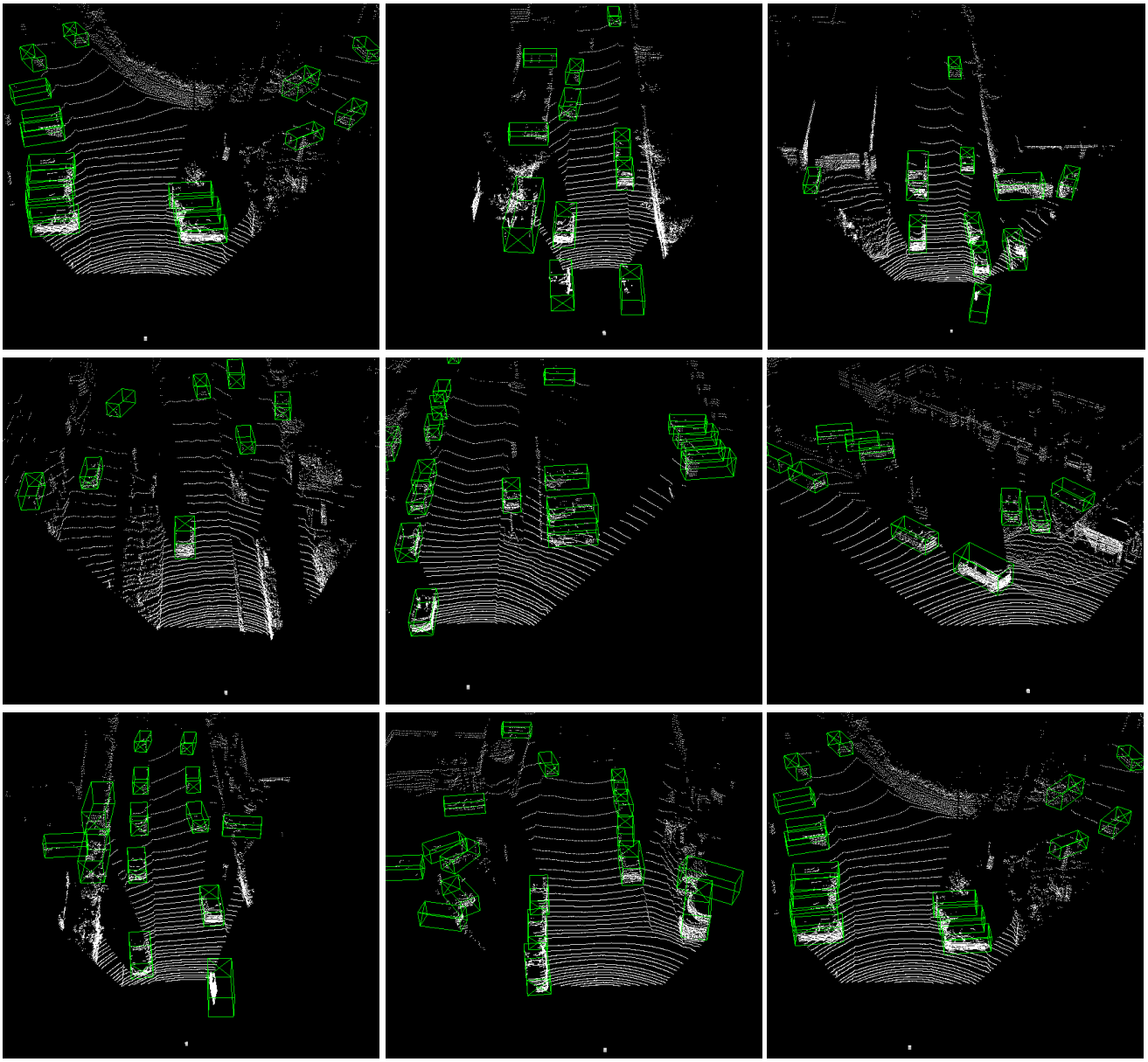


Figure S6. Qualitative results of Waymo → KITTI adaptation task.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 3

[5] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456. PMLR, 2015. 3

[6] 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. Lyft level 5 perception dataset 2020. https://level5.lyft.com/dataset/, 2019. 1, 2

[7] Cristiano Saltori, Stéphane Lathuiliére, Nicu Sebe, Elisa Ricci, and Fabio Galasso. Sf-uda3D: Source-free unsupervised domain adaptation for lidar-based 3d object detection. *arXiv preprint arXiv:2010.08243*, 2020. 5

[8] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 5

[9] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *arXiv preprint arXiv:1907.03670*, 2019. 5

Task	Method	SECOND-IoU	PVRCNN
Waymo → KITTI	Source Only	91.52 / 89.94	88.33 / 87.17
	SN [11]	87.18 / 85.91	86.32 / 85.72
	Ours	90.85 / 89.47	**92.40 / 92.18**
	Ours (w/ SN)	**92.65 / 92.36**	91.49 / 90.77
	Oracle	94.08 / 92.28	94.97 / 94.85
Waymo → Lyft	Source Only	81.82 / 79.73	82.38 / 80.45
	SN [11]	81.55 / 78.13	80.12 / 78.09
	Ours	**84.44 / 84.04**	**84.52 / 82.61**
	Ours (w/ SN)	83.98 / 83.40	82.21 / 81.70
	Oracle	94.62 / 92.32	92.38 / 91.87
Waymo → nuScenes	Source Only	43.32 / 37.58	40.48 / 36.95
	SN [11]	43.19 / 37.74	40.27 / 36.59
	Ours	**43.03** / 38.99	40.90 / 38.67
	Ours (w/ SN)	42.89 / **40.21**	**41.42 / 38.99**
	Oracle	63.17 / 58.91	61.52 / 58.04
nuScenes → KITTI	Source Only	84.32 / 79.18	80.88 / 78.47
	SN [11]	48.32 / 46.74	66.22 / 65.82
	Ours	85.59 / 83.62	83.75 / 83.64
	Ours (w/ SN)	**86.85 / 85.65**	**90.47 / 90.25**
	Oracle	94.08 / 92.28	94.97 / 94.85

Table S10. Result of different adaptation tasks. We report AP of the car category in AP_{BEV} and AP_{3D} at **IoU = 0.5**. The reported result is for the moderate case on the adaptation tasks with KITTI as target domain, and is the overall result for other adaptation tasks.

AP_{BEV} / AP_{3D}	nuScenes→ Waymo	nuScenes → Lyft	Lyft → KITTI	Lyft → Waymo	Lyft → nuScenes
Source Only	20.47 / 09.39	39.79 / 17.29	77.55 / 55.39	51.87 / 37.89	30.43 / 17.52
SN	19.83 / 03.17	34.65 / 14.15	81.08 / 65.01	51.85 / 39.42	30.18 / 18.13
ST3D	**49.29 / 23.86**	**58.12 / 33.48**	85.03 / 68.92	56.64 / 40.89	**33.26** / 19.76
ST3D (w/ SN)	25.24 / 11.00	51.20 / 26.41	**85.10 / 71.42**	**57.76 / 42.89**	32.89 / **21.49**
Oracle	65.01 / 51.12	84.47 / 68.78	83.29 / 73.45	65.01 / 51.12	51.88 / 34.87

Table S11. Result of other five adaptation tasks. Notice that we sample $\frac{1}{20}$ of Waymo training frames and $\frac{1}{10}$ of Waymo validation frames when Waymo serves as target domain.

[10] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2

[11] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 2, 3, 5, 6, 8

[12] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 5