3D DA Survey

Xianghao Kong

11/20/2022

CVPR 2020

Train in Germany, Test in The USA: Making 3D Object Detectors Generalize

Yan Wang*[1] Xiangyu Chen*[1] Yurong You[1] Li Erran Li[2,3]
Bharath Hariharan[1] Mark Campbell[1] Kilian Q. Weinberger[1] Wei-Lun Chao[4]
[1]Cornell University [2]Scale AI [3]Columbia University [4]The Ohio State University
{yw763, xc429, yy785, bh497, mc288, kqw4}@cornell.edu erranlli@gmail.com chao.209@osu.edu

Motivation

- Dataset biases
  - Lidar beams, orientation
  - The physical world being sensed
  - …

Figure 1: **Datasets.** We show frontal view images (left) and the corresponding LiDAR signals (right) from the bird's-eye view for five datasets: KITTI [18, 19], Argoverse [7], nuScenes [4], Lyft [25], and Waymo [3]. These datasets not only capture scenes at different geo-locations, but also use different LiDAR models, making generalizing 3D object detectors a challenging problem.

Table 1: Dataset overview. We focus on their properties related to frontal-view images, LiDAR, and 3D object detection. The dataset size refers to the number of synchronized (image, LiDAR) pairs. For Waymo and nuScenes, we subsample the data. See text for details.

Dataset	Size	LiDAR Type	Beam Angles	Object Types	Rainy Weather	Night Time
KITTI [18, 19]	14,999	1×64-beam	$[-24°, 4°]$	8	No	No
Argoverse [7]	22,305	2×32-beam	$[-26°, 25°]$	17	No	Yes
nuScenes [4]	34,149	1×32-beam	$[-16°, 11°]$	23	Yes	Yes
Lyft [25]	18,634	1×40 or $64 + 2 \times 40$-beam	$[-29°, 5°]$	9	No	No
Waymo [3]	192,484	$1 \times 64 + 4 \times 200$-beam	$[-18°, 2°]$	4	Yes	Yes

Contributions

- The core domain difference between self-driving car environments: **size statistics** of cars in different locations
- Simple and effective approach to mitigate this issue by using easily obtainable aggregate statistics of car size

Results and Analysis

- The detector tends to predict box sizes that are similar to the ground-truth sizes in source domain

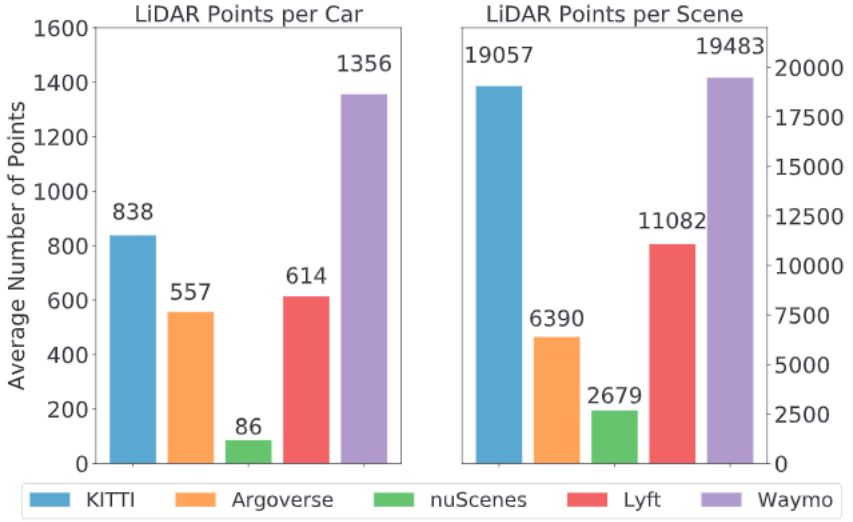


Figure 2: The average numbers of 3D points per car (left) and per scene (right). We only include points within the frontal-view camera view and cars whose depths are within 70 meters.

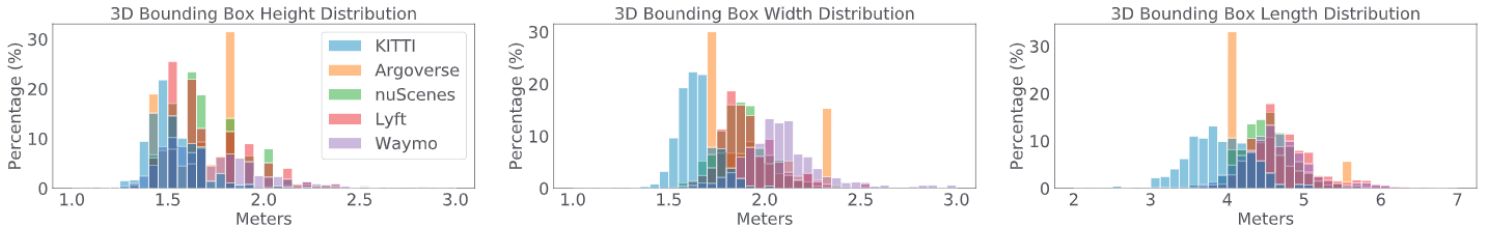


Figure 3: Car size statistics of different datasets.

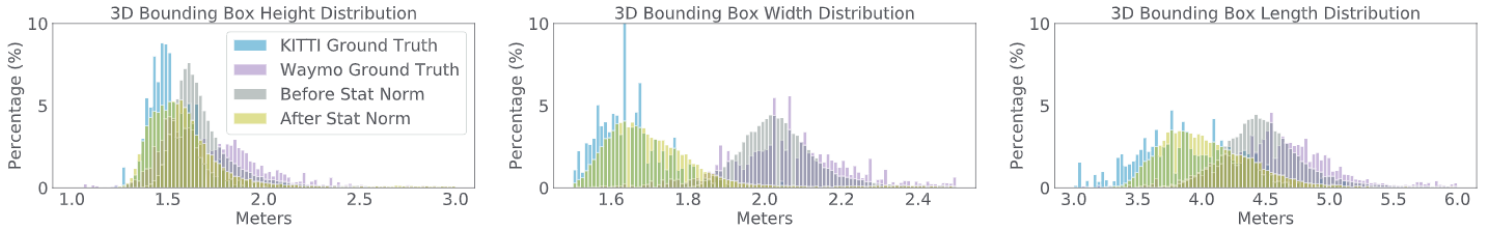


Figure 4: Sizes of detected bounding boxes before and after our Statistical Normalization (Stat Norm). The detector is trained on Waymo (w/o or w/ Stat Norm) and tested on KITTI. We also show the distribution of ground-truth box sizes in both datasets.

Method

- Few-shot fine-tuning
- Statistical normalization

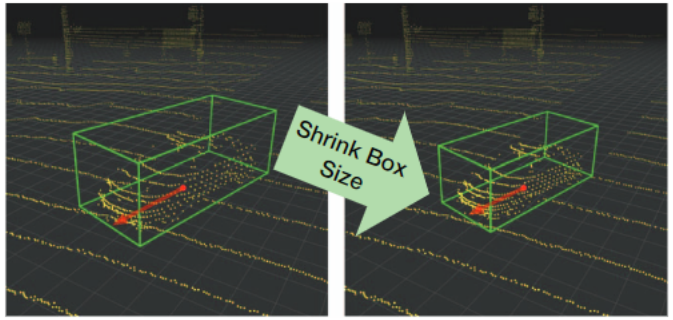


Figure 7: **Statistical Normalization (SN).** We shrink (or enlarge) the bounding box sizes (in the output space) and the corresponding point clouds (in the input space) in the training scenes of the source domain to match the *mean* statistics of the target domain. We fine-tune the detector with these modified source scenes.

Experiments

Table 2: **3D object detection across multiple datasets** (evaluated on the validation sets). We report average precision (AP) of the *Car* category in bird's-eye view (AP_{BEV}) and 3D (AP_{3D}) at IoU = 0.7, using the POINTRCNN detector [52]. We report results at different difficulties (following the KITTI benchmark, but we replace the 40, 25, 25 pixel thresholds on 2D bounding boxes with 30, 70, 70 meters on object depths, for *Easy*, *Moderate*, and *Hard* cases, respectively) and different depth ranges (using the same truncation and occlusion thresholds as KITTI *Hard* case). The results show a significant performance drop in cross-dataset inference. We indicate the best generalization results per column and per setting by red fonts and the worst by blue fonts. We indicate in-domain results by bold fonts.

Setting	Source\Target	KITTI	Argoverse	nuScenes	Lyft	Waymo
Easy	KITTI	**88.0 / 82.5**	55.8 / 27.7	47.4 / 13.3	81.7 / 51.8	45.2 / 11.9
	Argoverse	69.5 / 33.9	**79.2 / 57.8**	52.5 / 21.8	86.9 / 67.4	83.8 / 40.2
	nuScenes	49.7 / 13.4	73.2 / 21.8	**73.4 / 38.1**	89.0 / 38.2	78.8 / 36.7
	Lyft	74.3 / 39.4	77.1 / 45.8	63.5 / 23.9	**90.2 / 87.3**	87.0 / 64.7
	Waymo	51.9 / 13.1	76.4 / 42.6	55.5 / 21.6	87.9 / 74.5	**90.1 / 85.3**
Moderate	KITTI	**80.6 / 68.9**	44.9 / 22.3	26.2 / 8.3	61.8 / 33.7	43.9 / 12.3
	Argoverse	56.6 / 31.4	**69.9 / 44.2**	27.6 / 11.8	66.6 / 42.1	72.3 / 35.1
	nuScenes	39.8 / 10.7	56.6 / 17.1	**40.7 / 21.2**	71.4 / 25.0	68.2 / 30.8
	Lyft	61.1 / 34.3	62.5 / 35.3	33.6 / 12.3	**83.7 / 65.5**	77.6 / 53.2
	Waymo	45.8 / 13.2	64.4 / 29.8	28.9 / 13.7	74.2 / 53.8	**85.9 / 67.9**
Hard	KITTI	**81.9 / 66.7**	42.5 / 22.2	24.9 / 8.8	57.4 / 34.2	41.5 / 12.6
	Argoverse	58.5 / 33.3	**69.9 / 42.8**	26.8 / 14.5	64.4 / 42.7	68.5 / 36.8
	nuScenes	39.6 / 10.1	53.3 / 16.7	**40.2 / 20.5**	67.7 / 25.7	66.9 / 29.0
	Lyft	60.7 / 33.9	62.9 / 35.9	30.6 / 11.7	**79.3 / 65.5**	77.0 / 53.9
	Waymo	46.3 / 12.6	61.6 / 29.0	28.4 / 14.1	74.1 / 54.5	**80.4 / 67.7**
0-30m	KITTI	**88.8 / 84.9**	58.4 / 34.7	47.9 / 14.9	77.8 / 54.2	48.0 / 14.0
	Argoverse	74.2 / 46.8	**83.3 / 63.3**	55.3 / 26.9	87.7 / 69.5	85.7 / 44.4
	nuScenes	50.7 / 13.9	73.7 / 26.0	**73.2 / 42.8**	89.1 / 43.8	79.8 / 43.4
	Lyft	75.1 / 45.2	81.0 / 54.0	61.6 / 25.4	**90.4 / 88.5**	88.6 / 70.9
	Waymo	56.8 / 15.0	80.6 / 48.1	57.8 / 24.0	88.4 / 76.2	**90.4 / 87.2**
30m-50m	KITTI	**70.2 / 51.4**	46.5 / 19.0	9.8 / 4.5	60.1 / 34.5	50.5 / 21.4
	Argoverse	33.9 / 11.8	**72.2 / 39.5**	9.5 / 9.1	65.9 / 39.1	75.9 / 42.1
	nuScenes	24.1 / 3.8	46.3 / 6.4	**17.1 / 4.1**	70.1 / 18.9	69.4 / 29.2
	Lyft	39.3 / 16.6	59.2 / 21.8	11.2 / 9.1	**83.8 / 62.7**	79.4 / 55.5
	Waymo	31.7 / 9.3	58.0 / 18.8	9.9 / 9.1	74.5 / 51.4	**87.5 / 68.8**
50m-70m	KITTI	**28.8 / 12.0**	9.2 / 3.0	1.1 / 0.0	33.2 / 9.6	27.1 / 12.0
	Argoverse	10.9 / 1.3	**29.9 / 6.9**	0.5 / 0.0	35.1 / 14.5	46.2 / 23.0
	nuScenes	6.5 / 1.5	15.2 / 2.3	**9.1 / 9.1**	41.8 / 5.3	37.9 / 15.2
	Lyft	13.6 / 4.6	23.1 / 3.9	1.1 / 0.0	**62.7 / 33.1**	54.6 / 27.5
	Waymo	5.6 / 1.8	26.9 / 5.6	0.9 / 0.0	50.8 / 21.3	**63.5 / 41.1**

Table 4: **Improved 3D object detection across datasets** (evaluated on the validation sets). We report AP_{BEV}/ AP_{3D} of the *Car* category at IoU = 0.7, using POINTRCNN [52]. We investigate **(OT)** *output transformation* by directly adjusting the predicted box sizes, **(SN)** *statistical normalization*, and **(FS)** *few-shot fine-tuning* (with 10 labeled instances). We also include **(Direct)** directly applying the detectors trained on the source domain and **(Within)** applying the detectors trained on the target domain for comparison. We show adaption results from KITTI to other datasets, and vice versa. We mark the best result among Direct, OT, SN, and FS in red fonts, and worst in blue fonts.

		From KITTI (KITTI as the source; others as the target)					**To KITTI** (KITTI as the target; others as the source)				
Setting	Dataset	Direct	OT	SN	FS	Within	Direct	OT	SN	FS	Within
Easy	Argoverse	55.8 / 27.7	72.7 / 9.0	74.7 / 48.2	75.8 / 49.2	79.2 / 57.8	69.5 / 33.9	53.3 / 5.7	76.2 / 46.1	80.0 / 49.7	88.0 / 82.5
	nuScenes	47.4 / 13.3	55.0 / 10.4	60.8 / 23.9	54.7 / 21.7	73.4 / 38.1	49.7 / 13.4	75.4 / 31.5	83.2 / 35.6	83.8 / 58.7	88.0 / 82.5
	Lyft	81.7 / 51.8	88.2 / 23.5	88.3 / 73.3	89.0 / 78.1	90.2 / 87.3	74.3 / 39.4	71.9 / 4.7	83.5 / 72.1	85.3 / 72.5	88.0 / 82.5
	Waymo	45.2 / 11.9	86.1 / 16.2	84.6 / 53.3	87.4 / 70.9	90.1 / 85.3	51.9 / 13.1	64.0 / 3.9	82.1 / 48.7	81.0 / 67.0	88.0 / 82.5
Mod.	Argoverse	44.9 / 22.3	59.9 / 7.9	61.5 / 38.2	60.7 / 37.3	69.9 / 44.2	56.6 / 31.4	52.2 / 7.3	67.2 / 40.5	68.8 / 42.8	80.6 / 68.9
	nuScenes	26.2 / 8.3	30.8 / 6.8	32.9 / 16.4	28.7 / 12.5	40.7 / 21.2	39.8 / 10.7	58.5 / 27.3	67.4 / 31.0	67.2 / 45.5	80.6 / 68.9
	Lyft	61.8 / 33.7	70.1 / 17.8	73.7 / 53.1	74.2 / 53.4	83.7 / 65.5	61.1 / 34.3	60.8 / 5.6	73.6 / 57.9	73.9 / 56.2	80.6 / 68.9
	Waymo	43.9 / 12.3	69.1 / 13.1	74.9 / 49.4	75.9 / 55.3	85.9 / 67.9	45.8 / 13.2	54.9 / 3.7	71.3 / 47.1	66.8 / 51.8	80.6 / 68.9
Hard	Argoverse	42.5 / 22.2	59.3 / 9.3	60.6 / 37.1	59.8 / 36.5	69.9 / 42.8	58.5 / 33.3	53.5 / 8.6	68.5 / 41.9	66.3 / 43.0	81.9 / 66.7
	nuScenes	24.9 / 8.8	27.8 / 7.6	31.9 / 15.8	27.5 / 12.4	40.2 / 20.5	39.6 / 10.1	59.5 / 27.8	65.2 / 30.8	64.7 / 44.5	81.9 / 66.7
	Lyft	57.4 / 34.2	66.5 / 19.1	73.1 / 53.5	71.8 / 52.9	79.3 / 65.5	60.7 / 33.9	63.1 / 6.9	75.2 / 58.9	74.1 / 56.2	81.9 / 66.7
	Waymo	41.5 / 12.6	68.7 / 13.9	69.4 / 49.4	70.1 / 54.4	80.4 / 67.7	46.3 / 12.6	58.0 / 4.1	73.0 / 49.7	68.1 / 52.9	81.9 / 66.7
0-30	Argoverse	58.4 / 34.7	73.0 / 13.7	73.1 / 54.2	73.6 / 55.2	83.3 / 63.3	74.2 / 46.8	64.9 / 10.1	83.3 / 53.9	84.0 / 56.9	88.8 / 84.9
	nuScenes	47.9 / 14.9	56.2 / 13.9	60.0 / 29.2	54.0 / 23.6	73.2 / 42.8	50.7 / 13.9	74.6 / 36.6	83.6 / 42.8	81.2 / 59.8	88.8 / 84.9
	Lyft	77.8 / 54.2	88.4 / 27.5	88.8 / 75.4	89.3 / 77.6	90.4 / 88.5	75.1 / 45.2	74.8 / 9.1	87.4 / 73.6	87.5 / 73.9	88.8 / 84.9
	Waymo	48.0 / 14.0	87.7 / 22.2	87.1 / 60.1	88.7 / 74.1	90.4 / 87.2	56.8 / 15.0	71.3 / 4.4	85.7 / 59.0	84.8 / 71.0	88.8 / 84.9
30-50	Argoverse	46.5 / 19.0	56.1 / 5.4	61.5 / 31.5	59.0 / 29.9	72.2 / 39.5	33.9 / 11.8	35.1 / 9.1	48.9 / 25.7	47.9 / 23.8	70.2 / 51.4
	nuScenes	9.8 / 4.5	10.8 / 9.1	11.0 / 2.3	9.5 / 6.1	17.1 / 4.1	24.1 / 3.8	35.5 / 15.5	44.9 / 18.6	45.0 / 25.1	70.2 / 51.4
	Lyft	60.1 / 34.5	67.4 / 10.7	73.8 / 52.2	73.7 / 50.4	83.8 / 62.7	39.3 / 16.6	43.3 / 3.9	58.3 / 38.0	57.7 / 33.3	70.2 / 51.4
	Waymo	50.5 / 21.4	73.6 / 10.4	78.1 / 54.9	78.1 / 57.2	87.5 / 68.8	31.7 / 9.3	39.8 / 4.5	57.3 / 36.3	49.2 / 29.2	70.2 / 51.4
50-70	Argoverse	9.2 / 3.0	20.5 / 1.0	23.8 / 5.6	20.1 / 6.3	29.9 / 6.9	10.9 / 1.3	8.0 / 0.8	9.1 / 2.6	8.1 / 3.8	28.8 / 12.0
	nuScenes	1.1 / 0.0	1.5 / 1.0	3.0 / 2.3	3.3 / 1.2	9.1 / 9.1	6.5 / 1.5	7.8 / 5.1	9.4 / 5.1	12.9 / 5.7	28.8 / 12.0
	Lyft	33.2 / 9.6	41.3 / 6.8	49.9 / 22.2	46.8 / 19.4	62.7 / 33.1	13.6 / 4.6	12.7 / 0.9	21.1 / 6.7	17.5 / 8.0	28.8 / 12.0
	Waymo	27.1 / 12.0	42.6 / 4.2	46.8 / 25.1	45.2 / 24.3	63.5 / 41.1	5.6 / 1.8	7.7 / 1.1	14.4 / 5.7	10.5 / 4.8	28.8 / 12.0

3DV 2020

SF-UDA^{3D}: Source-Free Unsupervised Domain Adaptation for LiDAR-Based 3D Object Detection

Cristiano Saltori
University of Trento
cristiano.saltori@unitn.it

Stéphane Lathuilière
LTCI, Télécom Paris, Institut Polytechnique de Paris
stephane.lathuiliere@telecom-paris.fr

Nicu Sebe
University of Trento
Huawei Research
niculae.sebe@unitn.it

Elisa Ricci
University of Trento
Fondazione Bruno Kessler
eliricci@fbk.eu

Fabio Galasso
Sapienza University of Rome
galasso@di.uniroma1.it

Motivation

- LiDAR-based detectors are prone to domain shift issues
- The density of the LiDAR point cloud, spatial resolution and ranges

- Not access source domain when adapting

Contributions

- New problem: source free UDA
- New method: pseudo-annotations, reversible scale-transformations and motion coherency
- SOTA

Method

1. Lowest MVV
2. Sample from W*

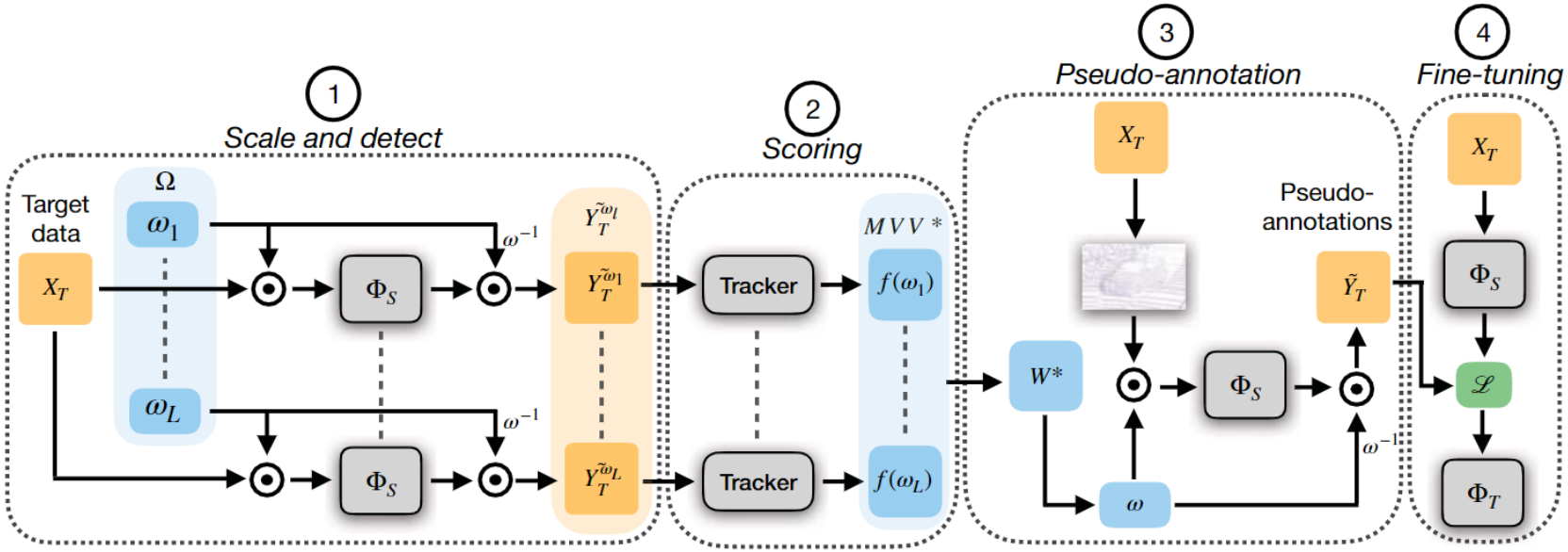


Figure 2: **Full SF-UDA^{3D} pipeline overview.** Given a scaling solution space Ω, in the first step detections over target sequences are obtained by scaling input data by ω and by re-scaling predictions by $1/\omega$. Next, time consistency of each sequence is used through a tracker to score each solution. During the third stage, scores are used to identify the best scaling interval W^* and pseudo-annotations are obtained over multiple iterations with the same procedure of step one and are merged through NMS. Finally, we obtain the target adapted model $\Phi_{\mathcal{T}}$ by fine-tuning the source model over target data and pseudo-annotations.

Scale Scoring with Temporal Consistency

- Utilize a tracker
- Stable volume: good detector

$$MVV(V) = \frac{1}{J}\sum_{j=1}^{J}\sqrt{\frac{\sum_{t=t_j}^{t_j+T_j}(v_j^t - \bar{v_j})^2}{T_j - 1}}$$

$$MVV^*(V) = \begin{cases} MVV(V), & J \neq 0 \\ H^*, & \text{if no tracks} \end{cases}$$

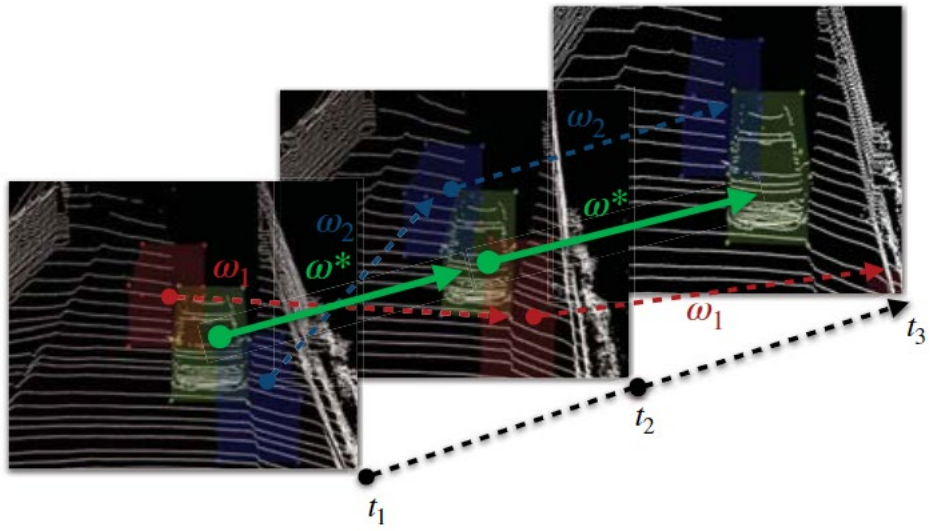


Figure 3: Given multiple possible scales ω, SF-UDA^{3D} selects the best ω^* as the one generating the most time consistent detections.

Experiments

Method	Easy	Moderate	Hard	Avg-AP
Source	0.273	0.196	0.188	0.219
AdaBN [22]	0.277	0.200	0.188	0.222
OT [41]	0.199	0.166	0.153	0.173
FS [41]	0.506	0.436	0.396	0.446
SF-UDA^{3D}(SS)	0.589	0.414	0.388	0.464
SF-UDA^{3D}(MS-3)	**0.688**	**0.498**	**0.450**	**0.545**
SF-UDA^{3D}(MS-5)	0.657	0.479	0.427	0.521
Target	0.873	0.769	0.760	0.801

Table 2: Adaptation results: nuScenes→KITTI

Method	AP-0.5	AP-1.0	AP-2.0	AP-4.0	Avg-AP
Source	0.143	0.208	0.224	0.234	0.202
AdaBN [22]	0.144	0.208	0.224	0.234	0.203
OT [41]	0.124	0.202	0.224	0.233	0.196
FS [41]	0.170	0.211	0.235	0.250	0.216
SF-UDA^{3D}(SS)	0.136	0.260	**0.290**	**0.308**	0.249
SF-UDA^{3D}(MS-3)	0.203	**0.266**	**0.290**	**0.308**	0.267
SF-UDA^{3D}(MS-5)	**0.211**	0.264	0.288	0.307	**0.268**
Target	0.370	0.422	0.440	0.455	0.422

Table 3: Adaptation results: KITTI→nuScenes

CVPR 2021

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

Motivation

- Domain shifts
  - Different types of 3D sensors
  - Weather conditions
  - Geographical locations
- Costly to collect data

- Few 3D UDA works
- 2D approaches not readily applicable

- Self-training
  - Pretrain on **labeled source**
  - Iterating between pseudo label generation and model training on **unlabeled target**
  - Naïve self-training doesn't work well

Contributions

- Pretraining: **random object scaling**
- Pseudo label generation: **quality-aware triplet memory bank**
- Training: **curriculum data augmentation**, progressively increasing the intensity of augmentation

Compared to fully-supervised

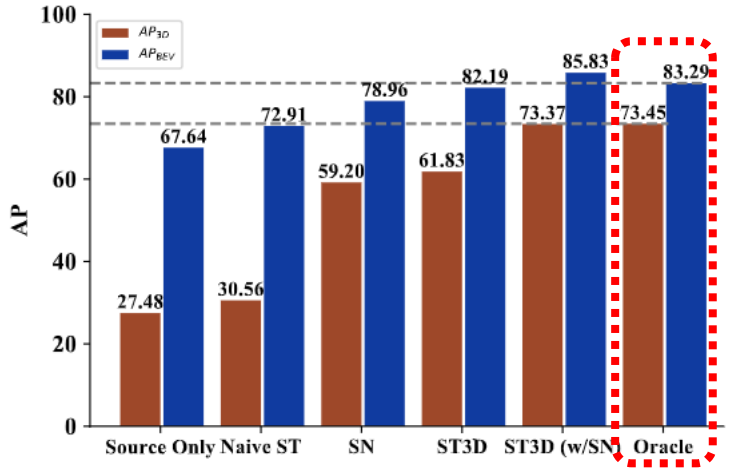


Figure 1. Performance of ST3D on Waymo → KITTI task using SECOND-IoU [46], compared to other unsupervised (*i.e.* source only, naive ST), weakly-supervised (*i.e.* SN [41]) and fully supervised (*i.e.* oracle) approaches. Dashed line denotes fully supervised target labeled data trained SECOND-IoU.

Method

Apply random scaling on bounding boxes

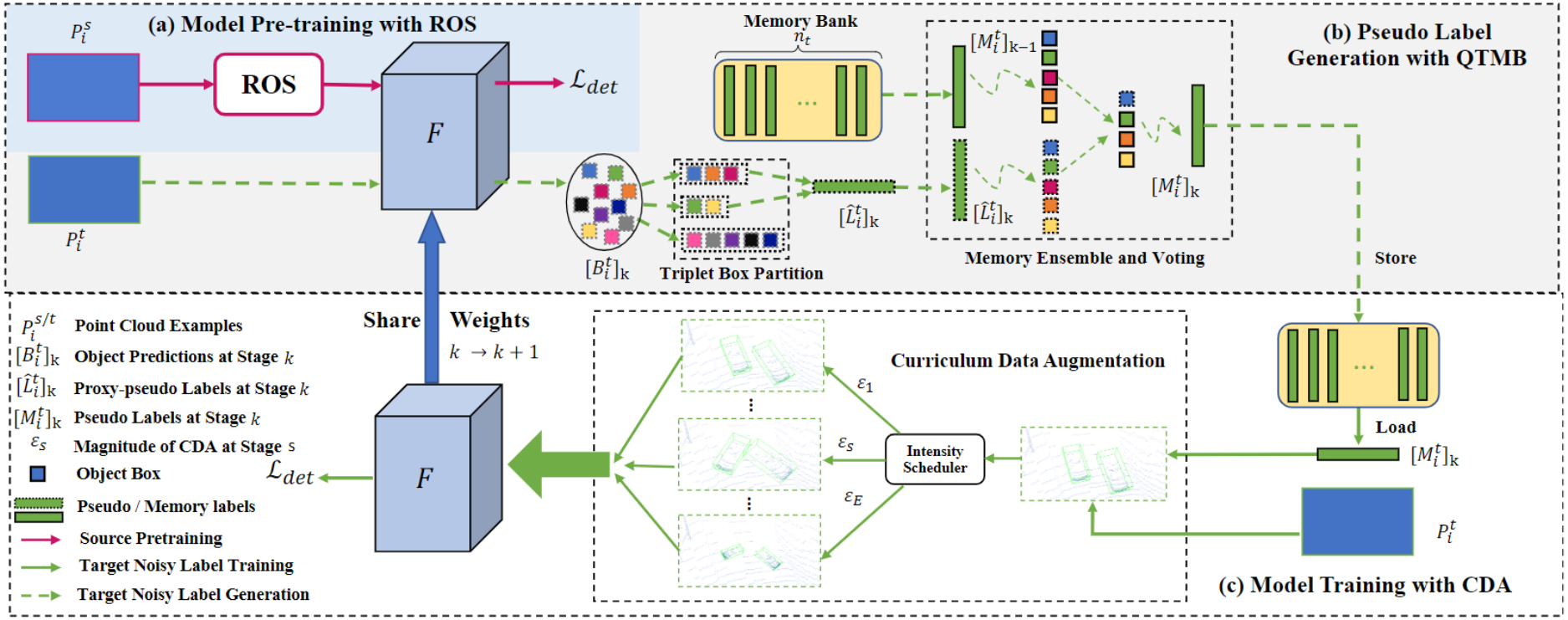


Figure 2. Our ST3D framework consists of three phases: (a) Pre-train the object detector F with ROS in source domain to mitigate object-size bias. (b) Generate high-quality and consistent pseudo labels on target unlabeled data with our QTMB. (c) Train model effectively on pseudo-labeled target data with CDA to progressively simulate hard examples. Best viewed in color.

Problems in Pseudo label Generation

- The confidence of object category prediction may **not necessarily reflect the precision of location**
- The **fraction of false labels** is much increased in confidence score intervals with medium values
- **Model fluctuations** induce inconsistent pseudo labels

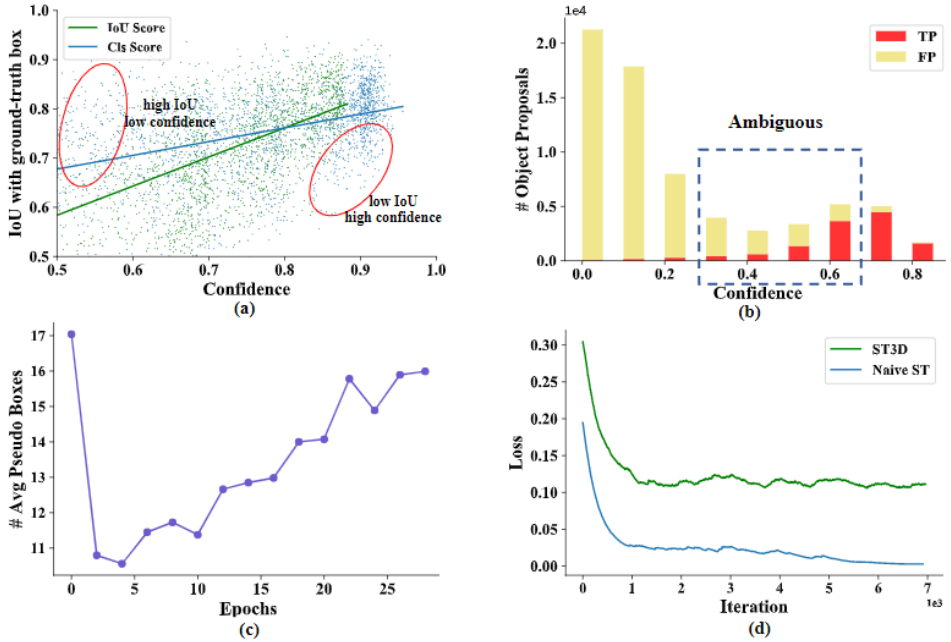


Figure 3. (a) Correlation between confidence value and box IoU with ground-truth (b) Lots of boxes with medium confidence may be assigned with ambiguous labels. (c) The average number of pseudo boxes fluctuates at different epochs. (d) Training loss curve comparison between naive ST and our ST3D with CDA.

Quality-aware Triplet Memory Bank

- Add a lightweight IoU regression head
- Avoid assigning labels to ambiguous examples

$$\text{state}_b = \begin{cases} \text{Positive (Store to } [\hat{L}_i^t]_k), & T_{\text{pos}} \le u_b, \\ \text{Ignored (Store to } [\hat{L}_i^t]_k), & T_{\text{neg}} \le u_b < T_{\text{pos}} \\ \text{Negative (Discard)}, & u_b < T_{\text{neg}}. \end{cases} \tag{4}$$

Ignored in training

- Match the history boxes with the latest proxy-pseudo labels
- Memory Voting

$$(\text{cnt}_b)_j^k = \begin{cases} 0 & , \quad \text{if } b \in [\hat{L}_i^t]_k, \\ (\text{cnt}_b)_j^{k-1} + 1 & , \quad \text{if } b \in [M_i^t]_{k-1}, \end{cases}$$

$$\begin{cases} \text{Discard} & , \quad (\text{cnt}_b)_j^k \ge T_{\text{rm}}, \\ \text{Ignore (Store to} [M_i^t]_k) & , \quad T_{\text{ign}} \le (\text{cnt}_b)_j^k < T_{\text{rm}}, \\ \text{Cache (Store to} [M_i^t]_k) & , \quad (\text{cnt}_b)_j^k < T_{\text{ign}}. \end{cases}$$

Quality-aware Triplet Memory Bank (Cont.)

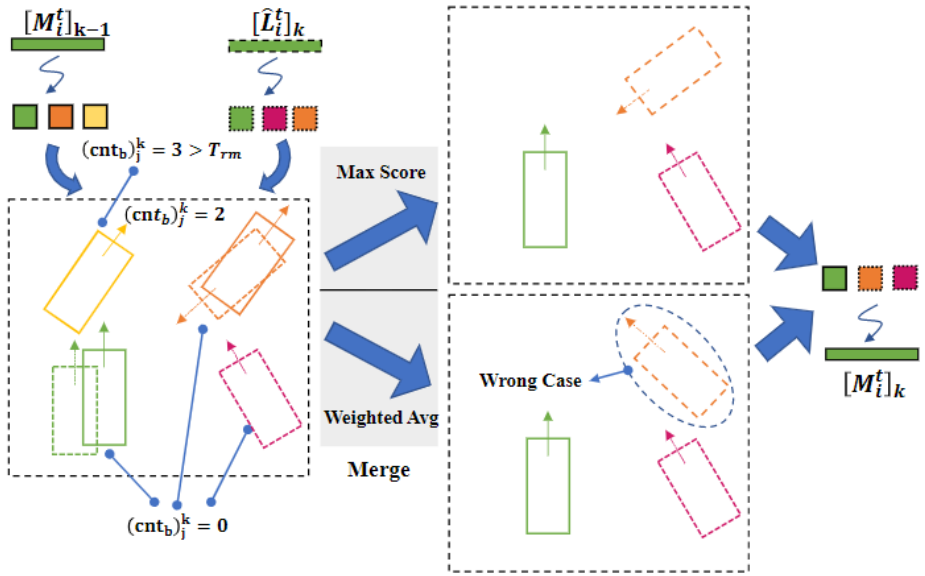


Figure 4. An instance of memory ensemble and voting (MEV). Given proxy-pseudo labels and historical memory labels, MEV automatically matches and merges boxes while ignoring or discarding successively unmatched boxes. The weighted average boxes merging strategy could produce wrong final box for boxes with very different heading angles.

Model training with CDA

- Most of positive pseudo boxes are **easy examples** since they are generated from previous high-confident object predictions
- Strong augmentation might confuse the learner and hence be harmful to model training at the initial stage

- Gradually generate **increasingly harder examples** to facilitate improving the model and ensure effective learning at the early stages

Experiments

We also observe that it is hard to adapt detectors from the point clouds with more LiDAR beams (*e.g.* Waymo) to the point clouds with fewer LiDAR beams (*e.g.* NuScenes), while the opposite adaptation is relatively easy as shown in Table 1 nuScenes → KITTI. It demonstrates that the point

Task	Method	SECOND-IoU		PV-RCNN	
		AP_{BEV} / AP_{3D}	Closed Gap	AP_{BEV} / AP_{3D}	Closed Gap
Waymo → KITTI	Source Only	67.64 / 27.48	-	61.18 / 22.01	-
	SN [41]	78.96 / 59.20	+72.33% / +69.00%	79.78 / 63.60	+66.91% / +68.76%
	ST3D	82.19 / 61.83	+92.97% / +74.72%	84.10 / 64.78	+82.45% / +70.71%
	ST3D (w/ SN)	**85.83** / **73.37**	+116.23% / +99.83%	**86.65** / **76.86**	+91.62% / +90.68%
	Oracle	83.29 / 73.45	-	88.98 / 82.50	-
Waymo → Lyft	Source Only	72.92 / 54.34	-	75.49 / 58.53	-
	SN [41]	72.33 / 54.34	-05.11% / +00.00%	72.82 / 56.64	-24.34% / -14.36%
	ST3D	76.32 / **59.24**	+29.44% / +33.93%	**77.68** / **60.53**	+19.96% / +15.20%
	ST3D (w/ SN)	**76.35** / 57.99	+15.71% / +17.81%	74.95 / 58.54	-04.92% / +00.08%
	Oracle	84.47 / 68.78	-	86.46 / 71.69	-
Waymo → nuScenes	Source Only	32.91 / 17.24	-	34.50 / 21.47	-
	SN [41]	33.23 / 18.57	+01.69% / +07.54%	34.22 / 22.29	-01.50% / +04.80%
	ST3D	**35.92** / 20.19	+15.87% / +16.73%	36.42 / 22.99	+10.32% / +08.89%
	ST3D (w/ SN)	35.89 / **20.38**	+15.71% / +17.81%	**36.62** / **23.67**	+11.39% / +12.87%
	Oracle	51.88 / 34.87	-	53.11 / 38.56	-
nuScenes → KITTI	Source Only	51.84 / 17.92	-	68.15 / 37.17	-
	SN [41]	40.03 / 21.23	-37.55% / +05.96%	60.48 / 49.47	-36.82% / +27.13%
	ST3D	75.94 / 54.13	+76.63% / +59.50%	78.36 / 70.85	+49.02% / +74.30%
	ST3D (w/ SN)	**79.02** / **62.55**	+86.42% / +80.37%	**84.29** / **72.94**	+77.48% / +78.91%
	Oracle	83.29 / 73.45	-	88.98 / 82.50	-

Table 1. Result of different adaptation tasks. We report AP_{BEV} and AP_{3D} of the car category at IoU = 0.7 as well as the domain gap closed by various approaches along Source Only and Oracle. The reported AP is moderate case for the adaptation tasks for to KITTI tasks, and is the overall result for other adaptation tasks. We indicate the best adaptation result by **bold**.

Ablations

Method	AP_{BEV} / AP_{3D}
(a) Source Only	67.64 / 27.48
(b) Random Object Scaling (ROS)	78.07 / 54.67
(c) SN	78.96 / 59.20
(d) ST3D (w/o ROS)	75.54 / 34.76
(e) ST3D (w/ ROS)	82.19 / 61.83
(f) ST3D (w/ SN)	**85.83 / 73.37**

Table 2. Effectiveness analysis of Random Object Scaling.

Method	AP_{BEV} / AP_{3D}
SN (baseline)	78.96 / 59.20
ST (w/ SN)	79.74 / 65.88
ST (w/ SN) + Triplet	79.81 / 67.39
ST (w/ SN) + Triplet + QAC	83.76 / 70.64
ST (w/ SN) + Triplet + QAC + MEV-C	85.35 / 72.52
ST (w/ SN) + Triplet + QAC + MEV-C + CDA	**85.83 / 73.37**

Table 3. Component ablation studies. **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.

TPAMI 2022

ST3D++: Denoised Self-training for Unsupervised Domain Adaptation on 3D Object Detection

Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, Xiaojuan Qi

Improvements

- Multiple categories
- More analysis on pseudo label noise
- A hybrid quality-aware criterion to account for both the localization quality and the classification accuracy when assessing pseudo labels
- Source-assisted self-denoised (SASD) training to further leverage source examples in the self-training stage
- **5 quantitative indicators** on the quality of pseudo labels

Analysis on pseudo label noise

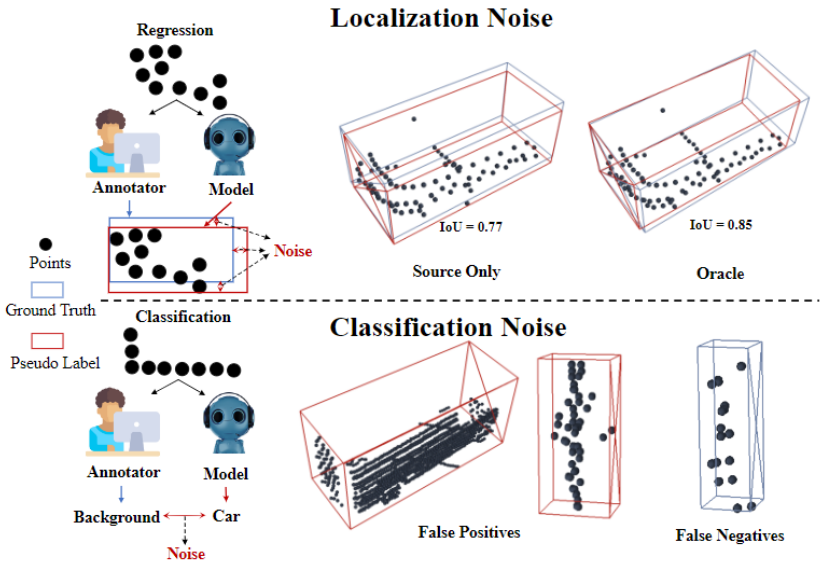


Fig. 2. Examples of two types self-training noise (red and blue boxes stand for predicted pseudo labels and GTs, respectively). Upper row: "localization noise" lies in the bounding box difference between model predictions and their corresponding human-annotated GTs. IoU is to measure the closeness of the pseudo labels and the GTs and thus localization noise. The smaller IoU is, the larger localization noise is. Bottom row: "classification noise" lies in the misclassified object proposals (*i.e.* false positives) and undetected objects (*i.e.* false negatives), which will confuse the model in the self-training process in the aspect of recognizing object proposals.

Hybrid quality-aware criterion

- For object categories that are **easily distinguishable from backgrounds** (e.g. “cars”), the IoU score not only correlates well with localization quality, but also generates more true positives (TPs) in confident if adopted as a criterion
- The classification score enjoys an obvious superiority in **categories that are easily confused with backgrounds** (e.g. pedestrians similar to background “trees” and “ poles”)

- Fuse IoU score and classification score $o = \phi p + (1 - \phi)u,$

Source-assisted self-denoised (SASD) training

- Domain-specific BN
- Joint optimization

- Rectify noisy gradients
- Provide challenging cases
- Negative transfer alleviated

Domain gaps across different dataset

TABLE 2
Dataset overview. Note that we use **version 1.0** of Waymo Open Dataset. * indicates that the information is obtained from [11]. † means that we count this statistical information only on the validation set.

Dataset	# Beam Ways	Beam Angles	# Points Per Scene†	# Training Frames	# Validation Frames	Location	# Night/Rain
Waymo [8]	64-beam	[-18.0°, 2.0°]*	160k	158,081	39,987	USA	Yes/Yes
KITTI [7]	64-beam	[-23.6°, 3.2°]	118k	3,712	3,769	Germany	No/No
Lyft [10]	64-beam	[-29.0°, 5.0°]*	69k	18,900	3,780	USA	No/No
nuScenes [9]	32-beam	[-30.0°, 10.0°]	25k	28,130	6,019	USA and Singapore	Yes/Yes

Experiments

TABLE 3
Results of four different adaptation tasks. We report average precision (AP) in bird's-eye view (AP_{BEV}) and 3D (AP_{3D}) of the car, pedestrian and cyclist at IoU threshold as 0.7, 0.5 and 0.5 respectively. The reported AP is for the moderate case when KITTI dataset is the source domain, and is the overall result for other settings. Note that results of ST3D [26] on pedestrian and cyclist are reproduced since they are not given. We indicate the best adaptation result by **bold**. ‡ indicates we apply random world sampling as an extra data augmentation on the source domain.

Task	Method	Car	Pedestrian	Cyclist
Waymo → KITTI	Source Only	67.64 / 27.48	46.29 / 43.13	48.61 / 43.84
	SN [1]	78.96 / 59.20	53.72 / 50.44	44.61 / 41.43
	ST3D	82.19 / 61.83	52.92 / 48.33	53.73 / 46.09
	ST3D (w/ SN)	85.83 / 73.37	54.74 / 51.92	56.19 / 53.00
	ST3D++	80.78 / 65.64	57.13 / 53.87	57.23 / 53.43
	ST3D++ (w/ SN)	**86.47 / 74.61**	**62.10 / 59.21**	**65.07 / 60.76**
	Oracle (w/o GT-Paste)	83.52 / 73.53	33.87 / 28.27	30.14 / 27.27
	Oracle (w/ GT-Paste)	88.60 / 82.07	46.16 / 41.72	62.55 / 60.15
Waymo → Lyft	Source Only	72.92 / 54.34	37.87 / 33.40	33.47 / 28.90
	SN [1]	72.33 / 54.34	39.07 / 33.59	30.21 / 23.44
	ST3D	76.32 / 59.24	36.50 / 32.51	35.06 / 30.27
	ST3D (w/ SN)	76.35 / 57.99	37.53 / 33.28	31.77 / 26.34
	ST3D++	**79.61 / 59.93**	**40.17 / 35.47**	**37.89 / 34.49**
	ST3D++ (w/ SN)	76.67 / 58.86	37.89 / 34.49	37.73 / 32.05
	Oracle (w/o GT-Paste)	84.58 / 68.86	43.36 / 34.00	38.69 / 33.50
	Oracle (w/ GT-Paste)	86.86 / 69.62	48.29 / 38.47	43.42 / 38.68
Waymo → nuScenes	Source Only	32.91 / 17.24	7.32 / 5.01	3.50 / 2.68
	SN [1]	33.23 / 18.57	7.29 / 5.08	2.48 / 1.8
	ST3D	35.92 / 20.19	5.75 / 5.11	4.70 / 3.35
	ST3D (w/ SN)	35.89 / 20.38	5.95 / 5.30	2.5 / 2.5
	ST3D++‡	35.73 / 20.90	12.19 / 8.91	**5.79 / 4.84**
	ST3D++ (w/ SN)‡	**36.65 / 22.01**	**15.50 / 12.13**	5.78 / 4.70
	Oracle (w/o GT-Paste)	50.37 / 33.07	25.10 / 18.46	10.59 / 8.05
	Oracle (w/ GT-Paste)	50.70 / 34.52	26.45 / 19.57	14.58 / 11.15
nuScenes → KITTI	Source Only	51.84 / 17.92	39.95 / 34.57	17.70 / 11.08
	SN [1]	40.03 / 21.23	38.91 / 34.36	11.11 / 5.67
	ST3D	75.94 / 54.13	44.00 / 42.60	29.58 / 21.21
	ST3D (w/ SN)	79.02 / 62.55	43.12 / 40.54	16.60 / 11.33
	ST3D++	80.56 / 66.01	48.00 / 45.23	31.65 / 25.98
	ST3D++ (w/ SN)	81.92 / 66.24	49.66 / 46.75	26.39 / 22.66
	Oracle (w/o GT-Paste)	83.52 / 73.53	33.87 / 28.27	30.14 / 27.27
	Oracle (w/ GT-Paste)	88.60 / 82.07	46.16 / 41.72	62.55 / 60.15

- Content gap caused by different locations and time: Waymo→KITTI and nuScenes→KITTI
- Object size gap: KITTI and others
- Point distribution gap owing to different LiDAR types: nuScenes→KITTI and Waymo→KITTI, **sparse to dense is better**

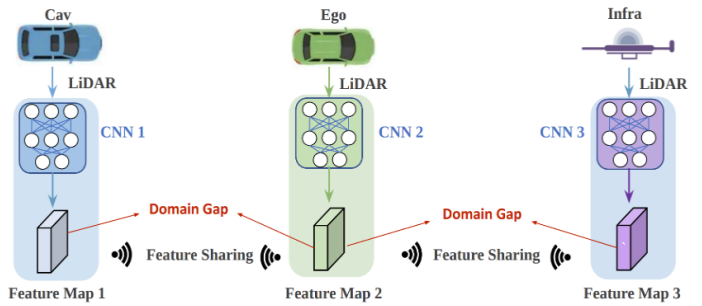
arXiv 2210

Bridging the Domain Gap for Multi-Agent Perception

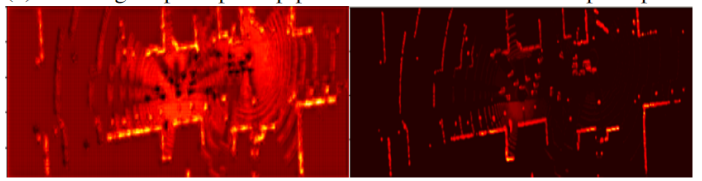
Runsheng Xu[1], Jinlong Li[2], Xiaoyu Dong[3], Hongkai Yu[2], Jiaqi Ma[1*]

Motivation

- Previous methods: strong assumption that all agents are equipped with **identical neural networks**, which is unrealistic
- Spatial resolution, channel number and patterns are different



(a) Multi-agent perception pipeline in the context of V2X perception



(b) PointPillar feature map (c) VoxelNet feature map

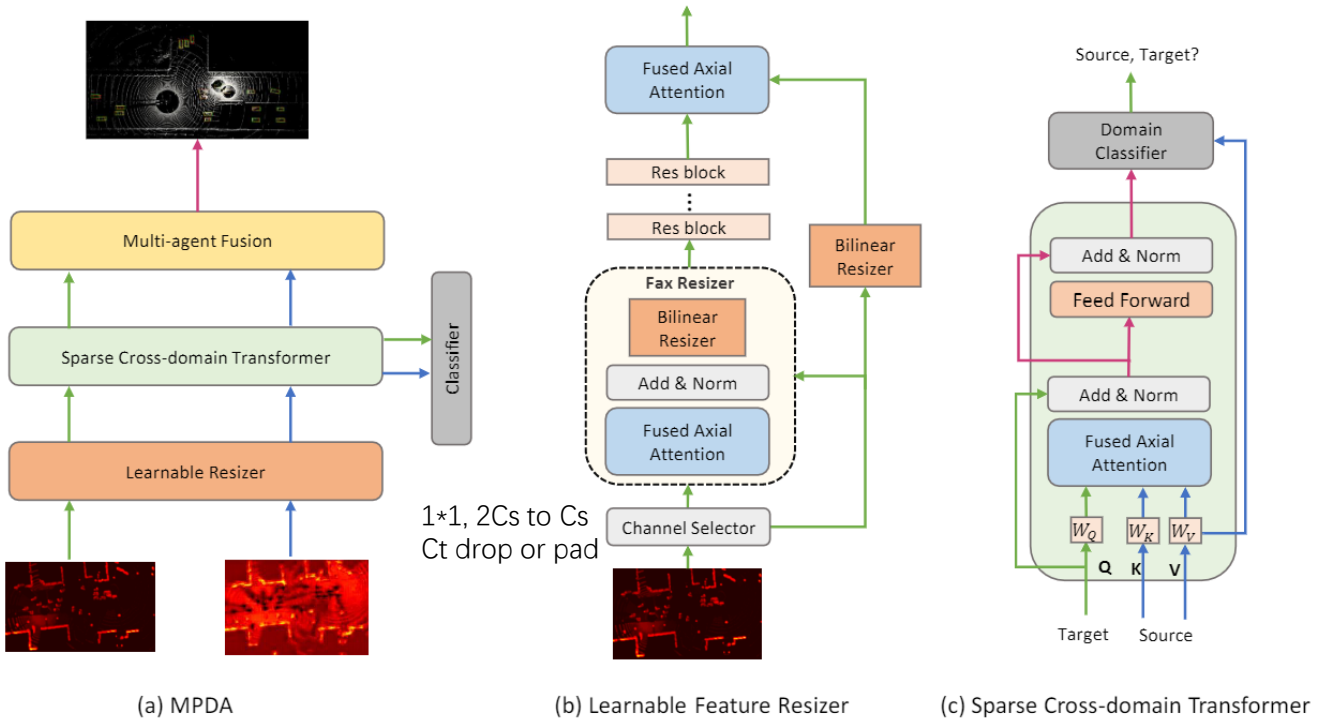
Fig. 1: **Illustration of domain gap of different feature maps for multi-agent perception.**. Here we use V2X cooperative perception in autonomous driving as an example. (a) Ego vehicle receives the shared feature maps from other CAV and infrastructure with different CNN models, which causes domain gaps. (b) Visualization of feature map from ego, which is extracted from PointPillar [10]. (c) Feature map from CAV, which is extracted from VoxelNet [11]. Brighter pixels represent higher feature values.

Contributions

- **The first work** to bridge the domain gap for multi-agent perception
- **Learnable Resizer** to better align spatial and channel features from other agents
- **Sparse cross-domain transformer** that can efficiently unify the feature patterns from various agents
- Can be easily combined with other multi-agent fusion algorithms and does not require confidential model information from other agents

Method

Assumptions: Accurate pose, no delay



(a) MPDA (b) Learnable Feature Resizer (c) Sparse Cross-domain Transformer

Fig. 2: **The overview and core components of our framework.** Our MPDA first aligns feature dimensions through a learnable feature resizer and then unifies the pattern through the sparse cross-domain transformer.

Experiments

TABLE III: **3D detection performance in Normal scenario (w/o domain gap) and Hetero scenarios (w/ domain gap).** We show the Average Precision (AP) at IoU=0.7. DC stands for domain classifier. * notes that we do not use the domain classifier when training on the normal scenario.

Method	Normal	Hetero 1	Hetero 2
No Fusion	40.2	40.2	40.2
Late Fusion	60.2	51.7	52.8
V2X-ViT	71.2	<u>26.7</u>	<u>34.5</u>
V2X-ViT (finetuned)	71.2	48.6	64.8
V2X-ViT + Resizer	72.3	54.8	72.1
V2X-ViT + MPDA (w/o DC)	73.4	56.3	72.5
V2X-ViT + MPDA	**73.4***	**57.6**	**73.3**