

# SF-UDA<sup>3D</sup>: Source-Free Unsupervised Domain Adaptation for LiDAR-Based 3D Object Detection

Cristiano Saltori  
University of Trento

cristiano.saltori@unitn.it

Nicu Sebe  
University of Trento  
Huawei Research

niculae.sebe@unitn.it

Elisa Ricci  
University of Trento  
Fondazione Bruno Kessler

eliricci@fbk.eu

Stéphane Lathuilière  
LTCl, Télécom Paris, Institut Polytechnique de Paris

stephane.lathuiliere@telecom-paris.fr

Fabio Galasso  
Sapienza University of Rome

galasso@di.uniroma1.it

## Abstract

3D object detectors based only on LiDAR point clouds hold the state-of-the-art on modern street-view benchmarks. However, LiDAR-based detectors poorly generalize across domains due to domain shift. In the case of LiDAR, in fact, domain shift is not only due to changes in the environment and in the object appearances, as for visual data from RGB cameras, but is also related to the geometry of the point clouds (e.g., point density variations). This paper proposes SF-UDA<sup>3D</sup>, the first Source-Free Unsupervised Domain Adaptation (SF-UDA) framework to domain-adapt the state-of-the-art PointRCNN 3D detector to target domains for which we have no annotations (unsupervised), neither we hold images nor annotations of the source domain (source-free). SF-UDA<sup>3D</sup> is novel on both aspects. Our approach is based on pseudo-annotations, reversible scale-transformations and motion coherency. SF-UDA<sup>3D</sup> outperforms both previous domain adaptation techniques based on features alignment and state-of-the-art 3D object detection methods which additionally use few-shot target annotations or target annotation statistics. This is demonstrated by extensive experiments on two large-scale datasets, i.e., KITTI and nuScenes.

## 1. Introduction

LiDAR is one of the key sensors for the longer-term autonomy of cars [15]. It is a native 3D sensor, which reads up to hundreds of meters, providing point clouds. Further to the proliferation of LiDAR companies, the efficacy of LiDAR is proven by the fact that LiDAR-only-based detectors are robust and accurate [49, 37, 47, 20] and provide state-of-the-art performance, currently held by PointRCNN [37].

LiDAR-based detectors, however, are prone to domain

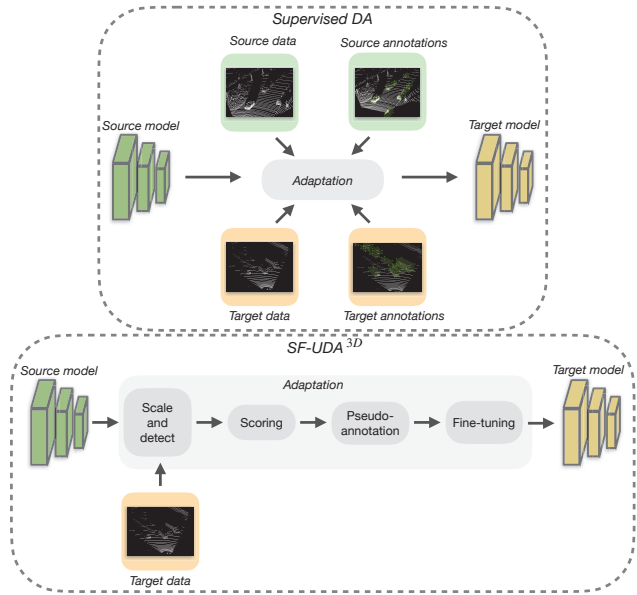


Figure 1: Existing supervised DA methods for LiDAR-based 3D detection [41] require both source and target data and annotations to adapt a pre-trained deep model to a target domain. Differently, leveraging on pseudo-annotations, reversible scale-transformations and motion coherency, SF-UDA<sup>3D</sup> adapts a pre-trained source network by using only unlabeled target data.

shift issues that may be more serious than for their RGB counterparts [41]. As in the RGB case, domain shift may be due to environmental changes (e.g. data collected in different cities and weather conditions) or to appearances variations of specific objects (e.g. car shapes and sizes may vary among different countries). Additionally, performance of LiDAR-based models significantly depends on the density of the LiDAR point cloud, spatial resolution and ranges.

To address this problem, in this paper we propose SF-UDA<sup>3D</sup>, the first Source-Free Unsupervised Domain Adaptation framework for LiDAR-based 3D detection. The proposed technique features the case where the 3D detector is applied *e.g.* in a different country, with differences in the local cars and roads, as well as to imagery acquired with different LiDAR sensors. Our approach is an *unsupervised* DA method because it does not require any annotation in the target domain and it is *source-free* because we assume that only the 3D detector trained on source data is available, while we do not have access to the source annotations and data (see Fig. 1). Both aspects are novel. To the best of our knowledge, in fact, despite its practical relevance, the problem of building LiDAR-based 3D detectors which are robust to domain shift has only recently been addressed in [41]. However, the method proposed in [41] assumes the availability of both source and target domain data and annotations. In this paper, we argue that this assumption is rarely satisfied in many real-world applications, where we may have only access to a pre-trained detector and it may be hard or even impossible to acquire target data annotations.

SF-UDA<sup>3D</sup> considers the PointRCNN [37] architecture and is based on the pseudo-annotation of the target unlabelled dataset by means of reversible scale-transformations and motion coherency. In details, SF-UDA<sup>3D</sup> annotates the unlabelled target data at multiple scales by a PointRCNN model pre-trained on the unavailable source dataset. Then, it assesses the quality of the annotations by scoring the resulting detection tracks by means of an unsupervised coherency metric (Mean Volume Variation). Subsequently, it reverses the scale-transformations, aggregates the best detection labels by confidence and finally fine-tunes PointRCNN. In spite of its simplicity, SF-UDA<sup>3D</sup> surpasses in performance state-of-the-art domain adaptation methods for 3D object detection which require target data annotations or annotation statistics [41]. Our algorithm also outperforms previous source-free general purpose unsupervised domain adaptation methods [22] by 30%. Overall, we fill in 66% of the gap between the source and target 3D detector when adapting from nuScenes to KITTI, and 30% in the much harder case of KITTI to nuScenes.

To summarize, our main contributions are as follows:

- We propose to study a novel problem, *i.e.* how to build LiDAR-based 3D detectors robust to domain shift when (i) we do not have access to source data and annotations and only a source pre-trained model is available, and (ii) no annotations are provided in the target domain, *i.e.* we are in an unsupervised domain adaptation setting;
- We propose SF-UDA<sup>3D</sup>, a novel approach for source-free unsupervised domain adaptation which empowers the state-of-the-art PointRCNN [37] architecture by means of pseudo-annotations, reversible scale-

transformations and motion coherency;

- We evaluate the proposed SF-UDA<sup>3D</sup> against relevant approaches and we show that our method outperforms both previous source-free feature-based domain adaptation methods [22] and, notably, state-of-the-art adaptation approaches for LiDAR-based 3D detection, although they additionally use few-shot target annotations or target annotation statistics [41].

## 2. Related Works

This work relates to LiDAR-based 3D detection, unsupervised 2D domain adaptation and 3D domain adaptation, which is currently at least weakly-supervised.

**LiDAR-based 3D Object Detection.** First successful 3D object detectors were two-stage detectors and employed RGB images to propose object bounding-boxes (with a region proposal network RPN), prior to a 3D classification branch (termed RCNN) [7, 27, 43, 47, 19]. More recently, better performance has been obtained by object proposal stages also based on 3D point clouds. This is the case of VoxelNet [51], SECOND [49] and PointPillar [20] which encoded the 3D point clouds into voxel or pillars and used by the RPN [51, 49] or by a 2D detections head [20] for generating object proposals. The state-of-the-art LiDAR-based 3D detector is PointRCNN [37], which avoids pseudo-images and the consequent quantization. It processes instead the point clouds in 3D, employing PointNet++ [28] 3D features both in the RPN and RCNN stages. Here we adopt PointRCNN both for its performance and robustness.

**Unsupervised Domain Adaptation (UDA) for 2D Object Detection.** There is a large body of literature on 2D UDA detection [38, 18, 40, 39, 12, 34, 36, 5], which may be divided into three types of approaches [21]. The first tackles the domain shift by enforcing domain confusion with adversarial training [8, 52, 33, 53], either with image-domain classifiers only [8] or by additional region-level alignments [52], possibly leveraging multiple scales and semantic information [53]. The second type of approaches exploits image-to-image translation between the source and target-domain data [2, 23, 24]. Some methods also add adversarial losses, which makes them hybrid between the two types [35, 18, 17, 30, 14]. A third type of methods, more closely related to our work, has considered self-training on target pseudo-labels [16, 4, 32]. In [16], noisy pseudo-annotations are obtained with the source model and refined by a joint source-target classifier; then, the target-adapted model is trained from scratch on the pseudo-labels. Differently, [4] adopts a teacher-student framework and similarly leverages the pseudo-labelling by a mean teacher, *i.e.* using the source supervised loss. In fact these techniques assume that the source dataset and labels are available during the domain adaptation phase, while our SF-UDA<sup>3D</sup> operates in a source-free setting. More recently, in the context

of 2D detection, [32] has adapted a source-trained model in a source-free manner by temporal consistency and knowledge distillation [13]. First, they labeled missed detections by tracking in the target dataset, then they distilled the soft-labels and re-train the detector. In this work, we also use temporal consistency, but we employ tracking to determine the scales which best transform the target point clouds for 3D pseudo-labelling.

**Domain Adaptation (DA) from 3D LiDAR data.** To the best of our knowledge, unsupervised domain adaptation from LiDAR data has been so far only researched for classification [29, 1] and segmentation tasks [45, 50]. In all cases, the algorithms assume that the source images and labels are available. This differs from SF-UDA<sup>3D</sup>, which is instead a source-free pipeline. There are currently two works of DA for 3D detection in LiDAR data. The first [42] aligns the global and local features by adversarial training, therefore [42] requiring supervision, *i.e.* annotation of the target data, and it is not source-free. The second work [41] evaluates both a few-shot DA, where a few annotations are given for the target dataset, and a weakly-supervised DA, where aggregated target statistics about the car scales are provided for adaptation via fine-tuning on the scale-transformed source annotations. Our method also leverages on scale-transformations but of the entire input point clouds and it is source-free and completely unsupervised, as it estimates the target scales via temporal coherency. Additionally, it aggregates pseudo-annotations from several estimated relevant scales, which makes a substantial difference in performance against the single output transformation or few-shot fine-tuning in [41], as we quantitatively evaluate in Sec. 4.

### 3. SF-UDA<sup>3D</sup>

In this section, we first present the problem statement and an overview of the proposed approach for domain adaptation, then we describe it in details in the subsections.

#### 3.1. Problem Statement

Given a 3D detection model  $\Phi_S$  trained on a source dataset  $\mathcal{X}_S = \{x_S^n\}_{1 \leq n \leq N}$  with source annotations  $\mathcal{Y}_S = \{y_S^n\}_{1 \leq n \leq N}$ , the goal of UDA in 3D detection is to obtain a target adapted 3D detector  $\Phi_T$  by exploiting a target dataset  $\mathcal{X}_T = \{x_T^m\}_{1 \leq m \leq M}$  without ground-truth annotations  $\mathcal{Y}_T = \{y_T^m\}_{1 \leq m \leq M}$ . In this work, we consider the challenging scenario of Source-Free Unsupervised Domain Adaptation (SF-UDA), where the source data  $\mathcal{X}_S$ , the source annotations  $\mathcal{Y}_S$  and the target annotations  $\mathcal{Y}_T$  are not available at adaption time, namely, when training the target model  $\Phi_T$ .

When the LiDAR sensors differ across source and target domains, the geometries of the point clouds are different. Assume that the source and target point clouds are sampled

by respective generating probability distributions  $P(\mathcal{X})$ , *i.e.*  $P(\mathcal{X}_S) \neq P(\mathcal{X}_T)$ . We illustrate in Sec. 4 that this is especially the case for point cloud densities in the nuScenes [3] and KITTI [11] datasets considered in this work. When the source and target datasets are acquired across different domains, *e.g.* countries, the ground-truth annotations also differ, *e.g.* since the shapes of cars are also different. Assuming that annotations are sampled from generating probability distributions, these would differ, *i.e.*  $P(\mathcal{Y}_S) \neq P(\mathcal{Y}_T)$ . This is the case of the nuScenes [3] and KITTI [11] datasets, acquired respectively in USA/Singapore and Germany. The same discrepancy should be reflected in the 3D detector output spaces, trained to mimic the ground-truth annotations.

In this work, we propose to align the annotation-scale distributions  $P(\mathcal{Y}_S)$  and  $P(\mathcal{Y}_T)$  by scale-transformation parameters, which we estimate by temporal coherency. Furthermore, we account for the misalignment of the point cloud distributions  $P(\mathcal{X}_S)$  and  $P(\mathcal{X}_T)$  by fine-tuning the source model  $\Phi_S$  on the pseudo-annotated target point cloud.

#### 3.2. Proposed Method

We propose a four-stage pipeline for adaptation, detailed in Secs. 3.2.1-3.2.3 and illustrated in Fig. 2. Firstly, we detect objects with the source model  $\Phi_S$  over a set of scaled target point clouds (Sec. 3.2.1). Secondly, we score the detections by a tracker for temporal consistency (Sec. 3.2.2). Thirdly, we aggregate detections at the best scales and lastly we fine-tune the source-model on the pseudo-annotated target point-cloud (Sec. 3.2.3).

##### 3.2.1 Scale and detect

In the *Scale-and-detect* stage, we consider a set of  $L$  scaling parameters  $\Omega = [\omega_1, \dots, \omega_L]$  where each  $\omega_l = (\omega_x, \omega_y, \omega_z) \in \mathbf{R}^{+3}$  parametrizes the scaling transformation along the 3D axes. To specify  $\Omega$ , we use a regular grid over the intervals centered around 1:  $[1 - \epsilon, 1 + \epsilon]^3$  with  $\epsilon > 0$ . For each  $\omega_l$ , we generate a transformed version of the dataset  $\mathcal{X}_T^{\omega_l}$  by re-scaling each sample in  $\mathcal{X}_T$ . Then, we employ the source object detector  $\Phi_S$  on every sample of  $\mathcal{X}_T^{\omega_l}$  obtaining detections  $\tilde{\mathcal{Y}}_T^{\omega_l}$ . Finally, to have detections  $\tilde{\mathcal{Y}}_T^{\omega_l}$  in the original target 3D space, we re-scale the detections by multiplying the position and dimension values by  $(1/\omega_x, 1/\omega_y, 1/\omega_z)$ . Besides, we note that this re-scaling step is required to obtain detections in the same 3D space and to allow a fair temporal consistency comparison. In all our experiments, we employ the PointRCNN detector [37] since it recently obtained the state-of-the-art performance in object detection benchmarks.

##### 3.2.2 Scale scoring with Temporal consistency

To identify the quality of the estimated detections  $\tilde{\mathcal{Y}}_T^{\omega_l}$ , we leverage the temporal consistency of detections between sequential frames (see Fig. 3). We propose to use a tracker

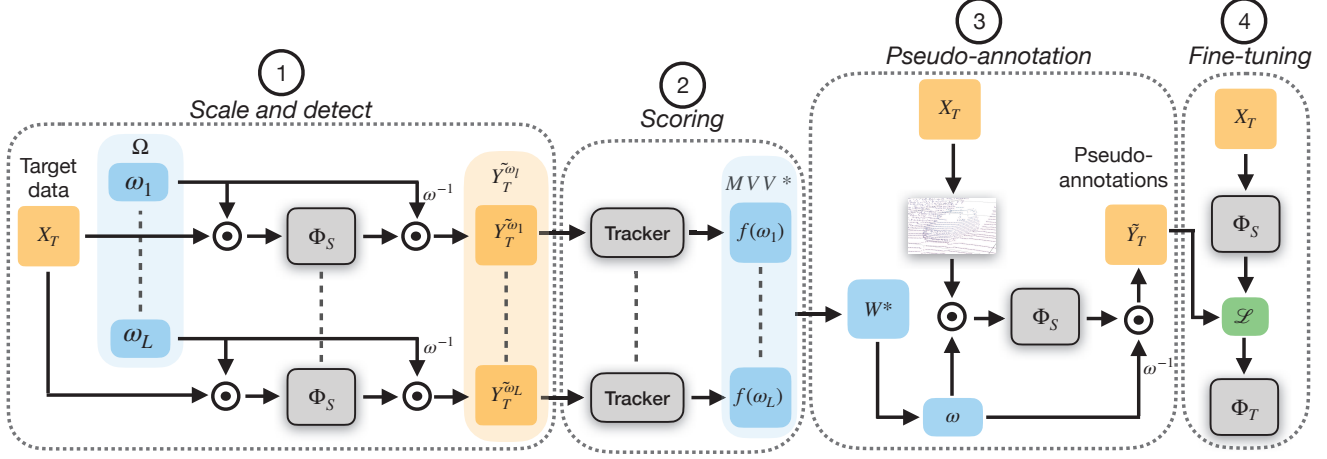


Figure 2: **Full SF-UDA<sup>3D</sup> pipeline overview.** Given a scaling solution space  $\Omega$ , in the first step detections over target sequences are obtained by scaling input data by  $\omega$  and by re-scaling predictions by  $1/\omega$ . Next, time consistency of each sequence is used through 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.

and to evaluate the stability of its prediction to *score the detection quality*. More specifically, we run a state-of-the-art tracking-by-detection pipeline [44]. For a given sequence  $V$ , we assume to obtain  $J$  tracks. Considering the tracked object with the index  $j < J$ , its track can be defined as lists of  $T_j$  consecutive bounding-boxes  $B_j = [b_{t_j}, \dots, b_{t_j+T_j}]$ , where  $t_j$  denotes the frame index where the object appears, and each  $b_t$  is the 3D bounding-box dimensions and locations at time  $t$ . Inspired by [46], we employ the Mean Volume Variation (MVV) between consecutive detections as scoring function:

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

where  $v_j^t$  is the bounding box volume of the  $j$ -th track at time  $t$  and  $\bar{v}_j$  is the mean volume of the bounding boxes in  $B_j$ . The intuition behind this scoring function is that, assuming an optimal detector and rigid objects, the bounding-box volume would be constant. Therefore, a good detector would predict detections with a stable volume leading to a small  $MVV$  value. Importantly, we observed that tracks that last less than 5 frames may be false-positives, and therefore, we propose to treat these tracks differently. More precisely, we introduce a penalty term  $H^*$  in the  $MVV$  score for every sequence without tracks longer than 5 frames. Our robust version of the  $MVV$  score can be written as:

$$MVV^*(V) = \begin{cases} MVV(V), & J \neq 0 \\ H^*, & \text{if no tracks} \end{cases} \quad (2)$$

Finally, as previously mentioned, we are interested in

scoring each scaling solution over all the target training sequences, therefore we consider as the scoring function  $f(\omega_i)$  for  $\omega_i$  the mean of  $MVV^*$  over all the target sequences.

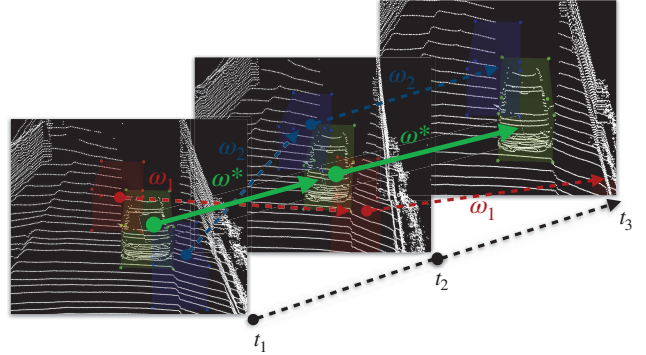


Figure 3: Given multiple possible scales  $\omega$ , SF-UDA<sup>3D</sup> selects the best  $\omega^*$  as the one generating the most time consistent detections.

### 3.2.3 Pseudo-annotation and Fine-tuning

Once each scaling parameters in  $\Omega$  is scored, we proceed with *Pseudo-annotation*. In a first *single-scale* (SS) approach, we only consider the scaling parameter corresponding to the lowest  $MVV^*$ , *i.e.* the best scale, referred to as  $\omega^*$  (see Fig. 3). In this approach, the detections  $\hat{Y}_{\mathcal{T}}^{\omega^*}$  at scale  $\omega^*$  from the first stage of our pipeline are considered as pseudo-annotations. We confirm in Sec. 4 that the identified best scale provides the best single-scale results. We also propose a *multi-scale* (MS) approach, which combines the top- $K$  best scaling parameters to improve the pseudo-annotations.



We name  $\Omega^*$  this set of best scales and we use it to determine a best scaling interval  $W^* = W_x^* \times W_y^* \times W_z^*$ , as follows:

$$\forall a \in [x, y, z], W_a^* = [\min_{\omega \in \Omega^*}(\omega_a), \max_{\omega \in \Omega^*}(\omega_a)] \quad (3)$$

Given  $W^*$ , for every frame of the target dataset, we sample a random  $\omega$  in  $W^*$ . The point cloud is scaled using  $\omega$ . Then, we use the source model  $\Phi_S$  to obtain 3D detections. Finally, we re-scale the predictions with  $1/\omega$  similarly to the *scale-and-detect* stage, to obtain bounding boxes in the original point-cloud space.

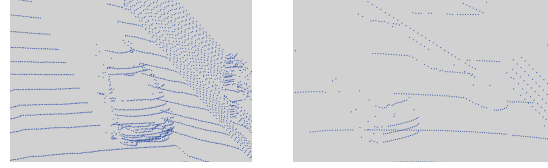
Sampling scales is repeated several times, the resulting 3D detections are collected and aggregated by non-maximal-suppression (NMS), yielding the final pseudo-labels  $\tilde{\mathcal{Y}}_{\mathcal{T}}$ . Sampling the scales multiple times is beneficial, possibly for two reasons: (i) cars within a point cloud occur at multiple scales and multiple sampling may find more of them; (ii) PointRCNN randomly sub-samples the input point cloud to obtain a constant number of input points and having multiple scale samples ensures more robustness against this randomness. Note that PointRCNN provides detections with confidence scores, which we find beneficial to threshold increasingly from low to high values. In other words, at early steps we use a low threshold to increase recall and include also low confidence detections. Later, we raise the threshold and only consider detections with a higher confidence. Finally, we merge pseudo-annotations from different steps through NMS and fine-tune the source model  $\Phi_S$  by using  $\tilde{\mathcal{Y}}_{\mathcal{T}}$  as annotations.

## 4. Experiments

In this section, we present the experimental evaluation of SF-UDA<sup>3D</sup> on two modern large-scale benchmarks of KITTI [11, 10] and nuScenes [3] against state-of-the-art methods, albeit none of them is source-free and unsupervised. Then, we conduct a thorough ablation study of the proposed framework. In the following, we introduce benchmarks and metrics.

**Datasets.** The KITTI object detection benchmark dataset [11, 10] has been acquired in Karlsruhe, Germany and is composed of 7481 training images, divided into 3712 training samples, 3769 validation samples and 7518 test images. The nuScenes dataset [3], acquired in Boston (USA) and Singapore, is  $\sim 2.3$  times larger, composed of 1000 driving sequences, for a total of 34149 images, divided into 28130 training samples and 6019 validation samples (which we treat as test samples), and it has both LiDAR scans and RGB images. The datasets differ under three main aspects:

- *Sensors.* Different sensors were used for data acquisition, as summarized in Table 1. These sensors sample points differently in terms of density (*i.e.* number of points), temporal frequency, spatial resolutions and ranges. Fig. 4 illustrate example differences, which affect the performances of the 3D object detectors.



(a) KITTI [10]

(b) nuScenes [3]

Figure 4: Example of cars from the KITTI and nuScenes datasets.

- *Environmental conditions.* Being acquired in diverse countries, the datasets depict objects of different shapes and sizes. Cars, which we target in this work, change much in these two aspects due to the differences between Germany and USA.
- *Dataset pre-processing.* The authors of the datasets made different choices for the data collection, annotation and filtering. For instance, in KITTI, only objects within 70m are annotated, while in nuScenes objects up to 100m have ground truth annotations. Another difference is the range where objects are annotated. Indeed, in KITTI, only objects visible in the front camera view are annotated, while in nuScenes also objects are annotated in the entire 360° surrounding space. In all our experiments, we consider the points which are visible from the frontal RGB-camera viewpoint (*e.g.* the CAM-FRONT in nuScenes and the rectified camera space for KITTI).

**Metrics.** For KITTI, we adopt the official metrics of [11, 10] and the classification into easy/moderate/hard detections, according to the visibility of the objects. In more details, we report the Average Precision (AP) over the 3D Intersection over the union (IoU) with a IoU threshold of 0.7. The final average (Avg) AP is obtained by averaging over the three difficulty categories.

For nuScenes we consider the official metrics [3]. In our experiments we consider the center-based definition of AP as used in the official nuScenes benchmark [3] and report both the AP at each of the four different distance thresholds of [0.5, 1.0, 2.0, 4.0] meters and the final average (Avg) over the four thresholds.

**Implementation details.** Our method is implemented in PyTorch, building upon the publicly-available PointRCNN 3D detector. More details can be found at the project website<sup>1</sup>. For training and evaluating our method, we run all our experiments on a DGX-1 server (8 Nvidia Tesla V-100 GPUs) and on a Lambda Blade server (8 NVidia Quadro RTX 6000 GPUs). The source training is performed with the ADAM optimization algorithm and one-cycle policy for 200 and 70 epochs for the RPN and RCNN respectively, with a batch size of 64 and 32 respectively, and with a maximum learning rate of 0.02 as in [37]. Similarly, during the

<sup>1</sup><https://github.com/saltoricristiano/SF-UDA-3DV>

Dataset	Samples	Max depth	Sensor	Channels	Resolution	Mean points	Classes
KITTI [11]	15k	70 m	HDL64E	64	$0.08^\circ \times 0.4^\circ$	16384	8
nuScenes [3]	34k	100 m	HDL32E	32	$0.08^\circ \times 1.33^\circ$	3808	23

Table 1: Datasets overview. Each dataset is acquired using sensors with different resolutions and numbers of channels. While nuScenes uses the original maximum depth, KITTI provides pre-filtered LiDAR data with a maximum depth of 70 m.

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 <sup>3D</sup> (SS)	0.589	0.414	0.388	0.464
SF-UDA <sup>3D</sup> (MS-3)	<b>0.688</b>	<b>0.498</b>	<b>0.450</b>	<b>0.545</b>
SF-UDA <sup>3D</sup> (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

target fine-tuning we keep the same training setup with the only difference that we train the RPN for 100 epochs with a maximum learning rate of 0.002. During the scale search, we use a grid size parameter  $\epsilon = 0.3$  and we employ a stride  $s = 0.075$  along each axis. We thus obtain a solution space of dimension of  $L = 125$ . As for the tracking-by-detection module, we use the publicly-available code of [44] and change the tracker hyper-parameters as follows: both the initialization and death thresholds are set to 2 matched detections and missed detections respectively. Finally, regarding the pseudo-annotation procedure, we annotate four times the target dataset for each confidence threshold in the list  $[0.05, 0.1, 0.2, 0.3]$  and use an NMS IoU threshold of 0.1 to merge pseudo-annotations.

#### 4.1. Comparison with the state-of-the-art

Our work is the first to tackle SF-UDA for 3D detection. For the sake of evaluation against state-of-the-art methods, we still compare with the following methods:

- *AdaBN* [22]: This is a state-of-the-art UDA method originally proposed for the classification tasks. We chose it because it is one of the few to operate in the source-free setting, differently from most previous approaches in UDA [9, 31, 6]. Here, we adapted AdaBN to the detection task by updating the batch normalization (BN) source-model mean and variance statistics of the PointRCNN features to the target validation set.
- *Few-Shot (FS) fine-tuning* [41]: This method proposes to adapt the detector to the target domain by fine-tuning on a set of 10 randomly sampled annotated target data. Following [41], we ran it 5 times and reported the average performance.
- *Output Transformation (OT)* [41]: This is a weakly-supervised DA technique. It uses average sizes of ob-

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 <sup>3D</sup> (SS)	0.136	0.260	<b>0.290</b>	<b>0.308</b>	0.249
SF-UDA <sup>3D</sup> (MS-3)	0.203	<b>0.266</b>	<b>0.290</b>	<b>0.308</b>	0.267
SF-UDA <sup>3D</sup> (MS-5)	<b>0.211</b>	0.264	0.288	0.307	<b>0.268</b>
Target	0.370	0.422	0.440	0.455	0.422

Table 3: Adaptation results: KITTI→nuScenes

jects in both the source and target domain to transform the predictions of the source model at test time over to the target data.

We compare SF-UDA<sup>3D</sup> in both the single-scale (SS) and top- $K$  multi-scale (MS- $K$ ) approaches, cf. Sec. 3.2.3. We consider two settings: (i) nuScenes as source and KITTI as target (nuScenes→KITTI) and (ii) KITTI as source and nuScenes as target (KITTI→nuScenes). In order to be comparable with the literature on 3D object detection, we performed the experiments in the nuScenes→KITTI setting by using the same split as in [37] and in the KITTI→nuScenes setting by using the official nuScenes splits [3]. Note that in [41] results are evaluated on different data splits which are not the official ones.

The results on the nuScenes→KITTI task are reported in Table 2; those on the KITTI→nuScenes are shown in Table 3. In both cases, SF-UDA<sup>3D</sup> outperforms both OS [41] and FS [41], although both of them use information from the target domain annotations. Analysing the performance of different variations of our method we observe that in the nuScenes→KITTI task (Table 2), the MS-3 version of SF-UDA<sup>3D</sup> is the best-performing method, gaining a +0.08 Avg-AP over SF-UDA<sup>3D</sup> (SS). Similarly, in the KITTI→nuScenes task (Table 3) SF-UDA<sup>3D</sup> (MS-5) leads to an improvement of +0.019 Avg-AP compared to SF-UDA<sup>3D</sup> (SS). These results confirm the effectiveness of our scale-based pseudo-annotation approach and the importance of the combination of the top- $K$  solutions in the annotation procedure. Notably, AdaBN [22] is not effective on either the adaptation tasks. This may indicate that geometry-based methods are better-suited than feature-based methods for 3D LiDAR-based detection adaptation and that features may need more sophisticated DA approaches. Additionally, to provide further insights on the

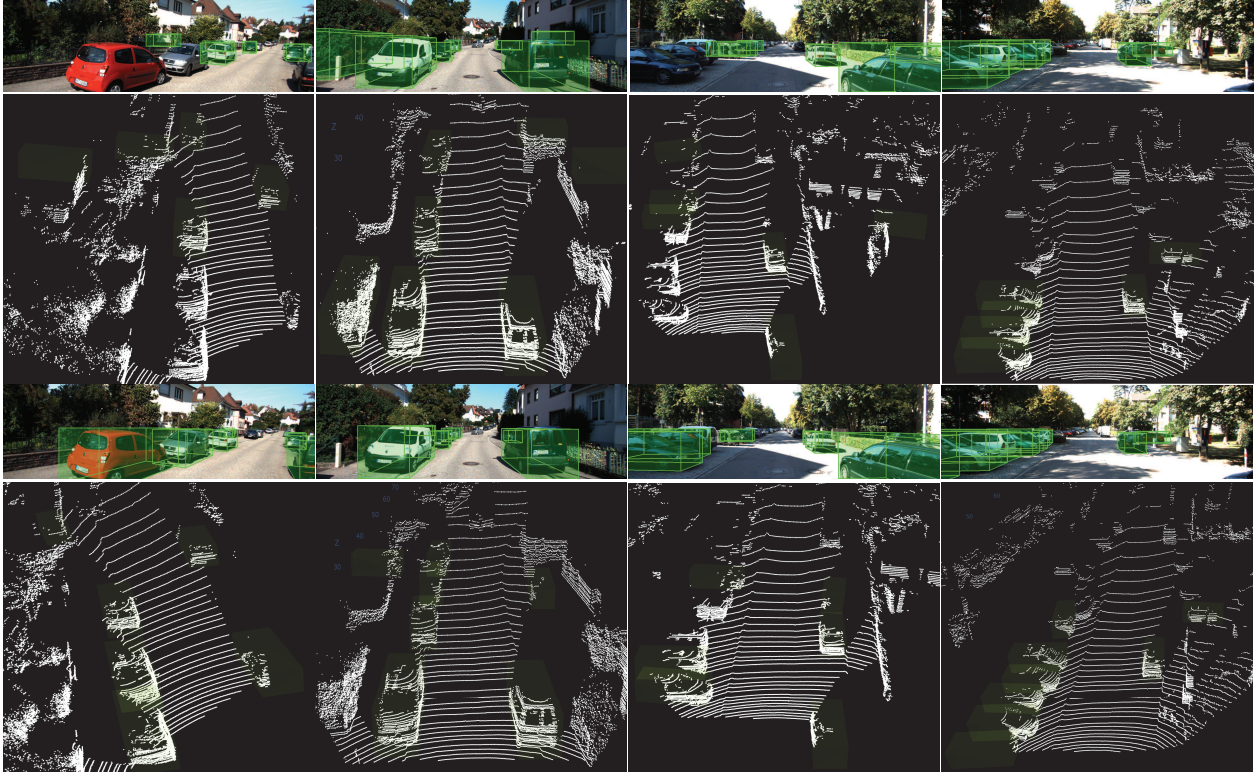


Figure 5: **Before (top) and after (bottom) adaptation on nuScenes→KITTI.** After adaptation with MS-3, performance improves and more objects are detected.

results of Tables 2 and 3, we also report and discuss the scaling parameters automatically selected by our method. Considering experiments on the nuScenes→KITTI task, SF-UDA<sup>3D</sup>(SS) estimates the parameters  $[1.30, 1.30, 1.15]$  along the axis X, Y and Z, respectively showing that KITTI data have to be upscaled along each axis to better match with nuScenes data. Similarly, SF-UDA<sup>3D</sup>(MS-3) provides an indication in the upscaling direction and selects the scale parameters within the intervals  $[1.15, 1.30]$  along X,  $[1.15, 1.30]$  along Y and  $[1.15, 1.30]$  along Z. Conversely, for the more challenging KITTI→nuScenes setting, SF-UDA<sup>3D</sup>(SS) computes the scaling parameters  $[0.85, 0.70, 1.00]$ , indicating that nuScenes objects point clouds should be downscaled along X and Y while keeping the original dimension along Z. Also the scale parameters adopted by the best performer SF-UDA<sup>3D</sup>(MS-5) indicate downsampling for adaptation, by the ranges  $[0.85, 1.15]$  along X,  $[0.70, 0.85]$  along Y and  $[0.85, 1.00]$  along Z. Here the model finds the wider range in X beneficial. Note that, in all of the above cases, the selected scales along the three axes are different. So further to scaling, SF-UDA<sup>3D</sup> implicitly learns to change the aspect ratios for domain adaptation.

#### 4.2. Ablation studies

Here we present the results of the ablation on each components of our method.

**Scaling parameters.** First we investigate the importance of the scaling parameters in the single-scale (SS) setup of SF-UDA<sup>3D</sup> during the pseudo-annotation phase. We consider the nuScenes→KITTI task and compare our results with three baselines:

- *No score*: we remove from SF-UDA<sup>3D</sup> the temporal-coherence tracking-based scores. This results in the random sampling of  $\omega$  in the range  $[0.7, 1.3]$  along each axis during the pseudo-annotation phase and quantifies the mere scale-augmentation.
- *No scale*: we remove from SF-UDA<sup>3D</sup> the scale-transformations altogether. So the target dataset is simply pseudo-annotated by the source model *as is*.
- *Supervised scale (Sup-scale)*: we make the single-scale SF-UDA<sup>3D</sup> weakly-supervised, by providing it with the ground-truth scale differences between the target and source average object sizes.

Table 4 reports the results of our evaluation, comparing different scaling methods in term of Avg-AP and reporting the selected scales. We observe from Table 4 that *No scale* is a poor adaptation strategy, probably because there is a significant average scale difference between KITTI and nuScenes. Among the three ablation variants, *No score* is the worst, a bit surprisingly. While data augmentation is mostly a positive tool, here it shows that it is im-



Method	Selected Scale			Avg-AP
	X	Y	Z	
No scale	1.00	1.00	1.00	0.306
No score	[0.70, 1.30]	[0.70, 1.30]	[0.70, 1.30]	0.223
SS	1.30	1.30	1.15	0.464
Sup-scale	1.18	1.09	1.16	<b>0.499</b>

Table 4: Ablation results: different scaling parameters.

portant to augment at the relevant scales, which we determine by tracking. *Sup-scale* respects the intuition and is slightly above our single-scale (SS) SF-UDA<sup>3D</sup>, since knowing the ground-truth is important. However the improvement is marginal, which confirms that the estimation scale-transformation parameters is effective.

**Scale selection metric.** In a second ablation experiment, we compare possible variants of unsupervised metrics to assess the quality of tracks (and thus to identify the most suitable scale transformations) on both the considered adaptation settings. In more details, we take inspiration from the work of [46] on benchmarking supervoxels and compare the applicable unsupervised metrics, namely the Time-Extension (TEX) and the Mean Volume Variation (MVV) – note that [46] introduces MSV, applicable to images, which we generalize to the point cloud volumes with MVV. TEX measures the temporal extent of tracking across frames, since intuitively a more stable tracking algorithm generates longer tracks. For MVV we compare the bare metric, as well as our proposed extension MVV\* with the penalty below a minimum-length track.

Metric	Avg-AP	
	nuScenes→KITTI	KITTI→nuScenes
TEX [46]	0.030	0.229
MVV [46]	<b>0.488</b>	0.125
MVV*	0.464	<b>0.249</b>

Table 5: Ablation results: different scoring metrics.

From Table 5, we see that TEX is not a suitable metric, though. In the realm of point clouds, longer tracks may correspond to wrong matches over time. In the Table, it shows that MVV is slightly better for the adaptation nuScenes→KITTI, but MVV\* greatly outperforms it for the adaptation KITTI→nuScenes. So the penalty terms plays an important role in the scoring over the noisier and more difficult nuScenes dataset.

**Assessing pseudo-annotations by scaling just.** In this final ablation study, we target to measure the quality of the pseudo-labels due to scaling transformations and to investigate why the combination of multiple scales is superior to the single-scale (e.g. MS-3 Vs. SS) pseudo-labelling. To this goal, we focus on estimating the best scales by temporal coherency. Then we consider each of the three best (1st, 2nd

Scale	Selected Scale			Avg-AP
	X	Y	Z	
No scale	1.00	1.00	1.00	0.202
1 <sup>st</sup> best	1.30	1.30	1.15	0.370
2 <sup>nd</sup> best	1.15	1.30	1.30	0.394
3 <sup>rd</sup> best	1.30	1.15	1.15	0.369
Sup-scale	1.18	1.09	1.16	<b>0.435</b>

Table 6: nuScenes→KITTI setting: quality of pseudo-annotations by using the best 3 scored scaling parameters. The results differ from Table 4 since here we measure the quality of pseudo-annotations before the fine-tuning step.

and 3rd best) and re-scale the target point-cloud according to it. Finally, we detect with the original source model  $\Phi_S$ , without fine-tuning. Compared to the full pipeline of Fig. 2, this study excludes step 4 of the pipeline. The results are reported in Table 6. Re-scaling the point cloud according to each of the three best scales improves the performance of the source model considerably. In fact, the source model passes from a performance of 0.202 Avg-AP on the non-rescaled point-cloud (*No scale*) to 0.370 Avg-AP in the case of the 1st best, which is 83% better. Also, re-scaling according to the ground-truth annotation statistics (*Sup-scale* entry in the Table) is understandably better. Finally, note that the three best scales resize the point-cloud in the same direction, by approximately similar upscaling factors, but with different aspect ratios. This may account for the better performance of MS-3 Vs. SS.

## 5. Conclusions

We have proposed SF-UDA<sup>3D</sup>, the first Source-Free Unsupervised Domain Adaptation approach for 3D LiDAR-based detection. Leveraging on pseudo-annotations, motion coherence and reversible scale-transformations, our method is capable to adapt a LiDAR-based detector trained on source data to a new target domain where only unlabelled pointclouds are available. Domain adaptation experiments conducted on the nuScenes→KITTI and KITTI→nuScenes tasks have shown that SF-UDA<sup>3D</sup> outperforms state of the art weakly supervised and few-shot supervised methods by a large margin. In future work, we plan to extend the method beyond cars, to the other object classes, to investigate the use of depth maps for adaptation [25, 26, 48] and to continue the investigation of unsupervised metrics for the scoring procedure.

## Acknowledgment

The work was partially supported by OSRAM GmbH and was carried out in the Vision and Learning joint laboratory of FBK and UNITN. We thank the CARITRO Deep Learning laboratory of ProM Facility for the granted GPU time.



## References

- [1] I. Achituve, H. Maron, and G. Chechik. Self-supervised learning for domain adaptation on point-clouds. *arXiv preprint arXiv:2003.12641*, 2020. [3](#)
- [2] V. F. Arruda, T. M. Paixão, R. F. Berriel, A. F. De Souza, C. Badue, N. Sebe, and T. Oliveira-Santos. Cross-domain car detection using unsupervised image-to-image translation: From day to night. In *IJCNN*, pages 1–8. IEEE, 2019. [2](#)
- [3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. [3](#), [5](#), [6](#)
- [4] Q. Cai, Y. Pan, C.-W. Ngo, X. Tian, L. Duan, and T. Yao. Exploring object relation in mean teacher for cross-domain detection. In *CVPR*, pages 11457–11466, 2019. [2](#)
- [5] Y. Cao, D. Guan, W. Huang, J. Yang, Y. Cao, and Y. Qiao. Pedestrian detection with unsupervised multispectral feature learning using deep neural networks. *Information Fusion*, 46:206–217, 2019. [2](#)
- [6] F. M. Cariucci, L. Porzi, B. Caputo, E. Ricci, and S. R. Bulò. Autodial: Automatic domain alignment layers. In *ICCV*, pages 5077–5085. IEEE, 2017. [6](#)
- [7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *CVPR*, pages 1907–1915, 2017. [2](#)
- [8] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool. Domain adaptive faster-rcnn for object detection in the wild. In *CVPR*, pages 3339–3348, 2018. [2](#)
- [9] G. Csurka. *Domain adaptation in computer vision applications*, volume 8. Springer, 2017. [6](#)
- [10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *IJRR*, 32(11):1231–1237, 2013. [5](#)
- [11] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361. IEEE, 2012. [3](#), [5](#), [6](#)
- [12] Z. He and L. Zhang. Multi-adversarial faster-rcnn for unrestricted object detection. In *ICCV*, pages 6668–6677, 2019. [2](#)
- [13] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. [3](#)
- [14] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang. Progressive domain adaptation for object detection. In *WACV*, pages 749–757, 2020. [2](#)
- [15] J. Janai, F. Guney, A. Behl, and A. Geiger. Computer vision for autonomous vehicles problems datasets and state of the art, 2019. [1](#)
- [16] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. G. Macready. A robust learning approach to domain adaptive object detection. In *ICCV*, pages 480–490, 2019. [2](#)
- [17] S. Kim, J. Choi, T. Kim, and C. Kim. Self-training and adversarial background regularization for unsupervised domain adaptive one-stage object detection. In *ICCV*, pages 6092–6101, 2019. [2](#)
- [18] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim. Diversify and match: A domain adaptive representation learning paradigm for object detection. In *CVPR*, pages 12456–12465, 2019. [2](#)
- [19] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, pages 1–8. IEEE, 2018. [2](#)
- [20] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, pages 12697–12705, 2019. [1](#), [2](#)
- [21] W. Li, F. Li, Y. Luo, and P. Wang. Deep domain adaptive object detection: a survey, 2020. [2](#)
- [22] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. Revisiting batch normalization for practical domain adaptation. *arXiv preprint arXiv:1603.04779*, 2016. [2](#), [6](#)
- [23] C.-T. Lin. Cross domain adaptation for on-road object detection using multimodal structure-consistent image-to-image translation. In *ICIP*, pages 3029–3030. IEEE, 2019. [2](#)
- [24] S. Liu, V. John, E. Blasch, Z. Liu, and Y. Huang. Ir2vi: Enhanced night environmental perception by unsupervised thermal image translation. In *CVPR-W*, pages 1153–1160, 2018. [2](#)
- [25] A. Pilzer, S. Lathuiliere, N. Sebe, and E. Ricci. Refine and distill: Exploiting cycle-inconsistency and knowledge distillation for unsupervised monocular depth estimation. In *CVPR*, pages 9768–9777, 2019. [8](#)
- [26] A. Pilzer, S. Lathuiliere, D. Xu, M. Puscas, E. Ricci, and N. Sebe. Progressive fusion for unsupervised binocular depth estimation using cycled networks. *T-PAMI*, 42(10):2380–2395, 2020. [8](#)
- [27] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *CVPR*, pages 918–927, 2018. [2](#)
- [28] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, pages 5099–5108, 2017. [2](#)
- [29] C. Qin, H. You, L. Wang, C.-C. J. Kuo, and Y. Fu. Pointdan: A multi-scale 3d domain adaption network for point cloud representation. In *NeurIPS*, pages 7190–7201, 2019. [3](#)
- [30] A. L. Rodriguez and K. Mikolajczyk. Domain adaptation for object detection via style consistency. *arXiv preprint arXiv:1911.10033*, 2019. [2](#)
- [31] S. Roy, A. Siarohin, E. Sangineto, S. R. Bulò, N. Sebe, and E. Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *CVPR*, 2019. [6](#)
- [32] A. RoyChowdhury, P. Chakrabarty, A. Singh, S. Jin, H. Jiang, L. Cao, and E. Learned-Miller. Automatic adaptation of object detectors to new domains using self-training. In *CVPR*, pages 780–790, 2019. [2](#), [3](#)
- [33] K. Saito, Y. Ushiku, T. Harada, and K. Saenko. Strong-weak distribution alignment for adaptive object detection. In *CVPR*, pages 6956–6965, 2019. [2](#)
- [34] K. Saleh, A. Abobakr, M. Attia, J. Iskander, D. Nahavandi, M. Hossny, and S. Nahvandi. Domain adaptation for vehicle detection from bird’s eye view lidar point cloud data. In *ICCV-W*, pages 0–0, 2019. [2](#)

- [35] Y. Shan, W. F. Lu, and C. M. Chew. Pixel and feature level based domain adaptation for object detection in autonomous driving. *Neurocomputing*, 367:31–38, 2019. 2
- [36] Z. Shen, H. Maheshwari, W. Yao, and M. Savvides. Scl: Towards accurate domain adaptive object detection via gradient detach based stacked complementary losses. *arXiv preprint arXiv:1911.02559*, 2019. 2
- [37] S. Shi, X. Wang, and H. Li. Pointtrcnn: 3d object proposal generation and detection from point cloud. In *CVPR*, pages 770–779, 2019. 1, 2, 3, 5, 6
- [38] E. Tzeng, K. Burns, K. Saenko, and T. Darrell. Splat: semantic pixel-level adaptation transforms for detection. *arXiv preprint arXiv:1812.00929*, 2018. 2
- [39] T. Wang, X. Zhang, L. Yuan, and J. Feng. Few-shot adaptive faster r-cnn. In *CVPR*, pages 7173–7182, 2019. 2
- [40] X. Wang, Z. Cai, D. Gao, and N. Vasconcelos. Towards universal object detection by domain attention. In *CVPR*, pages 7289–7298, 2019. 2
- [41] Y. Wang, X. Chen, Y. You, L. Erran, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao. Train in germany, test in the usa: Making 3d object detectors generalize, 2020. 1, 2, 3, 6
- [42] Z. Wang, S. Ding, Y. Li, M. Zhao, S. Roychowdhury, A. Wallin, G. Sapiro, and Q. Qiu. Range adaptation for 3d object detection in lidar. In *ICCV-W*, pages 0–0, 2019. 3
- [43] Z. Wang and K. Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *arXiv preprint arXiv:1903.01864*, 2019. 2
- [44] X. Weng and K. Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019. 4, 6
- [45] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer. Squeeze-seg2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. In *ICRA*, pages 4376–4382. IEEE, 2019. 3
- [46] C. Xu and J. J. Corso. Libsvx: A supervoxel library and benchmark for early video processing. *IJCV*, 119(3):272–290, 2016. 4, 8
- [47] D. Xu, D. Anguelov, and A. Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *CVPR*, pages 244–253, 2018. 1, 2
- [48] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *T-PAMI*, 41(6):1426–1440, 2019. 8
- [49] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2
- [50] S. Zhao, B. Li, X. Yue, Y. Gu, P. Xu, R. Hu, H. Chai, and K. Keutzer. Multi-source domain adaptation for semantic segmentation. In *NeurIPS*, pages 7285–7298, 2019. 3
- [51] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, pages 4490–4499, 2018. 2
- [52] X. Zhu, J. Pang, C. Yang, J. Shi, and D. Lin. Adapting object detectors via selective cross-domain alignment. In *CVPR*, pages 687–696, 2019. 2
- [53] C. Zhuang, X. Han, W. Huang, and M. R. Scott. ifan: Image-instance full alignment networks for adaptive object detection. *arXiv preprint arXiv:2003.04132*, 2020. 2