Unsupervised Domain Adaptation for Monocular 3D Object Detection via Self-Training

Zhenyu Li[1], Zehui Chen[2], Ang Li[3], Liangji Fang[3], Qinhong Jiang[3], Xianming Liu[1], and Junjun Jiang[1⋆]

[1] Harbin Institute of Technology
[2] University of Science and Technology
[3] SenseTime Research
{zhenyuli17, csxm, jiangjunjun}@hit.edu.cn lovesnow@mail.ustc.edu.cn
{liang1, fangliangji, jiangqinhong}@senseauto.com

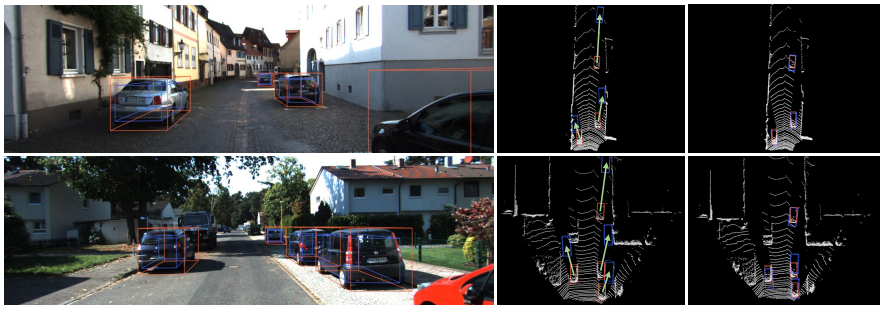
Abstract. Monocular 3D object detection (Mono3D) has achieved unprecedented success with the advent of deep learning techniques and emerging large-scale autonomous driving datasets. However, drastic performance degradation remains an unwell-studied challenge for practical cross-domain deployment as the lack of labels on the target domain. In this paper, we first comprehensively investigate the significant underlying factor of the domain gap in Mono3D, where the critical observation is a depth-shift issue caused by the geometric misalignment of domains. Then, we propose *STMono3D*, a new self-teaching framework for unsupervised domain adaptation on Mono3D. To mitigate the depth-shift, we introduce the *geometry-aligned multi-scale* training strategy to disentangle the camera parameters and guarantee the geometry consistency of domains. Based on this, we develop a teacher-student paradigm to generate adaptive pseudo labels on the target domain. Benefiting from the end-to-end framework that provides richer information of the pseudo labels, we propose the *quality-aware supervision* strategy to take instance-level pseudo confidences into account and improve the effectiveness of the target-domain training process. Moreover, the *positive focusing training* strategy and *dynamic threshold* are proposed to handle tremendous FN and FP pseudo samples. STMono3D achieves remarkable performance on all evaluated datasets and even surpasses fully supervised results on the KITTI 3D object detection dataset. To the best of our knowledge, this is the first study to explore effective UDA methods for Mono3D.

Keywords: Monocular 3D Object Detection, Domain Adaptation, Unsupervised Method, Self-Training

1 Introduction

Monocular 3D object detection (Mono3D) aims to categorize and localize objects from single input RGB images. With the prevalent development of cameras

⋆ Corresponding author (jiangjunjun@hit.edu.cn).



(a) Camera View (b) BEV View (c) STMono3D

Fig. 1. Depth-shift Illustration. When inferring on the target domain, models can accurately locate the objects on the 2D image but predict totally wrong object depth with tremendous shifts. Such unreliable predictions for pseudo labels cannot improve but hurt the model performance in STMono3D. GAMS guarantees the geometry consistency and enables models predict correct object depth. Best view in color: prediction and ground truth are in orange and blue. Depth-shift is shown in green arrows.

for autonomous vehicles and mobile robots, this field has drawn increasing research attention. Recently, it has obtained remarkable advancements [7, 2, 44, 40, 39, 31, 32] driven by deep neural networks and large-scale human-annotated autonomous driving datasets [16, 3, 20].

However, 3D detectors trained on one specific dataset (*i.e.* source domain) might suffer from tremendous performance degradation when generalizing to another dataset (*i.e.* target domains) due to unavoidable domain-gaps arising from different types of sensors, weather conditions, and geographical locations. Especially, as shown in Fig. 1, the severe depth-shift caused by different imaging camera devices leads to totally failed locations. Hence, a monocular 3D detector trained on data collected in Singapore cities with nuScenes [3] cameras **cannot** work well (*i.e.,* average precision drops to zero) when evaluated on data from European cities captured by KITTI [16] cameras. While collecting and training with more data from different domains could alleviate this problem, it is unfortunately infeasible, given diverse real-world scenarios and expensive annotation costs. Therefore, methods for effectively adapting a monocular 3D detector trained on a labeled source domain to a novel unlabeled target domain are highly demanded in practical applications. We call this task unsupervised domain adaptation (UDA) for monocular 3D object detection.

While intensive UDA studies [12, 26, 19, 9, 35, 15] on the 2D image setting are proposed, they mainly focus on handling lighting, color, and texture variations. However, in terms of the Mono3D, since detectors attend to estimate the spatial information of objects from monocular RGB images, the geometry alignment of domains is much more crucial. Moreover, for UDA on LiDAR-based 3D detection [47, 46, 27, 48], the fundamental differences in data structures and network architectures render these approaches not readily applicable to this problem.

In this paper, we propose *STMono3D*, for UDA on monocular 3D object detection. We first thoroughly investigate the depth-shift issue caused by the tight entanglement of models and camera parameters during the training stage. Models can accurately locate the objects on the 2D image but predict totally wrong object depth with tremendous shifts when inferring on the target domain. To alleviate this issue, we develop the *geometry-aligned multi-scale* (GAMS) training strategy to guarantee the geometry consistency of domains and predict pixel-size depth to overcome the inevitable misalignment and ambiguity. Hence, models can provide effective predictions on the unlabeled target domain. Based upon this, we adopt the mean teacher [37] paradigm to facilitate the learning. The teacher model is essentially a temporal ensemble of student models, where parameters are updated by an exponential moving average window on student models of preceding iterations. It produces stable supervision for the student model without prior knowledge of the target domain.

Moreover, we observe that the Mono3D teacher model suffers from extremely low confidence scores and numerous failed predictions on the target domain. To handle these issues, we adopt *Quality-Aware Supervision* (QAS), *Positive Focusing Training* (PFT), and *Dynamic Threshold* (DT) strategies. Benefitting from the flexibility of the end-to-end mean teacher framework, we utilize the readability of each teacher-generated prediction to dynamically reweight the supervision loss of the student model, which takes instance-level qualities of pseudo labels into account, avoiding the low-quality samples interfering the training process. Since the backgrounds of domains are similar in the Mono3D UDA of the autonomous driving setting, we ignore the negative samples and only utilize positive pseudo labels to train the model. It avoids excessive FN pseudo labels at the beginning of the training process impairing the capability of the model to recognize objects. In synchronization with training, we utilize a dynamic threshold to adjust the filter score, which stabilizes the increase of pseudo labels.

To the best of our knowledge, this is the first study to explore effective UDA methods for Mono3D. Experimental results on various datasets KITTI [16], nuScenes [3], and Lyft [20] demonstrate the effectiveness of our proposed methods, where the performance gaps between source only results and fully supervised oracle results are closed by a large margin. It is noteworthy that STMono3D even outperforms the oracle results under the nuScenes→KITTI setting. Codes will be released at https://github.com/zhyever/STMono3D.

2 Related Work

2.1 Monocular 3D Object Detection

Mono3D has drawn increasing attention in recent years [30, 33, 43, 2, 42, 40, 31, 39, 29, 24]. Earlier work utilizes sub-networks to assist 3D detection. For instance, 3DOP [8] and MLFusion [44] use depth estimators while Deep3DBox [30] adopts 2D object detectors. Another line of research makes efforts to convert the RGB input to 3D representations like OFTNet [33] and Pseudo-Lidar [43]. While these

methods have shown promising performance, they rely on the design and performance of sub-networks or dense depth labels. Recently, some methods propose to design the Mono3D framework in an end-to-end manner like 2D detection. M3D-RPN [2] implements a single-stage multi-class detector with a region proposal network and depth-aware convolution. SMOKE [25] proposes a simple framework to predict 3D objects without generating 2D proposals. Some methods [42, 10] develop a DETR-like [4] bbox head, where 3D objects are predicted by independent queries in a set-to-set manner. In this paper, we mainly conduct UDA experiments based on FCOS3D [40], a neat and representative Mono3D paradigm that keeps the well-developed designs for 2D feature extraction and is adapted for this 3D task with only basic designs for specific 3D detection targets.

2.2 Unsupervised Domain Adaptation

UDA aims to generalize the model trained on a source domain to unlabeled target domains. So far, tremendous methods have been proposed for various computer vision tasks [12, 26, 19, 9, 35, 15, 49] (e.g., recognition, detection, segmentation). Some methods [28, 36, 5] employ the statistic-based metrics to model the differences between two domains. Other approaches [34, 50, 21] utilize the self-training strategy to generate pseudo labels for unlabeled target domains. Moreover, inspired by Generative Adversarial Networks (GANs) [17], adversarial learning was employed to align feature distributions [38, 13, 14], which can be explained by minimizing the H-divergence [1] or the Jensen-Shannon divergence [18] between two domains. [23, 41] alleviated the domain shift on batch normalization layers by modulating the statistics in the BN layer before evaluation or specializing parameters of BN domain by domain. Most of these domain adaptation approaches are designed for the general 2D image recognition tasks, while direct adoption of these techniques for the large-scale monocular 3D object detection task may not work well due to the distinct characteristics of Mono3D, especially targes in 3D spatial coordination.

In terms of 3D object detection, [48, 47, 27] investigate UDA strategies for LIDAR-based detectors. SRDAN [48] adopt adversarial losses to align the features and instances with similar scales between two domains. ST3D [47] and MLC-Net [27] develop self-training strategies with delicate designs, such as random object scaling, triplet memory bank, and multi-level alignment, for domain adaptation. Following the successful trend of UDA on LIDAR-based 3D object detection, we investigate self-training strategies for Mono3D.

3 STMono3D

In this section, we first formulate the UDA task for Mono3D (Sec. 3.1), and present an overview of our framework (Sec. 3.2), followed by the self-teacher with temporal ensemble paradigm (Sec. 3.3). Then, we explain the details of the geometry-aligned multi-scale training (GAMS, Sec. 3.4), the quality-aware supervision (QAS, Sec. 3.5), and some other crucial training strategies consisting of positive focusing training (PFT) and dynamic threshold (DT) (Sec. 3.6).

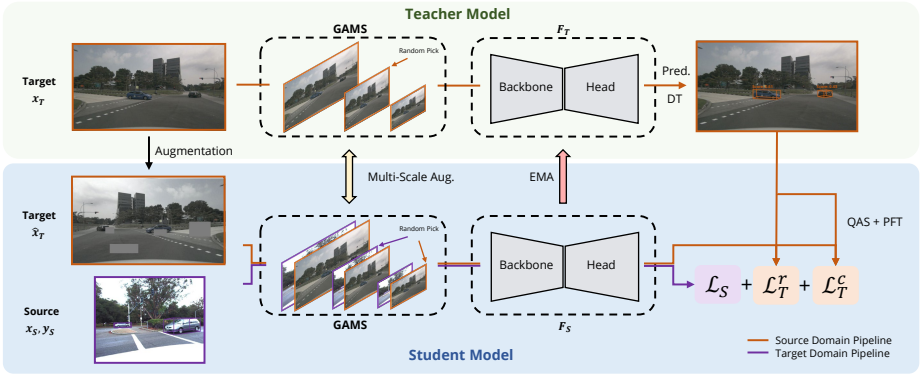


Fig. 2. Framework overview. STMono3D leverages the mean-teacher [37] paradigm where the teacher model is the exponential moving average of the student model and updated at each iteration. We design the GAMS (Sec. 3.4) to alleviate the severe depth-shift in cross domain inference and ensure the availability of pseudo labels predicted by the teacher model. QAS (Sec. 3.5) is a simple *soft-teacher* approach which leverages richer information from the teacher model to reweight losses and provide quality-aware supervision on the student model. PFT and DT are another two crucial training strategies presented in Sec. 3.6.

3.1 Problem Definition

Under the unsupervised domain adaptation setting, we access to labeled images from the source domain $\mathcal{D}_S = \{x_s^i, y_s^i, K_s^i\}_{i=1}^{N_S}$, and unlabeled images from the target domain $\mathcal{D}_T = \{x_t^i, K_t^i\}_{i=1}^{N_T}$, where N_s and N_t are the number of samples from the source and target domains, respectively. Each 2D image x^i is paired with a camera parameter K^i that projects points in 3D space to 2D image plane while y_s^i denotes the label of the corresponding training sample in the specific camera coordinate from the source domain. Label y is in the form of object class k, location (c_x, c_y, c_z), size in each dimension (d_x, d_y, d_z), and orientation θ. We aim to train models with $\{\mathcal{D}_S, \mathcal{D}_T\}$ and avoid performance degradation when evaluating on the target domain.

3.2 Framework Overview

We illustrate our STMono3D in Fig. 2. The labeled source domain data $\{x_S, y_S\}$ is utilized for supervised training of the student model F_S with a loss $\mathcal{L}_S$. In terms of the unlabeled target domain data x_T, we first perturb it by applying a strong random augmentation to obtain $\hat{x}_T$. Before passing to the models, both the target and source domain input are further augmented by the GAMS strategy in Sec. 3.4, where images and camera intrinsic parameters are cautiously aligned via simultaneously rescaling. Subsequently, the original and perturbed images are sent to the teacher and student model, respectively, where the teacher model

generates intuitively reasonable pseudo labels $\hat{y}_T$ and supervises the student model via loss $\mathcal{L}_T$ on the target domain:

$$\mathcal{L}_T = \mathcal{L}_T^r + \mathcal{L}_T^c, \tag{1}$$

where $\mathcal{L}_T^r$ and $\mathcal{L}_T^c$ are the regression loss and classification loss, respectively. Here, we adopt the QAS strategy in Sec. 3.5 to further leverage richer information from the teacher model by instance-wise reweighting the loss $\mathcal{L}_T$. In each iteration, the student model is updated through gradient descent with the total loss $\mathcal{L}$, which is a linear combination of $\mathcal{L}_S$ and $\mathcal{L}_T$:

$$\mathcal{L} = \lambda \mathcal{L}_S + \mathcal{L}_T, \tag{2}$$

where λ is the weight coefficient. Then, the teacher model parameters are updated by the corresponding parameters of the student model, where we introduce the details in Sec. 3.3. Moreover, we observe that the teacher model suffers from numerous FN and FP pseudo labels on the target domain. To handle this issue, we utilize the PFT and DT strategies illustrated in Sec. 3.6.

3.3 Self-Teacher with Temporal Ensemble

Following the successful trend of the mean teacher paradigm [37] in the semi-supervised learning, we adapt it to our Mono3D UDA task as illustrated in Fig. 2. The teacher model F_T and the student model F_S share the same network architecture but have different parameters θ_T and θ_S, respectively. During the training, the parameters of the teacher model are updated via taking the exponential moving average (EMA) of the student parameters:

$$\theta_T = m\theta_T + (1-m)\theta_S, \tag{3}$$

where m is the momentum that is commonly set close to 1, *e.g.,* 0.999 in our experiments. Moreover, the input of the student model is perturbed by a strong augmentation, which ensures that the pseudo labels generated by the teacher model are more accurate than the student model predictions, thus providing available optimization directions for the parameter updating. In addition, the strong augmentation can also improve the model generalization to handle the different domain inputs. Hence, by supervising the student model with pseudo targets $\hat{y}_T$ generated by the teacher model (*i.e.,* forcing the consistency between predictions of the student and the teacher model), the student can learn domain-invariant representations to adapt to the unlabeled target domain. Fig. 4 shows that the teacher model can provide effective supervision to the student model and Tab. 4, 5 demonstrate the effectiveness of the mean teacher paradigm.

3.4 Geometry-Aligned Multi-Scale Training

Observation. As shown in Fig. 1, depth-shift drastically harms the quality of pseudo labels on the target domain. It is mainly caused by the domain-specific

geometry correspondences between 3D objects and images (*i.e.,* camera imaging process). For instance, since the pixel size (defined in Eq. 6) of the KITTI dataset is larger than the nuScenes dataset, objects in images captured by KITTI cameras are smaller than nuScenes ones. While the model can predict accurate 2D locations on image planes, it tends to estimate relatively more distant object depth based on the cue that far objects tend to be smaller in perspective view. We call the phenomenon depth-shift: models localize accurate 2D location but predict depth with tremendous shifts on the target domain. To mitigate it, we propose a straightforward yet effective augmentation strategy, *i.e., geometry-aligned multi-scale* training, disentangling the camera parameters and detectors and ensuring the geometry consistency in the imaging process.

Method. Given the source input $\{x_S, y_S, K_S\}$ and the target input $\{x_T, K_T\}$, a naive geometry-aligned strategy is to rescale camera parameters to the same constant values and resize images correspondingly:

$$\mathbf{K} = \begin{bmatrix} r_x & r_y & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & p_x \\ 0 & f_y & p_y \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where r_x and r_y are resize rates, f and p are focal length and optical center, x and y indicate image coordinate axises, respectively. However, since the f/p cannot be changed by resizing, it is impracticable to strictly align the geometry correspondences of 3D objects and images between different domains via convenient transformations. The inevitable discrepancy and ambiguity lead to a failure on UDA.

To solve the issue, motivated by DD3D [31], we propose to predict the *pixel-size depth* d_p instead of the *metric depth* d_g:

$$d_p = \frac{s}{c} \cdot d_g, \tag{5}$$

$$s = \sqrt{\frac{1}{f_x^2} + \frac{1}{f_y^2}}, \tag{6}$$

where s and c are the pixel size and a constant, d_p is the model prediction and is scaled to the final result d_g. Therefore, while there are inevitable discrepancies between aligned geometry correspondences of two domains, the model can infer the depth from the pixel size and be more robust to the various imaging process. Moreover, we further rescale camera parameters into a multi-scale range, instead of the same constant values, and resize images correspondingly to enhance the dynamic of models. During the training, we keep ground-truth 3D bounding boxes y_S and pseudo labels $\hat{y}_T$ unchanged, avoiding changing real 3D scenes.

3.5 Quality-Aware Supervision

Observation. The cross-domain performance of the detector highly depends on the quality of pseudo labels. In practice, we have to utilize a higher threshold

on the foreground score to filter out most false positive (FP) box candidates with low confidence. However, unlike the teacher model that can detect objects with high confidence in the semi-supervised 2D detection or UDA of LiDAR-based 3D detector (*e.g.,* the threshold is set to 90% and 70% in [45] and [47], respectively), we find the Mono3D cross-domain teacher **suffers from a much lower confidence** as shown in Fig. 3, which is another unique phenomenon in Mono3D UDA caused by the much worse oracle monocular 3D detection performance than 2D detection and LiDAR-based 3D detection. It indicates that though the prediction confidence surpasses the threshold, we cannot ensure the sample quality, much less for the ones near the threshold. To alleviate the impact, we propose the *quality-aware supervision* (QAS) to leverage richer information from the teacher and take instance-level quality into account.

Method. Thanks to the flexibility of the end-to-end mean teacher framework, we assess the reliability of each teacher-generated bbox to be a real foreground, which is then used to weight the foreground classification loss of the student model. Given the foreground bounding box set $\{b_i^{fg}\}_{i=1}^{N^{fg}}$, the classification loss of the unlabeled images on the target domain is defined as:

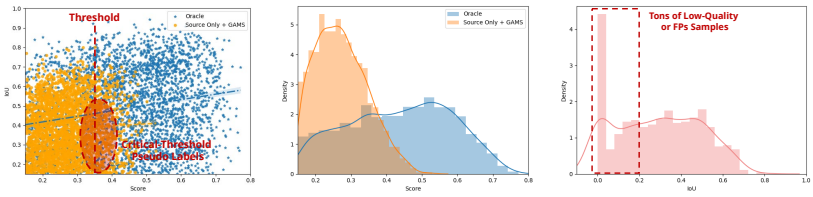
$$\mathcal{L}_T^c = \frac{\mu}{N^{fg}} \sum_{i=1}^{N^{fg}} w_i \cdot l_{cls}(b_i^{fg}, \mathcal{G}_{cls}), \tag{7}$$

where $\mathcal{G}_{cls}$ denotes the set of pseudo class labels, l_{cls} is the box classification loss, w_i is the confidence score for i^{th} foreground pseudo boxes, N^{fg} is the number of foreground pseudo box, and μ is a constant hyperparameter.

The QAS resembles a *simple positive mining* strategy, which is intuitively reasonable that there should be more severe punishment for pseudo labels with higher confidence. Moreover, compared with semi-supervised and supervised tasks that focus on simple/hard negative samples [45, 6], it is more critical for UDA Mono3D models to prevent harmful influence caused by low-quality pseudo labels near the threshold. Such an instance-level weighting strategy balances the loss terms based on foreground confidence scores and significantly improves the effectiveness of STMono3D.

3.6 Crucial Training Strategies

Positive Focusing Training. Since the whole STMono3D is trained in an end-to-end manner, the teacher model can hardly detect objects with confident scores higher than the threshold at the start of the training. Tons of FN pseudo samples impair the capability of the model to recognize objects. Because backgrounds of different domains are similar with negligible domain gaps in Mono3D UDA (*e.g.,* street, sky, and house), we propose the *positive focusing training* strategy. As for the $\mathcal{L}_T^c$, we discard negative background pseudo labels and only utilize the positive samples to supervise the student model, which ensures that the model does not crash to overfit on the FN pseudo labels during the training stage.



(a) (b) (c)

Fig. 3. (a) Correlation between confidence value and box IoU with ground-truth. (b) Distribution of confidence scores. The teacher suffers from low scores on the target domain. (c) Distribution of IoU between ground-truth and pseudo labels near the threshold (0.35-0.4). We highlight the existence of numerous low-quality and FP samples in these pseudo labels.

Dynamic Threshold. In practice, we find that the mean confidence score of pseudo labels gradually increases in synchronization within training duration. Increasing false positive (FP) samples appear in the middle and late stages of training, which harshly hurts the model performance. While the QAS strategy proposed in Sec. 3.5 can reduce the negative impact of low-quality pseudo labels, the completely wrong predictions still introduce inevitable noise to the training process. To alleviate the issue, we propose a simple *progressively increasing threshold* strategy to dynamic change the threshold τ as:

$$\tau = \begin{cases} \alpha, & iter < n_1, \\ \alpha + k \cdot (iter - n_1), & n_1 \leqslant iter < n_2, \\ \alpha + k \cdot (n_2 - n_1), & iter \geqslant n_2, \end{cases} \tag{8}$$

where α is the base threshold that is set to 0.35 based on the statistics in Fig. 3(a) in our experiments, k is the slope of increasing threshold, $iter$ is the iteration of training stage. The threshold is fixed to a minimum during the first n warmup steps as the teacher model can hardly detect objects with confident scores higher than the base threshold. It then linearly increases after the teacher model predicts pseudo labels with FP samples to avoid the model being blemished by increasing failure predictions. Finally, we find that the increasing of average scores tends to a saturation. Therefore, the threshold is fixed at the end of the training stage to guarantee the number of pseudo labels.

Table 1. Dataset Overview. We focus on their properties related to frontal-view cameras and 3D object detection. The dataset size refers to the number of images used in training stage. For Waymo and nuScenes, we subsample the data. See text for details.

Dataset	Size	Anno.	Loc.	Shape	FOV	Objects	Night
KITTI [16]	3712	17297	EUR.	(375,1242)	(29°,81°)	8	No
nuScenes [3]	27522	252427	SG.,EUR.	(900,1600)	(39°,65°)	23	Yes
Lyft [20]	21623	139793	SG.,EUR.	(1024,1224)	(60°,70°)	9	No

4 Experiments

4.1 Experimental Setup

Datasets. We conduct experiments on three widely used autonomous driving datasets: KITTI [16], nuScenes [3], and Lyft [20]. Two aspects are lying in our experiments: Cross domains with different cameras (existing in all the source-target pairs) and adaptation from label rich domains to insufficient domains (*i.e.,* nuScenes→KITTI). We summarize the dataset information in detail in Tab. 1, and present more visualization comparisons in the *supplementary material*.

Comparison Methods. In our experiments, we compare STMono3D with three methods: (*i*) **Source Only** indicates directly evaluating the source domain trained model on the target domain. (*ii*) **Oracle** indicates the fully supervised model trained on the target domain. (*iii*) **Naive ST (with GAMS)** is the basic self-training method. We first train a model (with GAMS) on the source domain, then generate pseudo labels for the target domain, and finally fine-tuning the trained model on the target domain.

Evaluation Metric. We adopt the KITTI evaluation metric for evaluating our methods in nuScenes→KITTI and Lyft→KITTI and the NuScenes metric for Lyft→nuScenes. We focus on the commonly used car category in our experiments. For Lyft→nuScenes, we evaluate models on ring view, which is more useful in real-world applications. For KITTI, We report the average precision (AP) where the IoU thresholds are 0.5 for both the bird's eye view (BEV) IoUs and 3D IoUs. For nuScenes, since the attribute labels are different from the source domain (*i.e.,* Lyft), we discard the average attribute error (mAAE) and report the average trans error (mATE), scale error (mASE), orient error (mAOE), and average precision (mAP). Following [47], we report the closed performance gap between Source Only to Oracle.

Implementation Details. We validate our proposed STMono3D on detection backbone FCOS3D [40]. Since there is no modification to the model, our method can be adapted to other Mono3D backbones as well. We implement STMono3D based on the popular 3D object detection codebase mmDetection3D [11]. We utilize SGD [22] optimizer. Gradient clip and warm-up policy are exploited with the learning rate 2×10^{-2}, the number of warm-up iterations 500, warm-up ratio 0.33, and batch size 32 on 8 Tesla V100s. The loss weight λ of different domains in Eq. 2 is set to 1. We apply a momentum m of 0.999 in Eq. 3 following most of mean teacher paradigms [27, 45]. As for the strong augmentation, we adopt the widely used image data augmentation, including random flipping, random erase, random toning, *etc.* We subsample $\frac{1}{4}$ dataset during the training stage of NuScenes and Lyft dataset for simplicity. Notably, unlike the mean teacher paradigm or the self-training strategy used in UDA of LiDAR-based 3D detector [27, 47], our STMono3D is trained in a *totally end-to-end* manner.

Table 2. Performance of STMono3D on three source-target pairs. We report AP of the car category at IoU = 0.5 as well as the domain gap closed by STMono3D. In nus→KITTI, STMono3D achieves a even better results on AP_{11} compared with the Oracle model, which demonstrates the effectiveness of our proposed method.

nus→K	AP_{11}						AP_{40}					
Method	AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5			AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Source Only	0	0	0	0	0	0	0	0	0	0	0	0
Oracle	33.46	23.62	22.18	29.01	19.88	17.17	33.70	23.22	20.68	28.33	18.97	16.57
STMono3D	35.63	27.37	23.95	28.65	21.89	19.55	31.85	22.82	19.30	24.00	16.85	13.66
Closed Gap	106.5%	115.8%	107.9%	98.7%	110.1%	113.8%	94.5%	98.2%	93.3%	84.7%	88.8%	82.4%

L→K	AP_{11}						**L→nus**	Metrics			
Method	AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5			Method	AP	ATE	ASE	AOE
	Easy	Mod.	Hard	Easy	Mod.	Hard					
Source Only	0	0	0	0	0	0	Source Only	2.40	1.302	0.190	0.802
Oracle	33.46	23.62	22.18	29.01	19.88	17.17	Oracle	28.2	0.798	0.160	0.209
STMono3D	26.46	20.71	17.66	18.14	13.32	11.83	STMono3D	21.3	0.911	0.170	0.355
Closed Gap	79.0%	87.6%	79.6%	62.5%	67.0%	68.8%	Closed Gap	73.2%	77.5%	66.7%	82.9%

Table 3. Ablation study of the geometry-aligned multi-scale training.

Nus→K	AP_{11}						AP_{40}					
GAMS	AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5			AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
	0	0	0	0	0	0	0	0	0	0	0	0
√	35.63	27.37	23.95	28.65	21.89	19.55	31.85	22.82	19.30	24.00	16.85	13.66

4.2 Main Results

As shown in Tab. 2, we compare the performance of our STMono3D with Source Only and Oracle. Our method outperforms the Source Only baseline on all evaluated UDA settings. Caused by the domain gap, the Source Only model cannot detect 3D objects where the mAP almost drops to 0%. Otherwise, STMono3D improves the performance on nuScenes→KITTI and Lyft→KITTI tasks by a large margin that around 110%/67% performance gap of AP_{3D} are closed. Notably, the AP_{BEV} and AP_{3D} of $AP_{11}, IoU \geqslant 0.5$ of STMono3D surpass the Oracle results, which indicates the effectiveness of our method. Furthermore, when transferring Lyft models to other domains that have full ring view annotations for evaluation (*i.e.*, Lyft→nuScenes), our STMono3D also attains a considerable performance gain which closes the Oracle and Source Only performance gap by up to 66% on AP_{3D}.

4.3 Ablation Studies and Analysis

In this section, we conduct extensive ablation experiments to investigate the individual components of our STMono3D. All experiments are conducted on the task of nuScenes→KITTI.

Effective of Geometry-Aligned Multi-Scale Training. We study the effects of GAMS in the mean teacher paradigm of STMono3D and the Naive ST

Table 4. Comparison of different self-training paradigms.

Nus→K	KITTI AP_{11}						Nus Metrics			
Method	AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5			AP	ATE	ASE	AOE
	Easy	Mod.	Hard	Easy	Mod.	Hard				
Naive ST	0	0	0	0	0	0	-	-	-	-
Naive ST with GAMS	9.05	9.08	8.82	3.72	3.69	3.58	14.0	0.906	0.164	0.264
STMono3D	35.63	27.37	23.95	28.65	21.89	19.55	36.5	0.731	0.160	0.167

Table 5. Ablation study of the exponential moving average strategy.

Nus→K	AP_{11}						AP_{40}					
EMA	AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5			AP_{BEV} IoU ⩾ 0.5			AP_{3D} IoU ⩾ 0.5		
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
	2.55	2.41	2.38	0.82	0.82	0.82	0.45	0.31	0.25	0.06	0.03	0.02
√	35.63	27.37	23.95	28.65	21.89	19.55	31.85	22.82	19.30	24.00	16.85	13.66

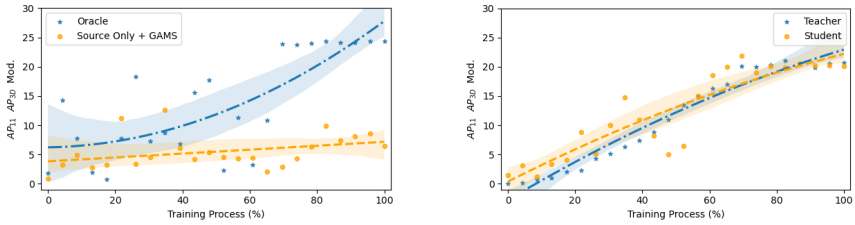
pipeline. Tab. 3 first reports the experimental results when GAMS is disabled. Caused by the depth-shift analyzed in Sec. 3.4, the teacher model generates incorrect pseudo labels on the target domain, thus leading to a severe drop in model performance. Furthermore, as shown in Tab. 4, GAMS is crucial for effective Naive ST as well. It is reasonable that GAMS supports the model trained on the source domain to generate valid pseudo labels on the target domain, making the fine-tuning stage helpful for the model performance. We present pseudo labels predicted by the teacher model of STMono3D in Fig. 1, which shows that the depth-shift is well alleviated. All the results highlight the importance of GAMS for effective Mono3D UDA.

Comparison of Self-Training Paradigm. We compare our STMono3D with other commonly used self-training paradigms (*i.e.*, Naive ST) in Tab. 4. While the GAMS helps the Naive ST teacher generate effective pseudo labels on the target domain to boost UDA performance, our STMono3D still outperforms it by a significant margin. One of the primary concerns lies in low-quality pseudo labels caused by the domain gap. Moreover, as shown in Fig. 4(a), while the performance of Oracle improves progressively, the Source Only model on the target domain suffers from a performance fluctuation. It is also troublesome to choose a specific and suitable model from immediate results to generate pseudo labels for the student model.

In terms of our STMono3D, the whole framework is trained in an end-to-end manner. The teacher is a temporal ensemble of student models at different time stamps. Fig. 4(b) shows that our teacher model is much more stable compared with the ones in Naive ST and has a better performance than the student model at the end of the training phase, where the teacher model starts to generate more predictions over the filtering score threshold. This validates our analysis in Sec. 3.3 that the mean teacher paradigm provides a more effective teacher model for pseudo label generation. Tab. 5 demonstrates the effectiveness of the EMA of STMono3D. The performance significantly degrades when the EMA is disabled, and the model is easily crashed during the training stage.

Table 6. Ablation study of QAS on different loss terms.

Nus→K		AP_{11}						AP_{40}					
L_T^{reg}	L_T^{cls}	AP_{BEV} IoU $\geqslant 0.5$			AP_{3D} IoU $\geqslant 0.5$			AP_{BEV} IoU $\geqslant 0.5$			AP_{3D} IoU $\geqslant 0.5$		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
		26.33	21.92	19.57	21.17	18.14	16.46	21.66	16.64	14.03	15.55	12.06	9.88
√		21.50	17.57	15.35	16.57	13.80	11.34	20.47	15.77	13.12	15.32	11.69	9.35
	√	35.63	27.37	23.95	28.65	21.89	19.55	31.85	22.82	19.30	24.00	16.85	13.66
√	√	21.74	19.56	17.22	18.09	15.67	14.71	16.01	13.26	11.15	10.89	9.22	7.49

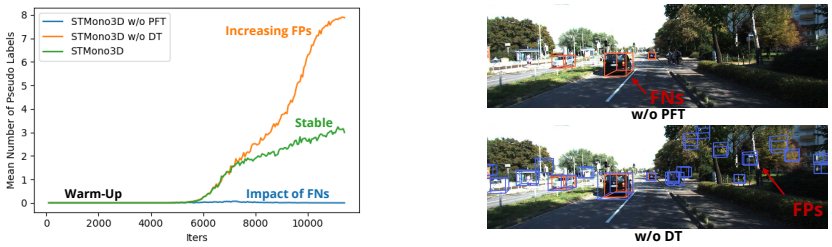


(a) Oracle *v.s.* Source Only + GAMS (b) STMono3D Teacher *v.s.* Student

Fig. 4. Performance comparision. (a) Oracle *v.s.* Source Only with GAMS: While the Oracle performance progressively improves, the Source Only model suffers from a drasical performance fluctuation. (b) Mean Teacher *v.s.* Student on the target domain: Not only does the teacher model outperforms the student at the end of the training phase, its performance curve is also smoother and more stable.

Effective of Quality-Aware Supervision. We study the effects of different applied loss terms of the proposed QAS strategy. Generally, the loss terms of Mono3D can be divided into two categories: (*i*) $\mathcal{L}_{cls}$ containing the object classification loss and attribute classification loss, and (*ii*) $\mathcal{L}_{reg}$ consisting of the location loss, dimension loss, and orientation loss. We separately apply the QAS on these two kinds of losses and report the corresponding results in Tab. 6. Interestingly, utilizing the confidence score from the teacher to reweight the $\mathcal{L}_{reg}$ cannot improve the model performance. We speculate it is caused by a loose correlation between the IoU score and localization quality (see yellow or blue line in Fig. 3(a)), which is in line with the findings in LiDAR-based method [47]. However, we find QAS is more applicable for the $\mathcal{L}_{cls}$, where the model performance increases about 20.6% AP_{3D}. It indicates the effectiveness of our proposed QAS strategy. It is intuitively reasonable since the score of pseudo labels itself is used to measure the confidence of predicted object classification.

Effective of Crucial Training Strategies. We then further investigate the effectiveness of our proposed PFT and DT strategies. We first present the ablation results in Tab. 7. When we disable the strategies, model performance suffers from drastic degradations, where AP_{3D} drops 64.3%. The results demonstrate they are crucial strategies in STMono3D. As shown in Fig. 5(a), we also present the influence of them in a more intuitive manner. If we disable the PFT, the model will be severely impaired by the numerous FN predcitions (shown in



(a) Num. of pseudo labels during training (b) Visualization examples

Fig. 5. Effects of the proposed DFT and DT. (a) Correlation between the average of the number of pseudo labels and training iters. (b) Examples of harmful FN and FP pseudo labels caused by disabling DFT and DT, respectively.

Table 7. Ablation study of PFT and DT.

Nus→K		AP_{11}						AP_{40}					
PFT	DT	AP_{BEV} IoU $\geqslant 0.5$			AP_{3D} IoU $\geqslant 0.5$			AP_{BEV} IoU $\geqslant 0.5$			AP_{3D} IoU $\geqslant 0.5$		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
		13.57	11.33	10.31	9.10	7.80	7.00	12.36	9.42	8.03	7.82	5.82	5.08
√		19.59	16.00	14.35	15.96	13.15	12.23	13.44	9.76	7.90	9.23	6.52	5.13
	√	18.90	16.57	15.75	15.15	13.73	12.85	12.74	10.35	9.42	8.41	6.81	5.96
√	√	35.63	27.37	23.95	28.65	21.89	19.55	31.85	22.82	19.30	24.00	16.85	13.66

Fig. 5(b) top) in the warm-up stage, leading to a failure to recognize objects in the following training iterations. On the other hand, for the teacher model w/o DT, the number of predictions abruptly increases at the end of training process, introducing more FPs predictions (shown in Fig. 5(b) down) that are harmful to the model perfomance.

5 Conclusion

In this paper, we have presented STMono3D, a meticulously designed unsupervised domain adaptation framework tailored for monocular 3D object detection task. We investigate that the depth-shift caused by the geometry discrepancy of domains leads to a drastic performance degradation when cross-domain inference. To alleviate the issue, we leverages a teacher-student paradigm for pseudo label generation and propose quality-aware supervision, positive focusing training and dynamic threshold to handle the difficulty in Mono3D UDA. Extensive experimental results demonstrate the effectiveness of STMono3D.

6 Acknowledgments

The research was supported by the National Natural Science Foundation of China (61971165, 61922027), and also is supported by the Fundamental Research Funds for the Central Universities.

References

1. Ben-David, S., Blitzer, J., Crammer, K., Kulesza, A., Pereira, F., Vaughan, J.W.: A theory of learning from different domains. Machine Learning **79**(1), 151–175 (2010)
2. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: International Conference on Computer Vision (ICCV). pp. 9287–9296 (2019)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Computer Vision and Pattern Recognition (CVPR). pp. 11621–11631 (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision (ECCV). pp. 213–229. Springer (2020)
5. Carlucci, F.M., Porzi, L., Caputo, B., Ricci, E., Bulo, S.R.: Autodial: Automatic domain alignment layers. In: International Conference on Computer Vision (ICCV). pp. 5077–5085. IEEE (2017)
6. Chen, X., Yuan, Y., Zeng, G., Wang, J.: Semi-supervised semantic segmentation with cross pseudo supervision. In: Computer Vision and Pattern Recognition (CVPR). pp. 2613–2622 (2021)
7. Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R.: Monocular 3d object detection for autonomous driving. In: Computer Vision and Pattern Recognition (CVPR). pp. 2147–2156 (2016)
8. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A.G., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. Advances in Neural Information Processing Systems (NIPS) **28** (2015)
9. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Computer Vision and Pattern Recognition (CVPR). pp. 3339–3348 (2018)
10. Chen, Z., Li, Z., Zhang, S., Fang, L., Jiang, Q., Zhao, F.: Graph-detr3d: Rethinking overlapping regions for multi-view 3d object detection. arXiv preprint arXiv:2204.11582 (2022)
11. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d (2020)
12. Dubourvieux, F., Audigier, R., Loesch, A., Ainouz, S., Canu, S.: Unsupervised domain adaptation for person re-identification through source-guided pseudo-labeling. In: International Conference on Pattern Recognition (ICPR). pp. 4957–4964 (2021)
13. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International Conference on Machine Learning (ICML). pp. 1180–1189. PMLR (2015)
14. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research (JMLR) **17**(1), 2096–2030 (2016)
15. Ge, Y., Zhu, F., Chen, D., Zhao, R., et al.: Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. Advances in Neural Information Processing Systems (NIPS) **33**, 11309–11321 (2020)

16. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Computer Vision and Pattern Recognition (CVPR). pp. 3354–3361 (2012)
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in Neural Information Processing Systems (NIPS) **27** (2014)
18. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in Neural Information Processing Systems (NIPS) **30** (2017)
19. Hoffman, J., Wang, D., Yu, F., Darrell, T.: Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649 (2016)
20. Kesten, R., Usman, M., Houston, J., Pandya, T., Nadhamuni, K., Ferreira, A., Yuan, M., Low, B., Jain, A., Ondruska, P., Omari, S., Shah, S., Kulkarni, A., Kazakova, A., Tao, C., Platinsky, L., Jiang, W., Shet, V.: Level 5 perception dataset 2020. https://level-5.global/level5/data/ (2019)
21. Khodabandeh, M., Vahdat, A., Ranjbar, M., Macready, W.G.: A robust learning approach to domain adaptive object detection. In: International Conference on Computer Vision (ICCV). pp. 480–490 (2019)
22. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
23. Li, Y., Wang, N., Shi, J., Hou, X., Liu, J.: Adaptive batch normalization for practical domain adaptation. Pattern Recognition (PR) **80**, 109–117 (2018)
24. Li, Z., Chen, Z., Li, A., Fang, L., Jiang, Q., Liu, X., Jiang, J., Zhou, B., Zhao, H.: Simipu: Simple 2d image and 3d point cloud unsupervised pre-training for spatial-aware visual representations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 1500–1508 (2022)
25. Liu, Z., Wu, Z., Tóth, R.: Smoke: Single-stage monocular 3d object detection via keypoint estimation. In: Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 996–997 (2020)
26. Long, M., Cao, Y., Wang, J., Jordan, M.: Learning transferable features with deep adaptation networks. In: International Conference on Machine Learning (ICML). pp. 97–105 (2015)
27. Luo, Z., Cai, Z., Zhou, C., Zhang, G., Zhao, H., Yi, S., Lu, S., Li, H., Zhang, S., Liu, Z.: Unsupervised domain adaptive 3d detection with multi-level consistency. In: International Conference on Computer Vision (ICCV). pp. 8866–8875 (2021)
28. Mancini, M., Porzi, L., Bulo, S.R., Caputo, B., Ricci, E.: Boosting domain adaptation by discovering latent domains. In: Computer Vision and Pattern Recognition (CVPR). pp. 3771–3780 (2018)
29. Mao, J., Shi, S., Wang, X., Li, H.: 3d object detection for autonomous driving: A review and new outlooks. arXiv preprint arXiv:2206.09474 (2022)
30. Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J.: 3d bounding box estimation using deep learning and geometry. In: Computer Vision and Pattern Recognition (CVPR). pp. 7074–7082 (2017)
31. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: International Conference on Computer Vision (ICCV). pp. 3142–3152 (2021)
32. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Computer Vision and Pattern Recognition (CVPR). pp. 8555–8564 (2021)
33. Roddick, T., Kendall, A., Cipolla, R.: Orthographic feature transform for monocular 3d object detection. arXiv preprint arXiv:1811.08188 (2018)

34. Saito, K., Ushiku, Y., Harada, T.: Asymmetric tri-training for unsupervised domain adaptation. In: International Conference on Machine Learning (ICML). pp. 2988–2997. PMLR (2017)
35. Saito, K., Ushiku, Y., Harada, T., Saenko, K.: Strong-weak distribution alignment for adaptive object detection. In: Computer Vision and Pattern Recognition (CVPR). pp. 6956–6965 (2019)
36. Sun, B., Saenko, K.: Deep coral: Correlation alignment for deep domain adaptation. In: European Conference on Computer Vision (ECCV). pp. 443–450. Springer (2016)
37. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. Advances in neural information processing systems (NIPS) **30** (2017)
38. Tzeng, E., Hoffman, J., Darrell, T., Saenko, K.: Simultaneous deep transfer across domains and tasks. In: International Conference on Computer Vision (ICCV). pp. 4068–4076 (2015)
39. Wang, T., Xinge, Z., Pang, J., Lin, D.: Probabilistic and geometric depth: Detecting objects in perspective. In: Conference on Robot Learning (CoRL). pp. 1475–1485 (2022)
40. Wang, T., Zhu, X., Pang, J., Lin, D.: Fcos3d: Fully convolutional one-stage monocular 3d object detection. In: International Conference on Computer Vision Workshop (ICCVW). pp. 913–922 (2021)
41. Wang, X., Jin, Y., Long, M., Wang, J., Jordan, M.I.: Transferable normalization: Towards improving transferability of deep neural networks. Advances in Neural Information Processing Systems (NIPS) **32** (2019)
42. Wang, Y., Guizilini, V.C., Zhang, T., Wang, Y., Zhao, H., Solomon, J.: Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning (CoRL). pp. 180–191 (2022)
43. Weng, X., Kitani, K.: Monocular 3d object detection with pseudo-lidar point cloud. In: International Conference on Computer Vision Workshops (ICCVW). pp. 0–0 (2019)
44. Xu, B., Chen, Z.: Multi-level fusion based 3d object detection from monocular images. In: Computer Vision and Pattern Recognition (CVPR). pp. 2345–2353 (2018)
45. Xu, M., Zhang, Z., Hu, H., Wang, J., Wang, L., Wei, F., Bai, X., Liu, Z.: End-to-end semi-supervised object detection with soft teacher. In: International Conference on Computer Vision (ICCV). pp. 3060–3069 (2021)
46. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. arXiv preprint arXiv:2108.06682 (2021)
47. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d: Self-training for unsupervised domain adaptation on 3d object detection. In: Computer Vision and Pattern Recognition (CVPR). pp. 10368–10378 (2021)
48. Zhang, W., Li, W., Xu, D.: Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In: Computer Vision and Pattern Recognition (CVPR). pp. 6769–6779 (2021)
49. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: European Conference on Computer Vision (ECCV). pp. 289–305 (2018)
50. Zou, Y., Yu, Z., Kumar, B., Wang, J.: Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In: European Conference on Computer Vision (ECCV). pp. 289–305 (2018)