Reg-TTA3D: Better Regression Makes Better Test-time Adaptive 3D Object Detection

Jiakang Yuan[1], Bo Zhang[2], Kaixiong Gong[3], Xiangyu Yue[3], Botian Shi[2], Yu Qiao[2], and Tao Chen[1,✉]

[1] School of Information Science and Technology, Fudan University, Shanghai, China
jkyuan22@m.fudan.edu.cn, eetchen@fudan.edu.cn
[2] Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3] The Chinese University of Hong Kong, Central Ave, Hong Kong

Abstract. Domain Adaptation (DA) has been widely explored and made significant progress on cross-domain 3D tasks recently. Despite being effective, existing works fail to deal with rapidly changing domains due to the unpredictable test time scenarios and meanwhile fast response time requirement. Thus, we explore a new task named test-time domain adaptive 3D object detection and propose Reg-TTA3D, a pseudo-label-based test-time adaptative 3D object detection method. By investigating the factor that limits the detection accuracy, we find that regression is essential in this task. To make better regression, we first design a noise-consistency pseudo-label generation process to filter pseudo-labels with instability under noise interference and obtain reliable pseudo-labels. Then, confidence-guided regression refinement is introduced, which uses the box regression results of high-confidence boxes to supervise boxes with relatively low confidence, further making the predicted box size gradually approach the distribution of the target domain. Finally, to better update the regression layer and alleviate the class-imbalance issue, a class-balance EMA updating strategy is proposed. Experimental results on multiple cross-domain scenarios including cross-beam, cross-location, and cross-weather demonstrate that Reg-TTA3D can achieve comparable or even better performance compared to unsupervised domain adaptation works by only updating less than **0.1% parameters** within less than **1% time**.

Keywords: Test-time adaptation · 3D object detection · Regression

1 Introduction

LiDAR-based 3D object detection [12, 18, 20, 21, 32, 36, 43] aims to predict the location and category of each object based on LiDAR point clouds and has attracted increasing attention recently due to its practical applications to autonomous driving. Although significant progress has been made, the stability of detectors under environmental changes, namely cross-domain 3D object detection, remains a challenge. However, the stability is important in practical usage

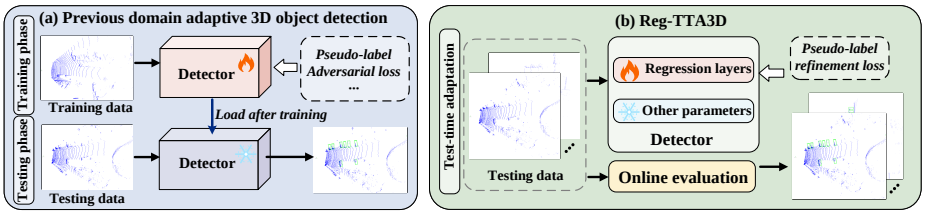

Fig. 1: Comparison between (a) previous domain adaptive 3D object detection and (b) Reg-TTA3D. Previous DA works perform training and testing in a split way. In contrast, Reg-TTA3D tackles domain discrepancies in an online manner and can deal with domain changes rapidly.

since the surrounding environment dynamically changes. For example, weather and geographical environment changes may be encountered in road scenes.

Inspired by unsupervised domain adaption (UDA) techniques in 2D cross-domain tasks [3, 5, 8, 13, 39–41, 44, 45], some works [6, 15, 30, 31, 33, 38] begin to deal with 3D domain discrepancies. ST3D [33] is a pioneer in using self-training to gradually transfer the knowledge learned from the source domain to the target domain. ReDB [6] attempts to deal with multi-class problems by designing cross-domain examination and overlapped boxes counting. Despite achieving large performance gains, the existing UDA model still requires a long training time to be adapted to the target domain, lacking the ability to rapidly deal with environmental changes in an online manner as shown in Fig. 1. Actually, in real applications, models are expected to be continuously updated at a low cost and quickly adapt to changing domains.

To make the model quickly adapt to the changing environment in an online manner, test-time adaptation (TTA) [1, 14, 16, 27, 28] is introduced to adapt the model during test-time with only source model and unlabeled target data. Tab. 1 shows differences between TTA and other adaptation methods. In fact, the TTA technique has been explored in 2D classification tasks such as TENT [27] and SHOT [14] recently, but it is still under-explored in 3D vision fields, especially in detection tasks. To verify the effectiveness of previous TTA methods in the domain adaptive 3D object detection tasks, we conduct experiments by directly applying such methods (*e.g.*, TENT [27], SHOT [14], MATE [16]) to our baseline detector (*i.e.*, SECOND-IoU [32]). As shown in Tab. 3, directly applying previous methods results in a large performance drop, even significantly behind the results of source only which directly evaluates with the pre-trained source model. For instance, TENT [27] only achieves 31.34% / 19.47% in AP_{BEV} / AP_{3D}, with 15.30% / 9.36% declination comparing to source only results.

Accordingly, previous TTA methods are hard to apply to domain adaptive 3D detection tasks due to two major reasons. Firstly, from the perspective of updating parameter selection, previous methods often only update batch normalization (BN) layers to alleviate the domain shifts in the statistical information of network. However, this is not well suited for 3D detection task that also needs rich localization information. Besides, the optimization objectives of previous

Table 1: Comparison of different settings to alleviate domain discrepancies.

Setting	Source data	Target data	Online
Fine-tuning	✗	x^t, y^t	✗
Unsupervised domain adaptation	x^s, y^s	x^t	✗
Source-free domain adaptation	✗	x^t	✗
Test-time adaptation	✗	x^t	✓

methods are designed for classification tasks, which would make the model optimize for extracting features with more category semantic information rather than regression information.

In order to tackle the challenges mentioned above, in this work, we propose Reg-TTA3D to perform test-time adaptation on 3D object detection tasks. To deal with the problem of updated parameter selection, we conduct extensive experiments and observe that high classification accuracy is already achieved by the source model, and the main factor limiting the performance improvement is the inaccurate regression results. Further, by only adapting regression layers, the model can achieve high performance at a low training cost as shown in Tab. 2 which motivates us to focus on regression adaptation in this work. Therefore, to make better regression, we propose noise-consistency pseudo-label generation (NPG) and confidence-guided regression refinement (CRR). The former module aims to filter boxes with instability under noise interference, *i.e.*, noisy point clouds, while the latter one makes the box regression results align better with the distribution of the target domain box. Finally, we introduce a class-balanced EMA updating strategy (CBU) that can effectively update the regression layers, to improve the performance on multi-class simultaneously.

Our contribution can be summarized as follows:

1. We present a new task, namely test-time domain adaptive 3D object detection, and propose Reg-TTA3D to solve this task from the novel view of focusing on adapting only regression parameters.
2. We propose a noise-consistency pseudo-label generation process, a confidence-guided regression refinement loss, and a class-balanced EMA updating strategy, to obtain better detection results and boost the model's performance in an efficient and effective way.
3. Experiments show that Reg-TTA3D reaches a comparable or even better performance by updating less than 0.1% parameters within less than 1% time compared to UDA works, and significantly outperforms previous TTA methods on multiple cross-domain scenarios.

2 Related Works

2.1 LiDAR-based 3D Object Detection

With the development of autonomous driving, LiDAR-based 3D object detection has attracted increasing attention. Current prevailing LiDAR-based 3D detectors can be roughly divided into point-based, voxel-based, and point-voxel-based methods according to the point cloud processing procedure. Point-based

methods [20, 35, 43] first sample a subset of point clouds, then extract features and generate proposals from sampled points. Among them, PointRCNN [20] uses a two-stage framework that is composed of a bottom-up proposal generation process and a proposal refinement process. In contrast, voxel-based methods [7, 12, 32, 36, 46] first divide disordered points into regular grids which are then fed into the convolutional network. SECOND [32] is a pioneer in using the 3D sparse backbone to extract features. CenterPoint [36] introduces a center-based detector and achieves promising performance. Point-voxel-based [18, 19] methods integrate the features extracted from points and voxels to obtain better feature representations. However, existing 3D detectors fail to deal with domain gaps which is essential in practical usage.

2.2 Test-time Adaptation

Test-time adaptation [1, 4, 10, 22, 24, 27, 28] aims to rapidly adapt the source model to target domain without accessing source domain data during test time, which is more practical in the real world since it can handle dynamic domain shifts. TENT [27] explores test-time adaptation for the first time and updates the batch normalization layers using designed entropy minimization loss. SHOT [14] deals with such a challenging task by maximizing the mutual information. More recently, MATE [16] extends test-time adaptation to 3D point clouds and finds that the model can be adapted effectively by masking and reconstruction.

Although TTA has been extensively explored in 2D scenarios, it is still under-explored in 3D tasks, especially in large-scale point cloud scenarios (*e.g.*, autonomous driving scenarios). Besides, most previous TTA works mainly focus on classification tasks, such as classification [1, 16, 27] and segmentation [22], and they fail to tackle detection tasks since detection considers classification and box regression at the same time. In contrast, Reg-TTA3D considers TTA for 3D object detection tasks from the perspective of regression, which is more suitable for 3D detection tasks.

2.3 Domain Adaptation for 3D Object Detection

3D detectors often suffer large performance drops when training and testing under different domains, which makes them unstable when the surrounding environment dynamically changes. Recently, researchers have tried to tackle such a problem using domain adaptation techniques [6, 17, 29, 33, 34, 38]. SN [29] proposes a statistical method to align the size of bounding boxes between source and target domains. ST3D [33] and ST3D++ [34] design a self-training pipeline and mitigate domain gaps using a memory bank and curriculum data augmentation. To deal with the class imbalance issue, ReDB [6] proposes a pseudo-label generation pipeline to obtain reliable, diverse, and class-balanced pseudo-labels. However, the existing works cannot deal with the emergency changes in the environment due to the time-consuming and large amount of calculation.

Table 2: Performance of updating different parameters.

Tuning layers	Waymo → KITTI	nuScenes → KITTI	Parameters
Batch normalization	52.84 / 31.77	31.86 / 18.25	9.08k
Classification layers	52.82 / 36.37	31.51 / 19.76	1.17k
Regression layers	59.38 / 45.54	**33.63** / **21.80**	2.73k
Detection head	59.62 / 46.08	32.51 / 20.39	4.68k
All parameters	**60.78** / **46.71**	33.23 / 19.60	12.12M

3 Method

The overall framework of Reg-TTA3D is shown in Fig. 2. To better illustrate our proposed method, we start by introducing the problem formulation and our baseline in Sec. 3.1. Then, we give an analysis of the updated parameters selection in Sec. 3.2. Finally, we detail each module of Reg-TTA3D and give the overall objective and adaptation strategy in Sec. 3.3 and Sec. 3.4.

3.1 Preliminary

Problem Formulation. Following typical test-time adaptation setting [22, 27], given the model M_S pre-trained on the source domain $D_S = \{(x_i^S, y_i^S)\}_{i=1}^{n_S}$ and an unlabeled target set $D_T = \{x_j^T\}_{j=1}^{n_T}$, where the source and target domain follow different distributions, our goal is to adapt the source model to the target domain without accessing source domain data in an online manner. During adaptation, in each timestep, a batch of unlabeled data $[x_1^t, x_2^t, ..., x_N^t]$ is sampled from D_T to update the model, where N is batch size.

Baseline Introduction. Following [6, 34], we use SECOND-IoU [32] as our baseline detector which is composed of a feature extractor (*i.e.*, 3D and 2D backbone) to obtain bird's eye view (BEV) features, a region proposal network (RPN) to generate proposals and a detection head to provide the classification and regression results. The detector is trained using the following loss function:

$$\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{rcnn}, \tag{1}$$

where $\mathcal{L}_{rpn}$ represents the loss of RPN and $\mathcal{L}_{rcnn}$ denotes box refinement loss.

3.2 Updated Parameters Selection

The aim of Reg-TTA3D is to transfer the source model to the target domain rapidly so that the model can tackle the continuous changes in the surrounding environment. To achieve this goal, the amount of updated parameters we select needs to be relatively small, and the accuracy needs to be largely improved by only updating selected parameters.

Previous TTA works [22, 27] mainly focus on adapting batch normalization (BN) layers to mitigate the domain discrepancies, since BN layers contain the statistics of domains. Such an adaptation strategy is effective for domain adaptive classification tasks, as more accurate semantic representations can be obtained, which is especially important for classification. However, only adapting

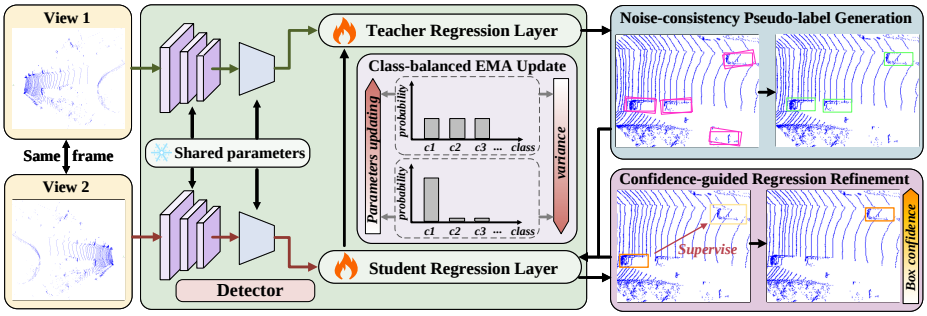

Fig. 2: The overall framework of the proposed Reg-TTA3D, which employs SECOND-IoU [32] as the baseline detector and consists of noisy-consistency pseudo-label generation process (NPG), confidence-guided regression refinement (CRR) and class-balanced EMA updating strategy. The data are first fed into the teacher model to generate pseudo-labels as the supervision to the student model using NPG. Then, the student model is trained using data from another view and supervised by detection loss and refinement loss obtained by CRR. Finally, the class-balanced EMA updating strategy is used to update the teacher model. **Note that the only difference between the teacher model and the student model lies in the regression layers.**

BN layers cannot rapidly transfer the source detector to the target domain since detection tasks contain both classification and regression. Some works [16, 26] try to solve this problem by updating the entire model but suffer from huge computational costs. To explore an effective and efficient method, by conducting extensive experiments, we find that the model can adapt to the target domain efficiently by only updating regression layers. As shown in Tab. 2, we fine-tune the parameters using 1% KITTI training data for 1 epoch and observe that only fine-tuning the regression layers can reach relatively high performance and even surpass the performance of fine-tuning the entire model. This is because in the autonomous driving scenario, there exists a large difference between categories such as car and pedestrian, the source model can already achieve high classification accuracy and the main factor for limiting the detection accuracy is the inaccurate regression results. Based on the observation, in this work, we focus on studying how to only adapt the parameters of regression layers and make better regression.

3.3 Reg-TTA3D

As mentioned above, the quality of box regression is essential to TTA for 3D object detection. To make better regression, we propose noise-consistency pseudo-label generation to obtain better regression supervision, and confidence-guided regression refinement to further refine the regression results. Besides, to effectively update the regression layers, we design a class-balanced EMA updating strategy to avoid overfitting the model to a single class.

Noise-consistency Pseudo-label Generation. As mentioned in [34, 38], the pseudo-labels directly generated by the source model contain a lot of noise due

to domain discrepancies. Previous methods try to alleviate pseudo-label noise by weighting pseudo-labels obtained in different epochs [33, 34], or adding the groundtruth instances from the source domain to target domain frames [6]. However, these methods cannot apply to test-time adaptation due to the inaccessibility of source data and online adaptation.

Inspired by test-time augmentation methods which comprehensively consider the model predictions under different data augmentation methods to provide final results. Specifically, the model's prediction of real objects is relatively stable, while being uncertain for the noise regions. To this end, we propose noise-consistency pseudo-label generation to obtain more accurate pseudo-labels, by measuring the stability of model predictions when adding random noise to regions of interest (RoIs). In detail, given proposals generated by the teacher model $B^t = \{(c_x^t, c_y^t, c_z^t, l^t, w^t, h^t, \theta^t)_i\}_{i=1}^n$, its corresponding features $\{f_i^t\}_{i=1}^n$ and confidence scores $\{s_i^t\}_{i=1}^n$ where (c_x^t, c_y^t, c_z^t) and (l^t, w^t, h^t) denote the center and the size of the box respectively, θ^t is the rotation angle and n is the number of boxes. We can obtain boxes with noise $B^{t'} = \{(c_x^{t'}, c_y^{t'}, c_z^{t'}, l^{t'}, w^{t'}, h^{t'}, \theta^{t'})\}_{i=1}^n$, corresponding features $\{f_i^{t'}\}_{i=1}^n$ and scores $\{s_i^{t'}\}_{i=1}^n$, by adding random noise to boxes' size and angle. To measure the stability of RoIs, we calculate the inconsistency rate as follows:

$$R_{inconsistency,i} = 1 - \frac{|s_i^t - s_i^{t'}|}{\mathrm{IoU}(b_i^t, b_i^{t'})}, \tag{2}$$

where $\mathrm{IoU}(b_i^t, b_i^{t'})$ denotes the intersection of union (IoU) between $b_i^t \in B^t$ and $b_i^{t'} \in B^{t'}$. A high inconsistency rate represents that the model prediction is unstable to the current object which is more likely to be a noise region. Therefore, we filter the boxes with high inconsistency rate by a threshold τ to get boxes and scores $\hat{B}, \{\hat{s}_i\}_{i=1}^n$ and $\hat{B}', \{\hat{s}_i'\}_{i=1}^n$ with stability under noise interference. Then, we ensemble $\hat{B}$ and $\hat{B}'$ by confidence scores as follows:

$$\tilde{B} = \{\tilde{b_i}\}, \;\; \tilde{b_i} = \begin{cases} \hat{b}_i, & \hat{s}_i > \hat{s}_i', \, \hat{b}_i \in \hat{B} \\ \hat{b}_i', & \hat{s}_i < \hat{s}_i', \, \hat{b}_i' \in \hat{B}' \end{cases}, \tag{3}$$

After obtaining $\tilde{B}$, the boxes are filtered by non-maximum suppression (NMS) with a relatively high confidence threshold, to remove redundant boxes and boxes with low confidence. Finally, the pseudo-labels B^{npg} and $\{l_i^{npg}\}_{i=1}^n$ with high stability and confidence can be acquired where $\{l_i^{npg}\}_{i=1}^n$ is classification results.

Confidence-guided Regression Refinement. Although lots of boxes containing noisy point cloud can be removed by measuring the stability under noise injection, there still exist the problem of inaccurate box regression results. This is because the distributions of bounding box size are inconsistent between source and target domains as mentioned in [29,37,42], and it is difficult to mitigate such domain discrepancy with only NPG. By analyzing the distribution of predicted bounding boxes, we find that boxes with higher confidence scores align better with the distribution of the target domain boxes as shown in Fig. 3, since precise boxes correspond to high-quality RoI features that are less disturbed by background representations. Based on this observation, we design a confidence-guided regression refinement to obtain more precise regression results of box size.

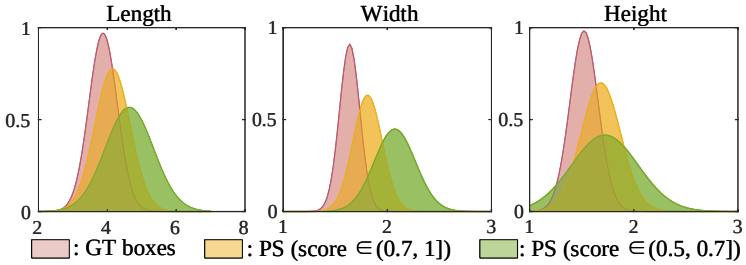

Fig. 3: Distribution of pseudo box size under different thresholds. PS denotes pseudo boxes.

In particular, given predicted boxes generated by the student model $B^s = \{(c_x^s, c_y^s, c_z^s, l^s, w^s, h^s, \theta^s)_j\}_{j=1}^k$ and scores $\{s_j^s\}_{j=1}^k$ of a batch of frames, where k is the total number of predicted boxes. We sort boxes belonging to the same class according to scores in descending order, and sample a proportion of boxes with high scores to calculate the mean value of sample boxes size $(\overline{l}^s, \overline{w}^s, \overline{h}^s)$. Please refer to our supplementary material for the study of the proportion of sampled boxes. Once obtained $(\overline{l}^s, \overline{w}^s, \overline{h}^s)$, we can use it as a supervision signal to make the box regression value closer to the distribution of the target domain as follows:

$$\begin{aligned} \mathcal{L}_{refine} &= \mathbb{E}_{i=1}^{n} \Sigma_{d \in [d_l, d_w, d_h]} d, \\ where \quad d &= \begin{cases} 0, & |(z^s - \overline{z}^s)| < p \\ (z^s - \overline{z}^s)^2, & |(z^s - \overline{z}^s)| \geq p \end{cases}, \end{aligned} \tag{4}$$

where $z \in [l^s, w^s, h^s]$, $\overline{z} \in [\overline{l}^s, \overline{w}^s, \overline{h}^s]$, n is batch size and p is set 0.1 to allow the size of objects to fluctuate within a certain range. The model can predict more accurate box size when using $\mathcal{L}_{refine}$ for optimization.

Class-balanced EMA Updating. To make the model more stable, we update the teacher regression layers by exponential moving average (EMA). However, the model tends to overfit to the category with a large number of instances (*i.e.*, car). As a result, the detection performance on categories with few instances will decrease. To make the model better balance the performance improvement for each class, we propose a class-balanced EMA updating strategy to make the model update more when the class is balanced, while updating less when a single category dominates. Specifically, given the pseudo-labels obtained from NPG $B^{npg} = \{b_i^{npg} \in \mathbb{R}^7\}$ and its corresponding class labels $\{c_i^{npg}\}$. We calculate the proportion $\{p_c\}_{c=1}^C$ of each category in pseudo-labels according to $\{c_i^{npg}\}$, where C is the number of total categories and $\Sigma_{c=1}^C p_c = 1$. Then the variance of $\{p_c\}_{c=1}^C$ is calculated and normalized to $[0.99, 0.999]$ to obtain the momentum α. A high value of α means that a severe class-imbalance issue exists in the current batch and the model should update less. Finally the parameter of the teacher regression layer ω_{reg}^t can be updated by the following formula:

$$\omega_{reg}^t \leftarrow \alpha \omega_{reg}^t + (1 - \alpha)\omega_{reg}^s, \tag{5}$$

where ω_{reg}^s is the parameters of student regression layers.

Algorithm 1 Single iteration of Reg-TTA3D strategy

Input: A batch of frames $\{x_i\}_{i=1}^n$ and $\{\dot{x_i}\}_{i=1}^n$, where $\dot{x_i}$ is obtained by performing data augmentation on x_i.
Output: Predictions of current batch B^t and $\{l_i^t\}_{i=1}^n$

1: Get the predictions $B^t = \{b_i^t \in \mathbb{R}^7\}_{i=1}^n$ and $\{l_i^t\}_{i=1}^n$ of $\{x_i\}_{i=1}^n$ using teacher model, B^t and $\{l_i^t\}_{i=1}^n$ are the detection results of current batch.
2: Use NPG to obtain boxes B^{npg} and $\{l_i^{npg}\}_{i=1}^n$ with stability under noise interference as pseudo-labels.
3: Get the predictions of student model B^s and $\{s_i^s\}_{i=1}^n$ of $\{\dot{x_i}\}_{i=1}^n$, supervise the student model by B^{npg} and $\{l_i^{npg}\}_{i=1}^n$ to get $\mathcal{L}_{det}$.
4: Calculate $\mathcal{L}_{refine}$ using CRR.
5: Optimize the student model using Eq. (6).
6: Update the teacher model using CBU and Eq. (5).
7: **Return:** Current predictions B^t and $\{l_i^t\}_{i=1}^n$.

3.4 Overall Objective and Test-time Adaptation Strategy

Overall Objective. The overall objective to optimize the student regression layers can be formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \mathcal{L}_{refine}, \tag{6}$$

where $\mathcal{L}_{det}$ and $\mathcal{L}_{refine}$ are defined in Eq. (1) and Eq. (4).
Test-time Adaptation Strategy. The test-time adaptation strategy of Reg-TTA3D can be summarized in Algorithm 1.

4 Experiments

4.1 Experimental Setup

Datasets and Cross-domain Settings. We conduct experiments on four commonly-used datasets: Waymo [23], KITTI [9], nuScenes [2] and nuScenes-C [11]. To verify the effectiveness of Reg-TTA3D, we consider three cross-domain settings including cross-location (*i.e.*, Waymo → KITTI), cross-beam (*i.e.*, Waymo → nuScenes, nuScenes → KITTI) and cross-weather (*i.e.*, nuScenes → nuScenes-C). Note that in the cross-weather setting, we use the medium-difficulty fog and snow scenarios in the nuScenes-C dataset. Following ReDB [6], in the first two cross-domain settings, we use the KITTI evaluation metric to evaluate for three common classes in autonomous driving scenarios (*i.e.*, car, pedestrian, and cyclist), we report the average precision in both 3D (*i.e.*, AP_{3D}) and BEV (*i.e.*, AP_{BEV}) over 40 recall positions, with IoU threshold 0.7 for car and 0.5 for pedestrian and cyclist. In the cross-weather setting, we use the official nuScenes evaluation metric and report mean average precision (mAP).
Implementation Details. Following [6], we evaluate the proposed Reg-TTA3D on SECOND-IoU [32], and the source model is trained for all categories using random object scaling (ROS) augmentation method. We set hyperparameter τ

to 1.5 and p to 0.1 for all cross-domain settings and use Adam optimizer with learning rate of 1×10^{-3}. We utilize commonly-used data augmentation methods (*e.g.*, random world flip, random world rotation, random world scaling) to get $\{\dot{x_i}\}_{i=1}^{n}$. Our code is built on 3DTrans [25].

Comparison Baselines. To verify the effectiveness of Reg-TTA3D, we mainly compare our methods to unsupervised domain adaptive 3D object detection and previous test-time adaptation methods that we apply to 3D object detection. **UDA methods:** (1) ST3D [33] is a self-training method that continuously updates the pseudo-labels, (2) ST3D++ [34] provides more insight analysis and extends ST3D to get better results, (3) ReDB [6] is a pioneer to perform domain adaptive 3D detection on multiple categories. **TTA methods:** (1) TENT [27] proposes an entropy minimization method which focuses on adapting BN layers, (2) SHOT [14] optimizes feature extraction module using both information maximization and pseudo-labeling. (3) MATE [16] uses masked autoencoder to apply test-time adaptation to 3D classification.

4.2 Main Results

Results on cross-location and cross-beam scenarios. We first evaluate Reg-TTA3D on cross-location and cross-beam scenarios and compare the results with previous UDA methods. As shown in Tab. 3, Reg-TTA3D can greatly improve the detection accuracy on Waymo → KITTI and nuScenes → KITTI settings, largely narrow the performance gap between source only and oracle. For example, in the Waymo → KITTI setting, our Reg-TTA3D can achieve 61.78% / 49.96% in AP_{BEV} / AP_{3D}, even surpass UDA methods such as ST3D [33] and ST3D++ [34] and it is comparable to the state-of-the-art UDA method (*i.e.*, 61.14% / 50.10% in AP_{BEV} / AP_{3D} reported in ReDB [6]). In nuScenes → KITTI and Waymo → nuScenes cross-domain scenarios, the performance of Reg-TTA3D is also better than some UDA methods (*e.g.*, ST3D [33]) which needs source domain data and training for multiple epochs. In contrast, Reg-TTA3D only needs target data and one epoch of adaptation which greatly reduces the time consumption.

Further, we compare Reg-TTA3D with the previous TTA methods in Tab. 3. We find that previous TTA methods suffer performance drops when applied to 3D detection tasks (*e.g.*, TENT [27] reduce the detection accuracy to 31.34% / 19.47% in the Waymo → KITTI setting). This is mainly because the detection task is composed of classification and regression and existing methods only consider the classification tasks. As a result, the model will concentrate on extracting effective representations for the classification tasks and ignore the localization information. However, our Reg-TTA3D fully considers the challenges in detection tasks and can improve the performance on multiple categories.

Results on the cross-weather scenario. To more comprehensively evaluate the effectiveness of our proposed method, we conduct experiments in a more realistic scenario (*i.e.*, cross-weather). Since nuScenes-C dataset only has the validation set, we only compare Reg-TTA3D with existing TTA methods. It can be seen in Tab. 4 that Reg-TTA3D can improve the performance on both snow

Table 3: Results on cross-location and cross-beam scenarios. Following [6, 33], we use SECOND-IoU [32] as out baseline detector. Source Only represents that the pre-trained detector is directly evaluated on the target domain, and Oracle denotes the detection results obtained using the fully annotated target training data. Red: Best UDA method, **Blue**: Best TTA method.

Task	Method	Setting	Car	Pedestrian	Cyclist	Mean AP
Waymo → KITTI	Source Only	-	51.48 / 19.78	40.80 / 31.26	47.63 / 35.45	46.64 / 28.83
	SN	-	76.61 / 54.14	52.48 / 48.20	34.56 / 32.74	54.55 / 45.03
	ST3D	UDA	77.62 / 49.24	44.45 / 42.04	47.74 / 42.95	56.60 / 44.70
	ST3D++	UDA	77.68 / 50.03	49.09 / 46.19	51.50 / 47.70	59.42 / 47.97
	ReDB	UDA	80.37 / 54.12	51.01 / 48.20	52.05 / 47.97	61.14 / 50.10
	TENT	TTA	71.98 / 39.01	4.54 / 3.95	17.51 / 15.45	31.34 / 19.47
	SHOT	TTA	79.52 / 41.31	33.52 / 29.02	40.79 / 35.83	51.27 / 35.38
	MATE	TTA	75.17 / 30.64	39.94 / 36.92	43.72 / 40.22	52.94 / 35.92
	Reg-TTA3D	TTA	**81.60** / **56.03**	**50.89** / **45.43**	**52.85** / **48.43**	**61.78** / **49.96**
	Oracle	-	83.29 / 73.45	46.64 / 41.33	62.92 / 60.32	64.28 / 58.37
Waymo → nuScenes	Source Only	-	30.64 / 17.18	1.81 / 0.58	0.97 / 0.88	11.14 / 6.22
	SN	-	29.56 / 17.78	1.33 / 1.07	2.92 / 2.61	11.27 / 7.15
	ST3D	UDA	28.42 / 17.83	1.64 / 1.39	4.01 / 3.54	11.36 / 7.59
	ST3D++	UDA	28.87 / 19.15	1.82 / 1.58	4.09 / 3.74	11.60 / 8.16
	ReDB	UDA	30.12 / 18.56	2.47 / 2.14	6.56 / 5.19	13.05 / 8.63
	TENT	TTA	16.40 / 3.43	0.82 / 0.25	0.28 / 0.21	5.83 / 1.29
	SHOT	TTA	12.26 / 1.42	1.56 / 1.36	0.22 / 0.20	4.68 / 0.99
	MATE	TTA	8.64 / 3.99	1.72 / 0.97	0.24 / 0.08	3.53 / 1.68
	Reg-TTA3D	TTA	**30.54** / **18.15**	**2.04** / **1.75**	**5.01** / **4.70**	**12.53** / **8.20**
	Oracle	-	51.88 / 34.87	25.24 / 18.92	15.06 / 11.73	30.73 / 21.84
nuScenes → KITTI	Source Only	-	39.15 / 7.65	21.54 / 16.87	6.31 / 2.44	22.33 / 8.98
	SN	-	56.08 / 28.67	23.05 / 16.84	5.11 / 2.32	28.08 / 15.94
	ST3D	UDA	71.50 / 48.09	22.64 / 17.61	7.86 / 5.20	34.00 / 23.64
	ST3D++	UDA	69.90 / 44.62	24.11 / 18.20	10.14 / 6.39	33.75 / 23.07
	ReDB	UDA	74.23 / 51.31	25.95 / 18.38	13.82 / 8.64	38.00 / 26.11
	TENT	TTA	43.35 / 19.83	9.40 / 5.77	2.71 / 2.70	18.49 / 9.43
	SHOT	TTA	60.24 / 23.01	17.51 / 9.44	2.47 / 1.66	26.74 / 11.37
	MATE	TTA	61.89 / 27.13	15.98 / 13.11	2.94 / 2.45	26.93 / 14.23
	Reg-TTA3D	TTA	**68.73** / **44.56**	**22.78** / **17.71**	**10.21** / **7.12**	**33.90** / **23.13**
	Oracle	-	83.29 / 73.45	46.64 / 41.33	62.92 / 60.32	64.28 / 58.37

and fog scenarios (*e.g.*, 45.68% → 47.09% mAP on the Normal → Snow setting). The results verify that our proposed method can handle dynamic environment changes in road scenes and can be applied in more practical scenarios.

4.3 Insight Analyses

Sensitivity to Detector Architecture. Following [6], to verify the effectiveness of Reg-TTA3D on different types of detectors, we further conduct experiments on nuScenes → KITTI cross-domain scenario using PointRCNN [20] as the baseline detector. As show in Tab. 5, our Reg-TTA3D outperforms all TTA methods and some UDA methods. Note that compared to the source-free domain adaptation method (*i.e.*, SF-UDA^{3D} [17]) which is inaccessible to the source data and need to adapt for multiple epochs, our Reg-TTA3D adapted on multi-class can achieve a better result on the car category (*e.g.*, 50.87% compared to 49.80%

Table 4: Results on cross-weather scenario (*i.e.*, nuScenes Normal → Fog and Normal → Snow). Fog and snow data are from nuScenes-C [11] dataset. We use CenterPoint [36] as our baseline detector since it is the most widely used detector on nuScenes.

Method	Task	mAP	Car	Truck	CV.	Bus	Trailer	Barrier	Motor.	Bicycle	Ped.	TC.
Source Only	Fog	48.40	79.12	52.58	8.10	67.53	31.00	**64.40**	7.30	26.65	81.28	65.97
TENT		41.83	73.62	15.23	7.58	58.04	21.85	60.45	1.56	23.84	77.59	52.24
SHOT		42.31	72.63	40.70	7.59	58.18	22.24	61.71	0.85	26.45	77.84	54.87
MATE		23.97	64.61	24.18	3.34	27.26	14.11	20.45	0.00	7.67	49.67	28.33
Reg-TTA3D		**49.67**	**80.30**	**54.43**	**10.86**	**68.65**	**34.53**	63.52	**7.79**	**27.07**	**82.20**	**67.28**
Source Only	Snow	45.86	80.74	48.77	4.68	64.15	28.84	62.62	**9.02**	24.72	78.37	56.62
TENT		40.51	75.09	39.37	6.51	57.57	19.96	59.82	1.89	23.33	76.78	44.73
SHOT		42.15	74.66	39.74	7.01	58.12	21.46	62.94	1.19	27.30	78.39	50.66
MATE		23.43	63.98	22.57	3.20	25.30	13.06	22.79	0.00	7.64	0.50	24.81
Reg-TTA3D		**47.09**	**82.01**	**49.71**	**7.19**	**68.16**	**30.54**	**62.96**	6.33	**26.29**	**80.37**	**57.30**

Table 5: Performance comparisons (AP_{3D}) with PointRCNN as baseline detector on nuScenes → KITTI task. ‡ indicates the results reported in the original paper. Red: Best UDA method, **Blue**: Best TTA method.

Method	Setting	Car			Pedestrian			Cyclist			Average		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
Source Only	-	42.77	32.11	28.75	43.26	37.29	33.16	3.28	4.09	4.18	29.77	24.60	22.03
SN		66.56	50.32	45.92	42.96	37.15	32.45	9.07	7.57	7.42	39.53	31.68	28.60
ST3D	UDA	48.85	41.90	38.92	45.67	38.71	33.09	26.50	19.35	18.38	40.34	33.32	30.13
ST3D++		60.45	49.36	45.88	50.77	42.43	36.64	27.20	17.94	16.52	46.14	36.58	33.01
ReDB		71.45	57.9	53.91	52.32	44.33	37.95	45.13	32.93	31.05	56.30	45.05	40.97
SF-UDA^{3D} ‡	SFDA	68.80	49.80	45.00	-	-	-	-	-	-	-	-	-
TENT	TTA	42.55	32.41	27.61	30.71	26.09	23.78	4.24	3.66	3.32	25.83	20.70	18.23
SHOT		30.09	23.09	20.41	19.24	16.29	14.31	2.43	1.70	1.51	17.25	13.69	12.07
Reg-TTA3D		**65.20**	**50.87**	**46.72**	**45.80**	**42.55**	**34.90**	**40.25**	**27.87**	**25.29**	**50.41**	**40.43**	**35.63**

reported by SF-UDA^{3D} in moderate level) in one epoch of adaptation even if SF-UDA^{3D} is trained on the unfair single class. The results further show that our proposed methods can applied to different detector architectures.

Ablation Studies. To further verify the effectiveness of Reg-TTA3D, we conduct extensive experiments to investigate the impact of each component we proposed including NPG, CRR and CBU. As shown in Tab. 6, the improvement (*i.e.*, 0.92% AP_{3D}) of using naive pseudo-labels is relatively small compared to the performance of source only since pseudo-labels contain lots of noise samples and the regression results are not accurate due to domain discrepancies. Reg-TTA can effectively handle such a problem by proposing NPG and CRR modules, with 2.90% and 3.60% improvement respectively compared to source only. Besides, when combining these two modules, the performance can be further improved because NPG focuses on filter noise samples and CRR concentrates on making more accurate regression results. Further, by combining with CBU which makes the model update in a class-balance manner, the performance can be further boosted and achieve 49.96% AP_{3D}.

Sensitivity to Hyperparameters. We ablate the impact of hyperparameters τ (*i.e.*, the threshold of inconsistency rate) and p in this subsection. As shown in Tab. 7, we first set τ to 1.0, 1.5, 2.0 respectively and observe that the fluctuation

Table 6: Ablation studies on each component. PS, NPG, CRR and CBU represent using naive pseudo-labels, noise-consistency pseudo-label generation, confidence-guided regression refinement and class-balanced EMA updating, respectively. The experiments are conducted on Waymo → KITTI scenario using SECOND-IoU as the baseline detector.

PS	NPG	CRR	CBU	mean AP_{BEV}	mean AP_{3D}
-	-	-	-	57.89	42.01
✓	-	-	-	58.73	42.93
✓	✓	-	-	58.89	44.91
✓	-	✓	-	59.84	45.61
✓	-	-	✓	59.73	44.05
✓	✓	✓	-	60.57	47.97
✓	✓	-	✓	60.12	46.62
✓	-	✓	✓	60.59	48.39
✓	✓	✓	✓	**61.78**	**49.96**

Table 7: Ablation studies of hyperparameters τ and p. The experiments are conducted on Waymo → KITTI scenario using SECOND-IoU. Note that when ablate on τ, we set $p = 0.1$ and when ablate on p, we set $\tau = 0.15$.

τ	AP_{BEV} / AP_{3D}
1.0	60.92 / 49.12
1.5	**61.78** / **49.96**
2.0	60.44 / 48.88

p	AP_{BEV} / AP_{3D}
0.05	60.29 / 48.26
0.10	**61.78** / **49.96**
0.15	61.56 / 49.63

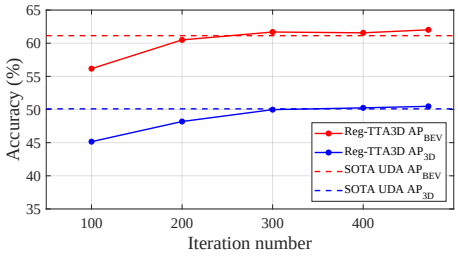

Fig. 4: The relationship between detection accuracy and the iteration number.

Table 8: Ablation studies on costs. The experiments are conducted on W → K scenario using SECOND-IoU. We use Tesla A100 to conduct the experiments.

Method	Time Cost	N_GPUs	Param.	Results
ST3D	4h	4	12.12M	56.60 / 44.70
ST3D++	3.5h	4	12.12M	59.42 / 47.97
ReDB	3.2h	4	12.12M	61.14 / **50.10**
TENT	**4min**	1	9.08K	31.34 / 19.47
MATE	6min	1	0.71M	52.94 / 35.92
Reg-TTA3D	4.5min	1	**2.73K**	**61.78** / 49.96

of the performance is 1.34% / 1.08%. Then, by setting p to different values (*i.e.*, 0.05, 0.1, 0.15), we find that there is only little fluctuation and the best performance can achieved when set to 0.1. The results further show that our method is robust to these hyperparameters.

Analysis on Pseudo-labeling Accuracy. In this section, we measure the accuracy of each iteration during adaptation. Fig. 4 reflects the detection accuracy on Waymo → KITTI as iteration increases. It can be seen that the accuracy continuously improves since the model is gradually adapted to the target domain which also verifies the stability of Reg-TTA3D during adaptation.

Analysis on Costs. Further, we report the time cost and total updated parameters in Tab. 8. Notably, Reg-TTA3D achieves comparable results to UDA methods adapting only less than 0.1% of parameters within less than 1% of time. When compared to TTA methods, our proposed method can achieve much higher performance with less updated parameters and comparable time cost which further verifies that our Reg-TTA3D is an effective and efficient method.

Ablation Studies of the momentum. We ablate the momentum in EMA updating and compare it to our class-balance EMA updating strategy. Specifi-

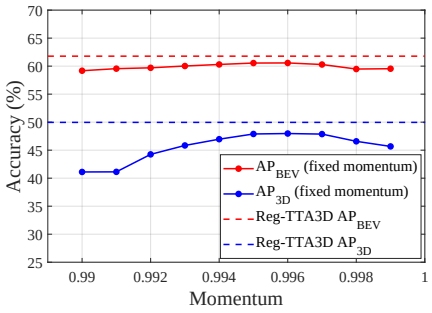

Fig. 5: The impact of momentum and comparisons with CBU.

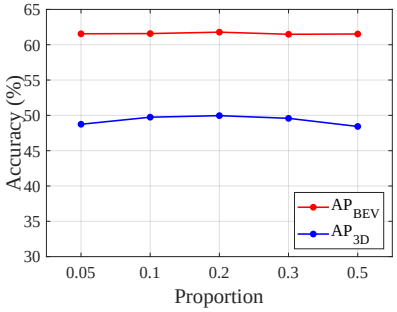

Fig. 6: ablation studies on the proportion of selected samples.

cally, we conduct the experiment for momentum between 0.99 and 0.999 at 0.001 intervals. As shown in Fig. 5, the result demonstrates that CBU can better update the regression layer due to it can balance update the model and improve the performance on multiple classes at the same time.

Ablation Studies on the proportion of Selected samples in CRR. In our manuscript, we use 20% boxes with the highest scores to supervise box sizes with relatively low scores. Here, we analyze the impact on the proportion of selected samples. As shown in Fig. 6, when the proportion is relatively high, the average box size will be distributed far from the target domain, and when set to relatively small values, the small number of samples results in the averages being affected by individual instances. Based on the experiments and analysis, we set the proportion to 20%.

5 Conclusion

In this work, for the first time, we explore test-time adaptation in 3D object detection tasks and propose Reg-TTA3D, which handles rapid changes in the surrounding environment from the perspective of updating parameters selection and adaptation algorithm. By adapting using the proposed noise-consistency pseudo-label generation process, confidence-guided regression refinement loss and class-balanced EMA updating strategy and only updating regression layers, Reg-TTA3D can achieve comparable performance to some UDA methods at a low cost.

6 Limitation

Although the proposed Reg-TTA3D can handle dynamic changes in the surrounding environments such as continuously changing weather. It cannot deal with emergencies such as some corner cases which is also important in real-world autonomous driving.

Acknowledgement

This work is supported by National Key Research and Development Program of China (No. 2022ZD0160101), National Natural Science Foundation of China (No. 62071127, and 62101137), Shanghai Natural Science Foundation (No. 23ZR1402900), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103). This work is also supported by the National Key R&D Program of China (Grant No. 2022ZD0160104), and Shanghai Rising Star Program (Grant No. 23QD1401000). The computations in this research were performed using the CFFF platform of Fudan University.

References

1. Boudiaf, M., Mueller, R., Ben Ayed, I., Bertinetto, L.: Parameter-free online test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8344–8353 (2022)
2. 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: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11621–11631 (2020)
3. Cai, Q., Pan, Y., Ngo, C.W., Tian, X., Duan, L., Yao, T.: Exploring object relation in mean teacher for cross-domain detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11457–11466 (2019)
4. Chen, D., Wang, D., Darrell, T., Ebrahimi, S.: Contrastive test-time adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 295–305 (2022)
5. Chen, Y., Li, W., Sakaridis, C., Dai, D., Van Gool, L.: Domain adaptive faster r-cnn for object detection in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3339–3348 (2018)
6. Chen, Z., Luo, Y., Huang, Z., Wang, Z., Baktashmotlagh, M.: Revisiting domain-adaptive 3d object detection by reliable, diverse and class-balanced pseudo-labeling. arXiv preprint arXiv:2307.07944 (2023)
7. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1201–1209 (2021)
8. Ganin, Y., Lempitsky, V.: Unsupervised domain adaptation by backpropagation. In: International conference on machine learning. pp. 1180–1189. PMLR (2015)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3354–3361 (2012)
10. Goyal, S., Sun, M., Raghunathan, A., Kolter, J.Z.: Test time adaptation via conjugate pseudo-labels. Advances in Neural Information Processing Systems **35**, 6204–6218 (2022)
11. Kong, L., Liu, Y., Li, X., Chen, R., Zhang, W., Ren, J., Pan, L., Chen, K., Liu, Z.: Robo3d: Towards robust and reliable 3d perception against corruptions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 19994–20006 (2023)

12. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
13. Liang, J., Gong, K., Li, S., Liu, C.H., Li, H., Liu, D., Wang, G., et al.: Pareto domain adaptation. Advances in Neural Information Processing Systems **34**, 12917–12929 (2021)
14. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning. pp. 6028–6039. PMLR (2020)
15. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8866–8875 (2021)
16. Mirza, M.J., Shin, I., Lin, W., Schriebl, A., Sun, K., Choe, J., Kozinski, M., Possegger, H., Kweon, I.S., Yoon, K.J., et al.: Mate: Masked autoencoders are online 3d test-time learners. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16709–16718 (2023)
17. Saltori, C., Lathuiliére, S., Sebe, N., Ricci, E., Galasso, F.: Sf-uda 3d: Source-free unsupervised domain adaptation for lidar-based 3d object detection. In: 2020 International Conference on 3D Vision (3DV). pp. 771–780. IEEE (2020)
18. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
19. Shi, S., Jiang, L., Deng, J., Wang, Z., Guo, C., Shi, J., Wang, X., Li, H.: Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. International Journal of Computer Vision **131**(2), 531–551 (2023)
20. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019)
21. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. IEEE transactions on pattern analysis and machine intelligence **43**(8), 2647–2664 (2020)
22. Shin, I., Tsai, Y.H., Zhuang, B., Schulter, S., Liu, B., Garg, S., Kweon, I.S., Yoon, K.J.: Mm-tta: multi-modal test-time adaptation for 3d semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16928–16937 (2022)
23. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)
24. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: International conference on machine learning. pp. 9229–9248. PMLR (2020)
25. Team, D.D.: 3dtrans: An open-source codebase for exploring transferable autonomous driving perception task. `https://github.com/PJLab-ADG/3DTrans` (2023)
26. VS, V., Oza, P., Patel, V.M.: Towards online domain adaptive object detection. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 478–488 (2023)

27. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. arXiv preprint arXiv:2006.10726 (2020)
28. Wang, Q., Fink, O., Van Gool, L., Dai, D.: Continual test-time domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7201–7211 (2022)
29. Wang, Y., Chen, X., You, Y., Li, L.E., Hariharan, B., Campbell, M., Weinberger, K.Q., Chao, W.L.: Train in germany, test in the usa: Making 3d object detectors generalize. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11713–11723 (2020)
30. Wei, Y., Wei, Z., Rao, Y., Li, J., Zhou, J., Lu, J.: Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. arXiv preprint arXiv:2203.14956 (2022)
31. Xu, Q., Zhou, Y., Wang, W., Qi, C.R., Anguelov, D.: Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15446–15456 (2021)
32. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
33. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d: Self-training for unsupervised domain adaptation on 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10368–10378 (2021)
34. Yang, J., Shi, S., Wang, Z., Li, H., Qi, X.: St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. IEEE transactions on pattern analysis and machine intelligence **45**(5), 6354–6371 (2022)
35. Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J.: Std: Sparse-to-dense 3d object detector for point cloud. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1951–1960 (2019)
36. Yin, T., Zhou, X., Krahenbuhl, P.: Center-based 3d object detection and tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11784–11793 (2021)
37. Yuan, J., Zhang, B., Yan, X., Chen, T., Shi, B., Li, Y., Qiao, Y.: Ad-pt: Autonomous driving pre-training with large-scale point cloud dataset. arXiv preprint arXiv:2306.00612 (2023)
38. Yuan, J., Zhang, B., Yan, X., Chen, T., Shi, B., Li, Y., Qiao, Y.: Bi3d: Bi-domain active learning for cross-domain 3d object detection. arXiv preprint arXiv:2303.05886 (2023)
39. Yue, X., Zhang, Y., Zhao, S., Sangiovanni-Vincentelli, A., Keutzer, K., Gong, B.: Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2100–2110 (2019)
40. Yue, X., Zheng, Z., Zhang, S., Gao, Y., Darrell, T., Keutzer, K., Vincentelli, A.S.: Prototypical cross-domain self-supervised learning for few-shot unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13834–13844 (2021)
41. Zhang, B., Chen, T., Wang, B., Li, R.: Joint distribution alignment via adversarial learning for domain adaptive object detection. IEEE Transactions on Multimedia (2021)
42. Zhang, B., Yuan, J., Shi, B., Chen, T., Li, Y., Qiao, Y.: Uni3d: A unified baseline for multi-dataset 3d object detection. arXiv preprint arXiv:2303.06880 (2023)

43. Zhang, Y., Hu, Q., Xu, G., Ma, Y., Wan, J., Guo, Y.: Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18953–18962 (2022)
44. Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H., Keutzer, K.: Multi-source domain adaptation for semantic segmentation. Advances in neural information processing systems **32** (2019)
45. Zheng, Y., Huang, D., Liu, S., Wang, Y.: Cross-domain object detection through coarse-to-fine feature adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13766–13775 (2020)
46. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)