# Test-Time Adaptation in 3D Object Detection Using Momentum-Based Pseudo-Labeling

Anonymous CVPR submission

Paper ID 15602

## Abstract

*Pre-trained 3D detection algorithms are typically trained on a large-scale dataset, however, it is infeasible to cover and perfectly handle every possible scene in the real world, underscoring the necessity of network adaptation. Test-time adaption dynamically adjusts models with online test-time stream data, thereby satisfying this demand. Thus, we introduce test-time adaptation to 3D object detection, leading to a novel problem setup named 3DTTA, and propose an approach, Momentum-based Pseudo-Labeling (MoPL). MoPL is inspired by the fact that adjacent frames have a strong temporal correlation in 3D object detection. Specifically, the same objects could exist in a series of consecutive frames, which is reliable information to identify pseudo-labels. MoPL exploits the temporal consistency to mine confident pseudo-labels and optimizes the model with them. MoPL is architecture-agnostic and can be applied to various detection models. We evaluate MoPL on Waymo, nuScenes, and Once datasets. MoPL achieves consistent performance gains and even outperforms 3D domain adaptation method ST3D by 19.41% $AP_{BEV}$ and 7.36% $AP_{3D}$ on nuScenes → Once.*

## 1. Introduction

3D object detection targets classifying and localizing the objects from 3D sensor data (*e.g*. LiDAR point clouds) in a scene, which plays an important role in autonomous driving [1]. Typically, the 3D object detection algorithm is trained on a central server and then dispatched to local devices (such as cars) for deployment. Though these algorithms demonstrate strong performance, they can not handle every encountered scene. In addition, when the testing data comes from a different distribution from the training one, these model usually suffers from performance degeneration [49, 54], an issue referred to as *domain shift*.

To remedy *domain shift* issue, some works [14, 43, 46, 50, 54] leverage domain adaptation to reduce the domain discrepancy (arising from varying types of 3D sensors, weather effects, and dissimilar object statistics, *etc*.) between the source and the target distributions. Domain adaptation exploits a labeled source domain (also referred to as the training domain) and an unlabeled target domain (also referred to as the testing domain) to transfer the knowledge learned from the source domain to target domain [23, 25, 31]. Domain generalization [13, 55, 56] is

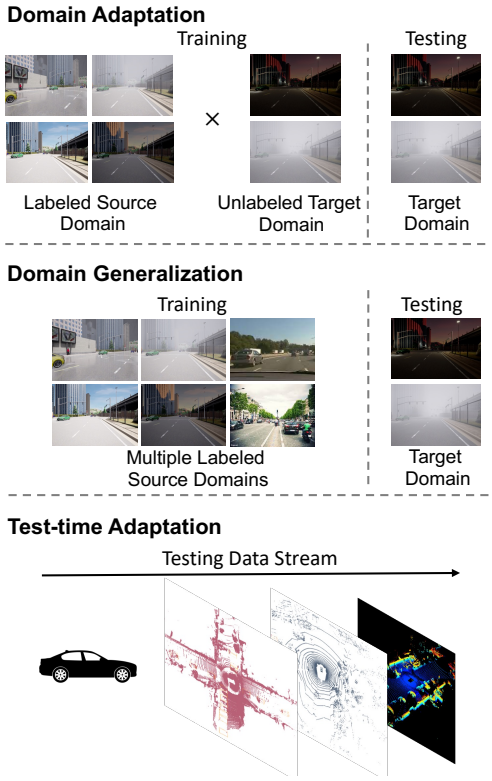


Figure 1. We introduce test-time adaptation (TTA) for 3D object detection and compare it with domain adaptation (DA) and domain generalization (DG) here. DA utilizes a labeled source domain and the unlabeled target domain to train a model that performs well on the target domain, while DG exploits several source domains to achieve the same goal. By contrast, TTA directly optimizes the model with online testing data streams.

another direction to enhance the generalization of models, which utilizes multiple source domains to simulate potential domain shifts. However, both domain adaptation and domain generalization can not solve continuously varying domains. Domain adaptation requires unlabeled test data during the training stage, which does not hold in many cases, while domain generalization struggles to simulate all possible real domain shifts.

In this work, we introduce test-time adaptation [41] to 3D object detection (3DOD). Test-time adaptation aims to address *domain shift* by adapting networks at the testing phase with online unlabeled test data streams, which does not rely on accessing unlabeled test data before the testing stage as domain adaptation. The intuitive comparison of domain adaptation, domain generalization, and test-time adaption is shown in Fig. 1. Prior research on test-time adaptation primarily focuses on developing and verifying their algorithms on 2D image recognition [2, 7, 9, 16, 17, 29, 38, 53]. Tent [41] uses entropy minimization as an objective to update the batch-normalization layers of neural networks. Sun et al. [38] employ an auxiliary self-supervised task to synergize the online training and the offline source training. These works effectively improve the model, but they do not consider how to adapt a detection model in 3DOD.

3D object detection involves two sub-tasks: classification and regression, which predict the category probability and 3D bounding box of objects, respectively. Self-training is an effective paradigm to simultaneously enhance the two sub-tasks [5]. Hence, we utilize pseudo-labels to update the parameters of the detection model. Naturally, this leads to a critical question: *how to acquire reliable pseudo labels in 3DOD?* One conventional way to select pseudo labels is to select confident predictions based on classification probability. Although effective, this strategy neglects 1) the inherent temporal information in 3DOD that is valuable and 2) the uncertainty of network prediction will cause inaccurate pseudo-labels. Temporal consistency commonly exists in consecutive frames where objects will move along their own trajectories until out of sensor range. This temporal clue can be exploited to identify reliable pseudo-labels. For the latter, networks will occasionally predict false-positive or false-negative results, leading to noisy pseudo-labels. Therefore, the temporal information can be used to augment pseudo-labels.

To this end, we present a Momentum-based Pseudo-Labeling (MoPL) for test-time adaptation. MoPL mine reliable pseudo-labels across consecutive frames, where only temporal-consistent pseudo-labels will be selected. Specifically, MoPL adopts a backtracking strategy to build trajectories of objects, which will be used to estimate the velocities of objects. Then, MoPL infers the next-frame bounding boxes with the estimated trajectories and velocities based on a consistent velocity model. The predicted bounding boxes are leveraged as pseudo-labels to optimize the detection model during test time. In addition, to prevent catastrophic forgetting of source knowledge, we exploit the mean-teacher framework [39]. The teacher model is EMA-updated and generates pseudo-labels for learning the student model. In addition, we update the detection heads of detection models instead of only batch-normalization layers as [41], since 3DOD is a more sophisticated task and only updates on batch-normalization layers are inadequate to adapt the detection model.

In summary, Our main contributions are as follows:

- We introduce a new and practical setup, 3DTTA that performs test-time adaptation for 3DOD to mitigate *domain shift* and enhance model performance.
- We introduce a novel momentum-based pseudo-labeling (MoPL) approach that leverages the inherent temporal consistency in consecutive point cloud frames to identify reliable pseudo-labels for self-training.
- We curate three cross-domain benchmarks to evaluate MoPL, on which MoPL achieves consistent improvement over the baseline model. Furthermore, MoPL outperforms the 3D domain adaptation method ST3D by 19.41% $AP_{BEV}$ and 7.36% $AP_{3D}$ on NuScenes $\rightarrow$ ONCE.

## 2. Related Work

### 2.1. LiDAR-based 3D Object Detection

The goal of LiDAR-based 3D object detection is to classify and localize 3D objects from point clouds [4, 8, 22, 32, 34–36, 47, 48, 50, 50, 57], which is challenging and has attracted a surge of interest in real-world applications, *e.g.*, autonomous driving and robotics. Some prior methods [4, 20, 48] project 3D point clouds to 2D bird's-eye-view (BEV) feature maps so that they can borrow the 2D detection methods to solve 3D object detection. Some other methods [15, 35, 36, 47, 57] model the 3D point clouds as voxels, a regular data format, and then use the 3D convolution network to extract features volumes. In this work, we adopt PV-RCNN and SECOND as the baseline detection, as in [49], to adapt the detection models to novel scenes during test time.

### 2.2. Domain Adaptation for 3D Point Clouds

Recently, some researchers have leveraged domain adaptation techniques to transfer the learned knowledge from the source to the target domain for 3D point clouds [5, 43]. To name a few, ST3D [49] leverages curriculum data augmentation strategy to yield pseudo-labels for self-training, thus reducing the domain gap. LiDAR Distillation [44] tries to distill the knowledge from high-beam point clouds to low-beam point clouds. Besides, some works are focusing on aligning object scales and ranges [54], mean-teacher paradigm [27], and contrastive learning [51]. Bi3D [52] se-

lects partial-yet-important target samples to achieve a good trade-off between high performance and low annotation cost. Nonetheless, these works all assume that the unlabeled target data is accessible before testing, which does not hold true in practice. LiDAR-UDA also utilizes temporal information to find the connection among intra-class masks across frames. MoPL from LiDAR-UDA in that 1) we focus on 3D detection task. 2) we leverage temporal consistency to identify the trajectory of each individual object and then obtain reliable pseud-labels. and 3) we perform model adaption during test time.

### 2.3. Test-Time Adaptation

Test-time adaptation (TTA) is a more challenging setup where only the source model and unlabeled target/testing data are available [6, 10, 18, 19, 21, 26, 33, 41]. TTA methods adapt models during test-time and can not rectify the predictions of previous testing data. In particular, Tent [41] explores the role of batch-normalization layers in TTA, which uses entropy minimization [12] as the training objective. SAR [30] improves entropy minimization in TTA via filtering out noisy samples with larger gradients, while COTTA [42] focuses on continually adapting networks to varying target distributions. Though effective, these approaches are for 2D images while the development of TTA in 3D object detection lags behind. In this work, we introduce a TTA method named MoPL for the 3DOD field, aiming to improve the performance of 3D detections during test time.

## 3. Method

### 3.1. Problem Definition and Baseline Selection

**Problem Definition**. Given a labeled source domain $D_s = \{\boldsymbol{x}_i^s, y_i^s\}_{i=1}^{N_s}$, where $\boldsymbol{x}_i^s$ is the input point cloud frame and $y_i^s$ is the corresponding label (including category label and 3D bounding box coordinates), one can train a source model with $D_s$ using typical supervised learning. Besides this, we have a target domain that is also the test domain, $D_t = \{\boldsymbol{x}_i^t\}_{i=1}^{N_t}$.

$D_t$ is fully unlabeled and comes from a different distribution with $D_s$, which will cause the *domain shift* issue. Domain adaptation (DA) assumes that $D_t$ is accessible during training, and DA methods leverage target data to reduce domain discrepancy for transferring source knowledge to the target domain. On the other hand, domain generalization (DG) removes the need for target data during the training phase and exploits multiple source domains to train a robust model. Neither can meet the practical need, since 1) the target data is inaccessible in the training stage (disadvantage of DA) and 2) the domain shift is unpredictable and the target domain could vary significantly from user to user (disadvantage of DG).

To tackle this, we introduce a novel setup 3DTTA, which performs test-time adaptation (TTA) for 3DOD. In 3DTTA, we only access the source model pre-trained on $D_s$, and the unlabeled target data stream. The goal of 3DTTA is to online adapt the detection model during test-time.

### 3.2. Momentum-based Pseudo-Labeling

Given that test-time data is unlabeled, pseudo-labeling is a straightforward means to generate supervision signals. We introduce a novel Momentum-based Pseudo-Labeling (MoPL) approach to exploit the temporal clues of recently consecutive frames to mine reliable pseudo-labels, which is complimentary to conventional confidence-based pseudo-labeling. The overview of our method is in Fig. 2 and we introduce details in the following.

**Confidence-based Pseudo-Labeling**. Prediction confidence is commonly adopted as a measurement to select pseudo-labels [24]. In 2D image classification, the prediction confidence is formulated as the maximal prediction probability of each image, while in 3D object detection, the prediction confidence is formulated as the maximal prediction probability of each 3D bounding box.

Formally, let $c$ denote the maximal prediction probability of a bounding box, we utilize a threshold $\beta$ to filter out those low-confident bounding boxes, leading to a pseudo-label set:

$$P_c = \{(b, c) | c \geq \beta, (b, c) \in \hat{y}^t\}, \tag{1}$$

where $\hat{y}^t$ is the set of predicted boxes and $b$ are the coordinates of detection boxes. The selected pseudo-labels are used as the surrogate of ground-truth labels to update the detection model. This strategy is based on the assumption that a detection network relies on bounding box coordinates to aggregate local features to perform classification, therefore high prediction confidence indicates high regression confidence to some extent. However, the potential false-positive and false-negative predictions limit the performance of this pseudo-labeling strategy.

**Momentum-based Pseudo-Labeling** can solve the above challenge via exploiting temporal consistency across frames to filter out noisy pseudo-labels. We consider the major application of 3DOD, autonomous driving. In this scenario, the consecutive frames have a temporal correlation. For example, vehicles in the current frame will still exist in the next frame with high probability. Based on this observation, we can predict the next-frame object boxes based on recent object states gathered from recent frames, which is similar to the spirit of object tracking [45]. Then, those predicted boxes can be used as pseudo-labels to supervise the model's adaptation phase.

Formally, we define $\boldsymbol{x}_i^t$ as $i^{th}$ input point cloud frame. During practical deployment/testing, the detection model

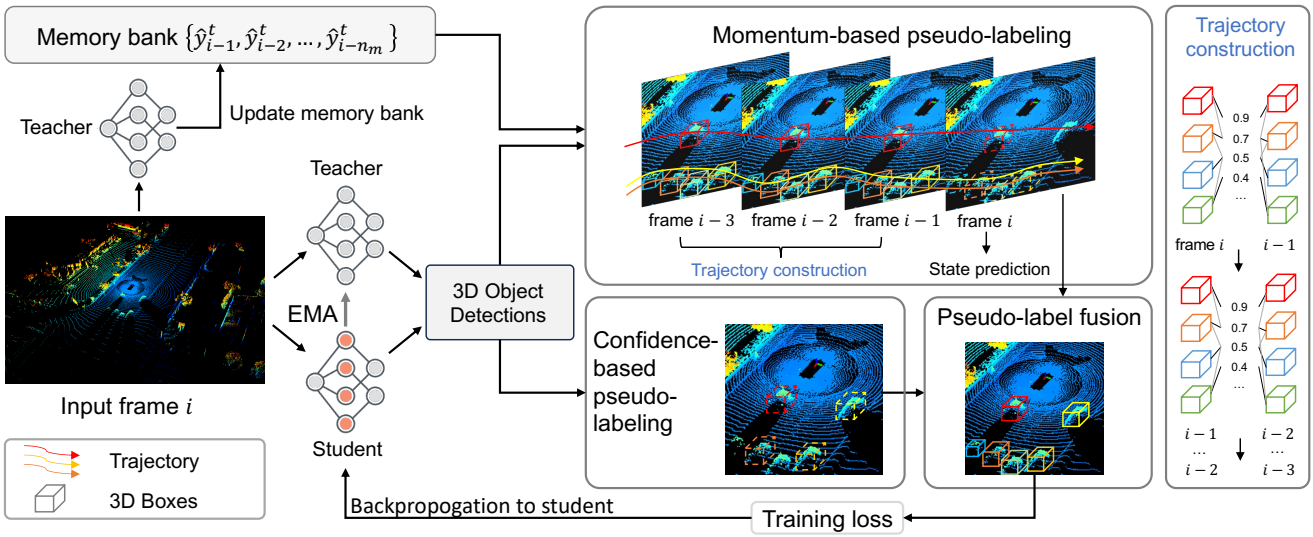


Figure 2. **Framework Overview**. Firstly, we adopt the mean-teacher framework to optimize models for stabilizing training and avoid catastrophic forgetting. The teacher model uses the EMA weights of the student model. During the inference of test-time data stream in 3D object detection, we utilize a memory bank to store the predictions of each frame. The memory bank and the current input frame are integrated to construct object trajectories. Finally, we fuse the pseudo-labels generated by trajectory propagation and confidence-based filtering to train the student model.

will continually receive data. Obviously, we need multiple frames to extract and leverage the temporal information. Thus, we establish a memory bank to store the necessary features of past frames. We first define the predicted bounding box set as $\hat{y}^t_i$, which contains multiple bounding boxes. Then the memory bank $\mathcal{M}$ is formulated as:

$$\mathcal{M} = \{\hat{y}^t_{i-1}, \hat{y}^t_{i-2}, ...\hat{y}^t_{i-n_m}\}. \tag{2}$$

where $i$ indicates current input frame and $n_m$ is the volume of memory bank $\mathcal{M}$. $\mathcal{M}$ only stores the prediction results of recent past frames, for saving the memory cost. We set the size of $\mathcal{M}$ as 3 in our implementation.

**Trajectory State Estimation**. For frame $i$, we build its trajectories $T_i$ by aggregating the predicted boxes of previous frames in $\mathcal{M}$ with frame $i$. Then we predict the state of trajectories $T_i$ to current frame $i$ as $T_{est}$ (we introduce how to construct trajectory in the next subsection). In detail, we define the state of a trajectory as $T = (\mathtt{x}, \mathtt{y}, \mathtt{z}, \theta, \mathtt{w}, \mathtt{h}, \mathtt{l}, \mathtt{v}_x, \mathtt{v}_y, \mathtt{v}_z)$, a 10-dimensional vector, where $\mathtt{x}, \mathtt{y}, \mathtt{z}$ denote the 3D spatial location of the center point of an object and $\mathtt{w}, \mathtt{h}, \mathtt{l}$ denote the width, height, and length of the corresponding bounding box, $\theta$ denotes the heading angle of object in 3D space, $\mathtt{v}_x, \mathtt{v}_y, \mathtt{v}_z$ indicate the velocity of object in 3D space. Note that we abuse notations $x, y$ here to denote spatial coordinates, we separate data $x$ and coordinate $\mathtt{x}$ with different formats.

At each frame, all the associated trajectories $T_{i-1}$ will be utilized to estimate the states at frame $i$ with the constant velocity model:

$$\begin{aligned} \mathtt{x}_{est} &= \mathtt{x} + \mathtt{v}_x \\ \mathtt{y}_{est} &= \mathtt{y} + \mathtt{v}_y \\ \mathtt{z}_{est} &= \mathtt{z} + \mathtt{z}_z, \end{aligned} \tag{3}$$

where $\mathtt{x}_{est}, \mathtt{y}_{est}, \mathtt{z}_{est}$ are the estimated spatial coordinates of frame $i$. Each trajectory in $T_i = \{T^1_i, T^2_{i-1}, ..., T^{m_t}_{i-1}\}$ ($m_t$ is the number of trajectories) will derive an estimated state at frame $i$, leading to a $T^j_{est} = (\mathtt{x}_{est}, \mathtt{y}_{est}, \mathtt{z}_{est}, \mathtt{w}, \mathtt{h}, \mathtt{l}, \theta, \mathtt{v}_x, \mathtt{v}_y, \mathtt{v}_z) \in T_{est}$. We remove the three velocity features $\mathtt{v}_x, \mathtt{v}_y, \mathtt{v}_z$ in $T_{est}$ and the rest are utilized as pseudo-labels, denoted as $P_m$.

**Trajectory Construction**. We construct trajectories to estimate pseudo-labels $P_m$. Considering test-time data is steaming instead of being static, we introduce a backtracking strategy that starts from the current frame $i$ to recent past frames. In detail, at frame $i$, let's assume that we have $n_i$ detection boxes at hand, we use these predicted boxes to query the $n_{i-1}$ predictions of frame $i-1$ (stored in the memory bank $\mathcal{M}$). Specifically, we calculate an affinity matrix with size $n_i \times n_{i-1}$ with a similarity measurement (*e.g.*, 3D IoU). We rank those similarities in a descent order and select the detection pairs with the highest similarity. Furthermore, we only consider the detection pairs whose prediction confidences exceed a threshold $\lambda$. In the next step, we use the selected predicted boxes of frame $i-1$ to query the predicted boxes of frame $i-2$ and repeat the above operations. In total, we query 3 past frames to build trajectories $T_i$.

Once the trajectories are built, we can obtain the velocities of $T_i$:

$$\begin{aligned} \mathrm{v}_x^t &= (\mathrm{x}^t - \mathrm{x}^{t-3})/3, \\ \mathrm{v}_y^t &= (\mathrm{y}^t - \mathrm{y}^{t-3})/3, \\ \mathrm{v}_z^t &= (\mathrm{z}^t - \mathrm{z}^{t-3})/3. \end{aligned} \tag{4}$$

Note that we assume that the objects are moving at a consistent velocity. Then we can predict the object state of the next frame using velocity estimated in Eq. (5).

Compared to the 3D Kalman filter [45] that adopts a feedforward strategy to derive trajectories and next-frame prediction, our backtracking strategy fits more the nature of test-time adaptation. We use a memory bank to store recent frames instead of trajectories, therefore we can update the trajectory each time new data arrives, removing the disappeared objects and adding new-coming objects.

### 3.3. Model Training with MoPL

**Pseudo-label Fusion**. In the previous section, we introduce two pseudo-labeling schemes. We fuse the pseudo-labels generated from the two schemes to train our detection model. The final pseudo-label set we use is defined as $P = P_c \cup P_m$. The two strategies are complementary to each other in that confidence-based pseudo-labeling discovers the newly entered object while momentum-based pseudo-labeling improves the reliability through temporal consistency.

**Random Object Scaling (ROS)**. As validated in [54], object scale discrepancy contributes to *domain shfit*. To remedy this, we introduce a random object scaling (ROS) strategy. In specific, after we obtain a pseudo-label denoted as $(\mathrm{x}, \mathrm{y}, \mathrm{z}, \theta, \mathrm{w}, \mathrm{h}, \mathrm{l})$. ROS randomly expands or shrinks the box size with random scale factor $(r_w, r_h, r_l)$. The scaled box is $(\mathrm{x}, \mathrm{y}, \mathrm{z}, \theta, r_w\mathrm{w}, r_h\mathrm{h}, r_l\mathrm{l})$. We do not change the coordinates of the center point $(\mathrm{x}, \mathrm{y}, \mathrm{z})$ and its heading angle $\theta$. ST3D [49] also introduces a random object scaling strategy as data augmentation. We would like to highlight that our ROS is used at the test-time stage while ST3D uses ROS in the source pre-training stage. We wrap the pseudo-labels instead of source ground-truth boxes as we cannot access source data. The ROS serves as a lightweight tool to enhance the robustness of our test-time model against object statistics bias.

**Mean-Teacher Framework**. Test-time adaptation methods are more fragile than supervised methods in parameter updation due to the lack of ground-truth labels. Therefore, we leverage mean-teacher [39] to stabilize the training by involving a teacher model and a student model. The teacher uses the EMA weights of the student model:

$$\Theta'_k = \alpha\Theta'_{k-1} + (1-\alpha)\Theta_{k-1}, \tag{5}$$

where $\Theta$ is the parameters of student model and $\Theta'$ is the parameters of teacher model. The weights of the teacher model are ensembled on the temporal axis. Thus, the teacher model is more robust against the student model and we use the teacher model to yield pseudo labels for optimizing the student model. In addition, using the averaged weights of EMA can avoid catastrophic forgetting of source knowledge.

## 4. Experiment

In this section, we evaluate our proposed MoPL on the 3DTTA setup. We leverage three LiDAR-based 3DOD datasets, *i.e.*, Waymo [37], nuScenes [3], and Once [28], to build three TTA benchmarks. We compare our method with existing TTA methods for 2D vision and DA methods for 3DOD. Furthermore, we conduct extensive analytical experiments to verify our design.

### 4.1. Setup

**Datasets.** Waymo [37] is composed of high-resolution sensor data collected by autonomous vehicles and it contains 1,950 20-second segments. NuScenes [3] includes bounding boxes of 1,000 scenes collected in Boston and Singapore. Each scene is 20 seconds long. Once [28] is a recently released large-scale dataset that contains 1 million LiDAR scenes. The details of these three datasets are shown in Tabel. 1. We build three test-time adaptation tasks: Waymo→nuScenes, nuScenes→Waymo, and nuScenes→Once. We leverage the validation sets of these datasets as the target/test domains.

**Implementation Details.** Our proposed MoPL is evaluated with two detection backbones, SECOND [47] and PV-RCNN [35] following [49]. As [49], we augment SECOND with an IoU head to form SECOND-IoU for a fair comparison. We implement our code based on open-source codebase 3DTrans [52] and OpenPCDet [40]. The hyper-parameters $\beta$ and $\lambda$ of confidence-based pseudo-labeling and momentum-based pseudo-labeling are both set as 0.5 in our experiments. The random scale factors, $r_w, r_h, r_l$ are uniformly sampled from $[0.95, 1.05]$. The coef $\alpha$ of Exponential Moving Average (EMA) is set at 0.999, as recommended in [39]. The size of the memory bank is configured to 4, which is pretty lightweight and meets our needs. We run all experiments on a single Nvidia 3090 GPU.

We adjust the parameters of detection heads of SECOND-IoU and PV-RCNN while freezing the feature extraction network. In contrast to Tent[41] and SAR[30], we deactivate the tracking of running statistics in batch-normalization layers during the adaptation phase, which empirically leads to better performance.

| Dataset | Beam | VFOV | Location | Night Time | Rainy Weather | Object Types |
|---|---|---|---|---|---|---|
| Waymo[37] | 64 | [-17.6°, 2.4° ] | USA | Yes | Yes | 4 |
| nuScenes[3] | 32 | [-30.0°, 10.0°] | USA & Singapore | Yes | Yes | 23 |
| Once[28] | 40 | [-25.0°, 15.0°] | China | Yes | Yes | 5 |

Table 1. Dataset overview and detailed parameters of LiDAR.

| Task | Setting | Method | SECOND-IoU | | PV-RCNN | |
|---|---|---|---|---|---|---|
| | | | $AP_{BEV}$ / $AP_{3D}$ | Closed Gap | $AP_{BEV}$ / $AP_{3D}$ | Closed Gap |
| Waymo → nuScenes | - | Source Only | 32.84 / 17.24 | - | 34.21 / 21.36 | - |
| | DA | SN [43] | 33.23 / 18.57 | 11.21% / 40.43% | 34.22 / 22.29 | 81.52% / 29.94% |
| | | ST3D [49] | **35.92 / 20.19** | 88.51% / 89.66% | 36.42 / **22.99** | 92.08% / 47.38% |
| | TTA | Tent [41] | 31.56 / 14.63 | -36.78% / -79.33% | 31.24 / 16.72 | -76.94% / -134.88% |
| | | SAR [30] | 33.32 / 15.83 | 13.79% / -42.86% | 34.70 / 17.62 | 83.82% / -108.72% |
| | | **MoPL (ours)** | 34.89 / 17.69 | 58.91% / 13.47% | **36.89** / 21.89 | 94.34% / 15.41% |
| | - | Oracle | 36.32 / 20.53 | - | 38.07 / 24.80 | - |
| nuScenes → Waymo | - | Source Only | 25.44 / 9.79 | - | 23.97 / 17.40 | - |
| | DA | SN [43] | 35.31 / 10.58 | 76.10% / 16.77% | 40.97 / 17.83 | 77.23% / 4.08% |
| | | ST3D [49] | 37.98 / 13.40 | 96.68% / 76.65% | 43.86 / 22.33 | 90.37% / 46.82% |
| | TTA | Tent [41] | 20.04 / 7.48 | -41.63% / -49.04% | 24.70 / 14.41 | 3.32% / -28.40% |
| | | SAR [30] | 20.27 / 8.71 | -39.86% / -22.93% | 24.09 / 15.17 | 0.55% / -21.18% |
| | | **MoPL (ours)** | **38.37 / 14.29** | 99.69% / 95.54% | **45.19 / 22.59** | 96.41% / 49.29% |
| | - | Oracle | 38.41 / 14.50 | - | 45.98 / 27.93 | - |
| nuScenes → Once | - | Source Only | 47.35 / 17.93 | - | 48.61 / 28.13 | - |
| | DA | SN [43] | 63.07 / 21.42 | 93.24% / 71.81% | 61.89 / 33.82 | 89.01% / 78.05% |
| | | ST3D [49] | 44.76 / 15.34 | -15.36% / -53.29 | 45.93 / 27.83 | -18.03% / -4.12% |
| | TTA | Tent [41] | 46.28 / 18.76 | -6.35% / 17.08% | 52.52 / 30.07 | 26.21% / 26.61% |
| | | SAR [30] | 50.63 / 20.02 | 19.45% / 43.00% | 52.53 / 30.05 | 26.27% / 26.34% |
| | | **MoPL (ours)** | **64.17 / 22.70** | 99.76% / 98.15% | **62.20 / 34.99** | 91.09% / 94.10% |
| | - | Oracle | 64.21 / 22.79 | - | 63.53 / 35.42 | - |

Table 2. **Adaptation results of 3D object detection on multiple datasets**. We report $AP_{BEV}$ and $AP_{3D}$ of the car category at IoU = 0.7 over 40 positions' recall, with their domain gap to Oracle. The reported results are evaluated by KITTI's[11] evaluation metric. We indicate the best adaptation results by **bold** fonts. Following [49], we report the close gap $= (AP_{method} - AP_{source\ only})/(AP_{oracle} - AP_{method}) \times 100\%$.

| Method | $AP_{BEV}$ / $AP_{3D}$ |
|---|---|
| Source only (baseline) | 23.97 / 17.40 |
| +Conf. PL | 43.05 / 21.63 |
| +Mo. PL | 43.79 / 21.15 |
| +Conf. PL + ROS | 43.11 / 21.69 |
| +Mo. PL + ROS | 44.56 / 22.08 |
| +Mo. PL + ROS + Mean teacher | 44.85 / 22.37 |
| MoPL | **45.19 / 22.59** |

Table 3. **Ablation study.** These experiment is conducted on nuScenes → Waymo with PV-RCNN. Conf. PL denotes confidence-based pseudo-labeling and Mo. PL denotes momentum-based pseudo-labeling strategy not the full method. ROS is a random object object. MoPL indicates the full method.

### 4.2. Results

We report the experiment results on three cross-domain tasks in Table 2. We divide the table into three sections according to tasks. In each task section, we roughly categorize all the methods into four groups: the source-only model, domain adaptation methods, test-time adaptation methods, and oracle. Source only is trained with the source labeled data and frozen once finishing training. Domain adaptation methods leverage the prior target information (SN) or unlabeled target data (ST3D) to train the detection model. Test-time adaptation methods utilize online target data streams to update the source-pretrained model. Since there are no test-time adaptation methods in 3D object detection, we adapt the 2D TTA methods to 3DOD. The last group is Oracle which trains the model with ground-truth labels in TTA fashion.

We can observe that our proposed MoPL consistently improves the model performance over the baseline model, *i.e.*, the source-only model, whose weights are leveraged to initialize our model in the beginning. This result manifests that MoPL can effectively enhance model generalization against various *domain shift* issues.

In addition, our MoPL demonstrates superior performance over domain adaptation methods, SN, and ST3D. SN reduces the domain gap by augmenting training bounding

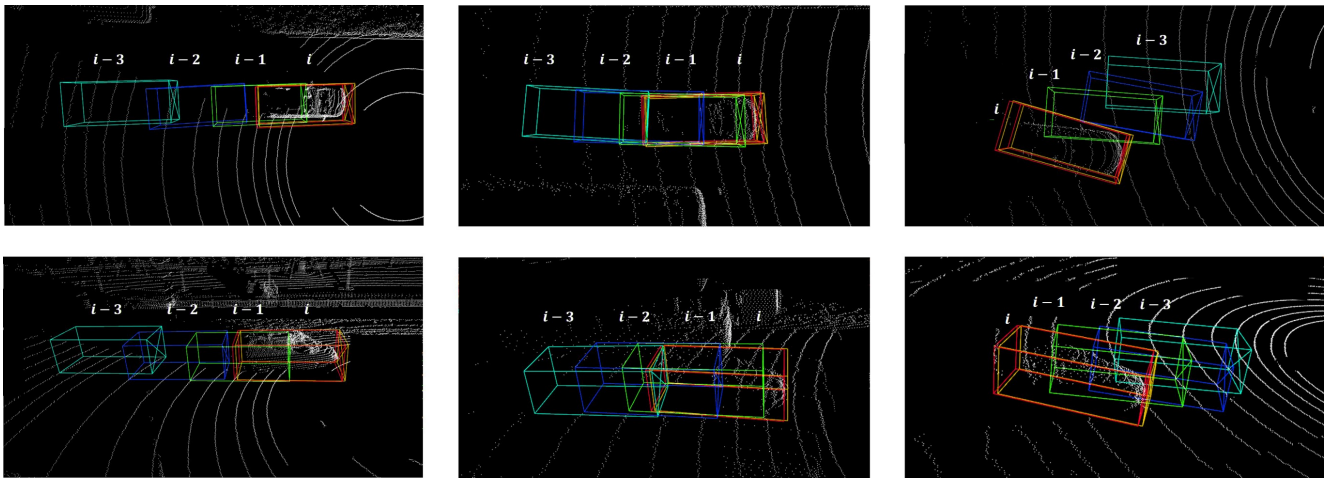


Figure 3. **Visualization of momentum-based pseudo-labeling**. The first and second rows are the car's trajectories from different views. frame $i-3$, frame $i-2$, frame $i-1$ indicate the previous trajectory and frame $i$ is the predicted pseudo-label by our momentum-based pseudo-labeling, which is extremely close to the ground truth box.

| Method | $AP_{BEV}$ / $AP_{3D}$ |
|---|---|
| Source only (baseline) | 34.21 / 21.36 |
| Fully training | 19.46 / 13.16 |
| Feature extractor | 22.00 / 14.78 |
| Detection heads | **36.89 / 21.89** |

Table 4. **Effect of updating schemes.** Experiments are conducted on Waymo → nuScenes with PV-RCNN. Fully training indicates updating all the parameters of the detection model. Feature extractor denotes updating the feature extraction network of the detection model. Detection heads denote updating all the detection heads of the detection model.

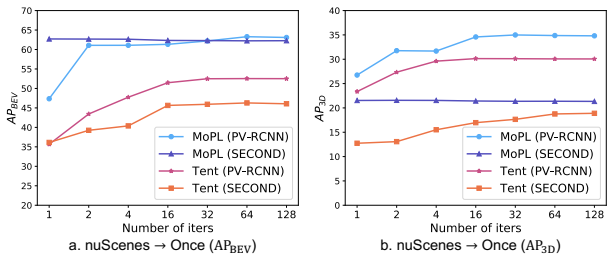


Figure 4. Effect of accumulation iteration on Tent and Mo .PL. We evaluate model performance with varying numbers of accumulation iterations.

boxes with statistical box information of the target domain, while ST3D trains the network with curriculum data augmentation and pseudo-labels selected using the estimated IoU score that measures the regression confidence. In specific, MoPL exceeds ST3D by 19.41% $AP_{BEV}$ and 7.36% $AP_{3D}$ on nuScenes → Once (with SECOND-IoU).

Furthermore, compared with 2D TTA methods such as Tent and SAR, our method outperforms them by large margins. The reason is that Tent and SAR put attention on improving classification performance while neglecting the regression branch that is vital for 3D object detection. Therefore, our method exceeds them by large margins since our method considers both classification and regression.

In many cases, MoPL achieves a close performance with the oracle, which further manifests the effectiveness of our method.

### 4.3. Analytical Experiment

**Ablation study.** We perform an ablation study on the major components of our method by evaluating the efficacy of each method in Table 3. We run ablation experiments on nuScenes → Waymo task using PV-RCNN. We begin with the basic baseline, source-only model. We add confidence-based pseudo-labeling to the source model, which achieves obvious improvement. This demonstrates the effectiveness of pseudo-labeling in the test-time adaption of 3D object detection. Then, we add momentum-based pseudo-labeling to the source model, which performs better than confidence-based pseudo-labeling. This result manifests that temporal information helps to identify more effective pseudo-labels.

After that, we add random object scaling (ROS) to both the pseudo-labeling strategies and witness extra gains over them, demonstrating that scaling object sizes can reduce the domain bias in object sizes and make the model more robust. Furthermore, we observe that the mean teacher enhances model performance. The teacher model is steadily updated with EMA weights, thus providing more stable pseudo labels for self-training. Finally, we unify the confidence-based and momentum-based pseudo-labeling strategies, which leads to our full method. We observe further gains, demonstrating that momentum-based pseudo-labeling is compatible with other pseudo-labeling

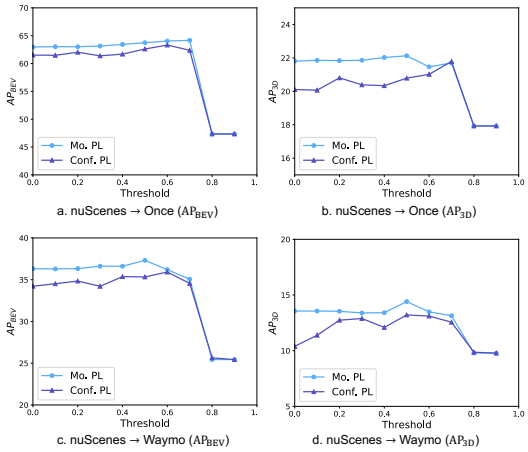


Figure 5. Effect of the threshold value of momentum-based pseudo-labeling (Mo. PL) and confidence-based pseudo-labeling (Conf. PL).

strategies, leaving more room for improvement in the future.

**Effect of updating schemes.** Test-time adaptation optimizes the parameters of the detection model with the online testing data stream. Recall that we initialize the detection model with weights of the source-only model. A critical issue to avoid in training is catastrophic learning. The source knowledge from the source model is important to recognize target objects and obtain reliable pseudo-labels. We empirically observe that if we optimize all the parameters of the detection model, the performance will decrease to the worst as shown in Table 4. Therefore, in order to stabilize the training and preserve the source knowledge during training, we only update some modules of the detection model. We evaluate the performance of different updating schemes and find that it's best to fix the feature extraction module and update all the detection heads, *e.g.*, the dense head, point head, and RoI head of PV-RCNN. A potential explanation is that the encoder contains low-level source knowledge which is vital to perceive the basis patterns of point clouds and is also sharable with testing data, while the detection heads are more biased to the specific characteristic of the source domain, *e.g.*, object statistics. Hence, it's more effective to reduce the domain bias in detection heads.

**Effect of accumulation iteration.** Processing a 3D point cloud costs more GPU memory than a 2D image, leading to a small training batch size. On some consumer-grade GPUs, the batch size can only be set as 1. However, a larger batch size can generate more stable gradients, which can accelerate convergence and potentially improve performance. A method to solve this dilemma is to accumulate the gradients of multiple samples and then average them for updating. We experiment with the effect of this strategy on TTA methods as shown in Fig. 4. We can observe that Tent is sensitive

| Model | Method | $AP_{BEV}$ / $AP_{3D}$ |
|---|---|---|
| 1% annotation budget | Bi3D [52] | 42.06 / 26.33 |
| | MoPL | **46.21 / 31.13** |
| 5% annotation budget | Bi3D [52] | 47.84 / 32.02 |
| | MoPL | **48.69 / 32.81** |

Table 5. Generalization of MoPL to Active learning model on Waymo → nuScenes.

to the number of accumulation iterations while our MoPL has a robust and stable performance when the number of accumulation iteration is greater than 2. This strength further enhances the application of our method in real-world deployment environments.

**Effect of thresholds $\beta$ and $\lambda$.** We leverage $\beta$ to filter out noisy pseudo-labels in confidence-based pseudo-labeling and $\lambda$ to filter out low-confident matched detection pairs. We evaluate the effect of the two hyperparameters in Fig. 5. We observe that our MoPL reaches a plateau at the threshold value $\in [0, 0.7]$. The robustness of MoPL is beneficial to obtain satisfactory performance.

**Generalization of MoPL to active learning.** In practical scenarios, we might utilize other techniques such as active learning [52] to enhance model performance when facing *domain shift*. We test the generalization ability of our method in active learning PV-RCNN model [52], as shown in Table 5. Active learning uses a small portion (1% or 5%) of labeled target data to train the model, thus leading to significantly better performance against the source-only model. We observe that our MoPL can further boost the performance, validating the generalization ability of our method.

**Visualization of momentum-based pseudo-labeling.** To intuitively understand our method, we visualize the predicted pseudo-label boxes and corresponding trajectories as shown in Fig. 3. The $i-3, i-2, i-1$ bounding boxes indicate the trajectory of an object. Then we predict the pseudo-labels at frame $i$ according to Eq. (4). One can observe that the pseudo-labels are extremely close to the ground truth boxes, demonstrating the effectiveness of utilizing temporal consistency to mine pseudo-labels.

## 5. Conclusion

In this paper, we introduce a new setup 3D TTA that leverages an online test-time data stream to adapt the detection model, and a novel momentum-based pseudo-labeling approach to exploit the temporal consistency of consecutive frames to mine reliable pseudo-labels which achieved striking performance on three cross-domain benchmarks. In future work, we will explore the role of heading angles in pseudo-labeling.

# References

[1] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019. 1

[2] Mathilde Bateson, Herve Lombaert, and Ismail Ben Ayed. Test-time adaptation with shape moments for image segmentation. In *MICCAI*, pages 736–745, 2022. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 5, 6

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 2

[5] Zhuoxiao Chen, Yadan Luo, Zheng Wang, Mahsa Baktashmotlagh, and Zi Huang. Revisiting domain-adaptive 3d object detection by reliable, diverse and class-balanced pseudo-labeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3714–3726, 2023. 2

[6] Zhixiang Chi, Yang Wang, Yuanhao Yu, and Jin Tang. Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning. In *CVPR*, pages 9137–9146, 2021. 3

[7] Sungha Choi, Seunghan Yang, Seokeon Choi, and Sungrack Yun. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. In *ECCV*, pages 440–458, 2022. 2

[8] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1201–1209, 2021. 2

[9] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei A Efros. Test-time training with masked autoencoders. In *NeurIPS*, 2022. 2

[10] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N. Metaxas. Visual prompt tuning for test-time domain adaptation. *CoRR*, abs/2210.04831, 2022. 3

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition*, 2012. 6

[12] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *NeurIPS*, pages 529–536, 2004. 3

[13] Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. *arXiv preprint arXiv:2007.01434*, 2020. 1

[14] Martin Hahner, Christos Sakaridis, Mario Bijelic, Felix Heide, Fisher Yu, Dengxin Dai, and Luc Van Gool. Lidar snowfall simulation for robust 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16364–16374, 2022. 1

[15] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11873–11882, 2020. 2

[16] Hengguan Huang, Xiangming Gu, Hao Wang, Chang Xiao, Hongfu Liu, and Ye Wang. Extrapolative continuous-time bayesian neural network for fast training-free test-time adaptation. In *NeurIPS*, 2022. 2

[17] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In *NeurIPS*, pages 2427–2440, 2021. 2

[18] Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In *CVPR*, pages 577–584, 2011. 3

[19] Minguk Jang and Sae-Young Chung. Test-time adaptation via self-training with nearest neighbor information. *CoRR*, abs/2207.10792, 2022. 3

[20] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 2

[21] Jogendra Nath Kundu, Naveen Venkat, Rahul M. V., and R. Venkatesh Babu. Universal source-free domain adaptation. In *CVPR*, pages 4543–4552, 2020. 3

[22] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 2

[23] Shuang Li, Mixue Xie, Kaixiong Gong, Chi Harold Liu, Yulin Wang, and Wei Li. Transferable semantic augmentation for domain adaptation. In *CVPR*, pages 11516–11525, 2021. 1

[24] Jian Liang, Ran He, Zhenan Sun, and Tieniu Tan. Exploring uncertainty in pseudo-label guided unsupervised domain adaptation. *Pattern Recognition*, 96:106996, 2019. 3

[25] Shikun Liu, Tianchun Li, Yongbin Feng, Nhan Tran, Han Zhao, Qiang Qiu, and Pan Li. Structural re-weighting improves graph domain adaptation. In *International Conference on Machine Learning*, pages 21778–21793. PMLR, 2023. 1

[26] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. TTT++: when does self-supervised test-time training fail or thrive? In *NeurIPS*, pages 21808–21820, 2021. 3

[27] Zhipeng Luo, Zhongang Cai, Changqing Zhou, Gongjie Zhang, Haiyu Zhao, Shuai Yi, Shijian Lu, Hongsheng Li, Shanghang Zhang, and Ziwei Liu. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8866–8875, 2021. 2

[28] Jiageng Mao, Minzhe Niu, Chenhan Jiang, Xiaodan Liang, Yamin Li, Chaoqiang Ye, Wei Zhang, Zhenguo Li, Jie Yu,

Chunjing Xu, et al. One million scenes for autonomous driving: Once dataset. 2021. 5, 6

[29] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *ICML*, pages 16888–16905, 2022. 2

[30] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023. 3, 5, 6

[31] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010. 1

[32] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 2

[33] Amelie Royer and Christoph H Lampert. Classifier adaptation at prediction time. In *CVPR*, pages 1401–1409, 2015. 3

[34] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 2

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 2, 5

[36] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020. 2

[37] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset, 2020. 5, 6

[38] Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, pages 9229–9248, 2020. 2

[39] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 2, 5

[40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 5

[41] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021. 2, 3, 5, 6

[42] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, pages 7191–7201, 2022. 3

[43] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 1, 2, 6

[44] Yi Wei, Zibu Wei, Yongming Rao, Jiaxin Li, Jie Zhou, and Jiwen Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. *arXiv preprint arXiv:2203.14956*, 2022. 2

[45] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 1(2):6, 2019. 3, 5

[46] Qiangeng Xu, Yin Zhou, Weiyue Wang, Charles R Qi, and Dragomir Anguelov. Spg: Unsupervised domain adaptation for 3d object detection via semantic point generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15446–15456, 2021. 1

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

[48] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[49] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10368–10378, 2021. 1, 2, 5, 6

[50] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019. 1, 2

[51] Zeng Yihan, Chunwei Wang, Yunbo Wang, Hang Xu, Chaoqiang Ye, Zhen Yang, and Chao Ma. Learning transferable features for point cloud detection via 3d contrastive co-training. *Advances in Neural Information Processing Systems*, 34:21493–21504, 2021. 2

[52] Jiakang Yuan, Bo Zhang, Xiangchao Yan, Tao Chen, Botian Shi, Yikang Li, and Yu Qiao. Bi3d: Bi-domain active learning for cross-domain 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15599–15608, 2023. 2, 5, 8

[53] Marvin Mengxin Zhang, Sergey Levine, and Chelsea Finn. MEMO: Test time robustness via adaptation and augmentation. In *NeurIPS*, 2022. 2

[54] Weichen Zhang, Wen Li, and Dong Xu. Srdan: Scale-aware and range-aware domain adaptation network for cross-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6769–6779, 2021. 1, 2, 5

[55] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. *arXiv preprint arXiv:2104.02008*, 2021. 1

[56] Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 1

[57] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2