# Sim-to-Real adversarial domain adaptation for 3D object detection

Maciej Wozniak, Mattias Hansson, Patric Jensfelt
KTH Royal Institute of Technology
maciejw,matthans,patric@kth.se

Marko Thiel
Hamburg University of Technology
marko.thiel@tuhh.de

## Abstract

*In this study, we address a gap in existing unsupervised domain adaptation approaches on LiDAR-based 3D object detection, which have predominantly concentrated on adapting between established, high-density autonomous driving datasets. We focus on sparser point clouds, capturing scenarios from different perspectives: not just from vehicles on the road but also from mobile robots on sidewalks, which encounter significantly different environment conditions and sensor configurations. We introduce Unsupervised Adversarial Domain Adaptation for 3D Object Detection (**UADA3D**). UADA3D does not depend on pre-trained source models or teacher-student architectures. Instead, it uses an adversarial approach to directly learn domain-invariant features. Code will be available at https://github.com/maxiuw/UADA3D.*

## 1. Introduction

LiDAR-based perception systems are essential for the safe navigation of autonomous vehicles such as self-driving cars [9] or mobile robots [17]. A key challenge is the reliable detection and classification of objects within a vehicle's environment [24]. SOTA 3D object detection methods highly depend on quality and diversity of the datasets used for training, but also on how closely these datasets reflect real-world conditions during inference. Acquiring and annotating such data remains significant technical and practical challenge presents a major obstacle in development and deployment of 3D object detection models at scale.

A crucial technique to mitigate these challenges is domain adaptation (DA). DA addresses the problem of adapting models trained on a source domain with ample labeled data to a target domain where labels might be scarce (as in semi-supervised DA) or completely unavailable (as in unsupervised DA – UDA) [5, 2, 15, 19]. UDA methods can substantially improve model performance in new, unfamiliar, or changing environments without the need to label new training samples. In the context of autonomous vehicles, discrepancies between source and target domains, often referred to as domain shift or domain gap, can be caused by changes in weather conditions [10], variations in object sizes [20], different sensor setups and deployment environments [24, 14, 13] but also due to the transition from simulated to real-world environments [4].

UDA has received considerable attention in the field of computer vision. However, recent UDA approaches [12, 7, 20, 18, 3, 11] for LiDAR-based 3D object detection primarily focus on automotive applications and corresponding datasets with dense LiDAR data, featuring 128, 64, or 32 layers [6, 16, 1, 22]. We find a notable research gap when it comes to UDA for 3D object detection models explicitly addressing larger domain shifts than those associated with classical self-driving cars, such as last-mile delivery mobile robots. These robots operate in an environment sharing many properties with that of self-driving cars, potentially allowing them to benefit from widely available datasets, yet they display significant differences: LiDAR sensors differ in both sensor position and resolution, resulting in sparser point clouds where the ground plane is located much closer to the sensor. Moreover, sidewalk environments are considerably different from roadways. While the same object classes are present, their distribution and relative distances to the sensor are distinctly different than from the car perspective, resulting in a different point density per object.

Inspired by 2D image-based adversarial DA, we propose a novel approach for 3D point cloud data: Unsupervised Adversarial Domain Adaptation for 3D Object Detection (**UADA3D**). Our method uses adversarial adaptation based on class-wise domain discriminators with a gradient reversal layer to facilitate the learning of domain-invariant representations. The domain discriminator is trained to maximize its ability to distinguish between the target and source domains, while the model is trained to minimize this ability, resulting in domain-invariant feature learning. The approach we present offers significant advantages over existing UDA methods for LiDAR-based 3D object detection: UADA3D does not require pre-trained source models, avoids the complexity of teacher-student architectures, and eliminates the inherent uncertainties of pseudo-labels. Instead, it directly learns features that are invariant across

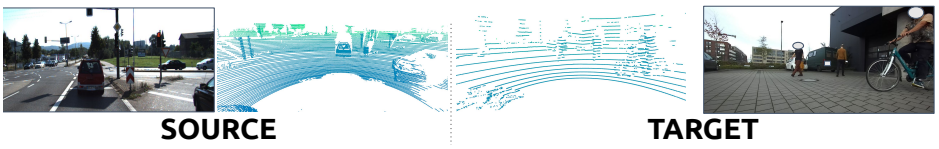


Figure 1. Comparison of KITTI (source) and robot data (target). We observe that differences in operating environments, sensor positions, and LiDAR density create a large domain gap. This presents a significant challenge for LiDAR-based 3D object detectors, as well as for the task of domain adaptation.

the source and target domains. Furthermore, our approach successfully adapts models to multiple object classes simultaneously (e.g. vehicle, pedestrian, and cyclist classes), a capability not often achieved by other SOTA methods.

## 2. Method

### 2.1. Problem Formulation

Suppose that $Q$ is a point cloud, and $X$ is its feature representation learned by the feature extraction network $f_{\theta_f}$. The detection head $h_{\theta_y}$ uses these features to predict $P(Y|X)$, where $Y = (y, b)$ are the category labels $y$ and bounding boxes $b$. $Q$ is sampled from the source domain $\mathcal{D}_s$ and target domain $\mathcal{D}_t$. The objective is to learn a generalized $\theta_f$ and $\theta_y$ between domains such that $P(Y_s, X_s) \approx P(Y_t, X_t)$. Since $P(Y, X) = P(Y|X)P(X)$, the domain adaptation task for LiDAR-based object detectors is to align the marginal probability distributions $P(X_s)$ and $P(X_t)$ as well as the conditional probability distributions $P(Y_s|X_s)$ and $P(Y_t|X_t)$. Note that target labels $Y_t$ are not available during training, thus we must use UDA.

### 2.2. Method Overview

Marginal adaptation, i.e., aligning $P(X)$, overlooks category and position labels, which can lead to uneven and biased adaptation. This may reduce the target domain's discriminative ability. Aligning $P(Y|X)$ places direct emphasis on the task-specific outcomes (class labels and bounding boxes) in relation to the features. By focusing on $P(Y|X)$, we hypothesize that the adaptation process also becomes more robust to variations in feature distributions across domains, concentrating on the essential task of detecting objects. Furthermore the distribution of points in different categories per object varies significantly between the datasets due to LiDAR density, position and operating environment, while the vehicle size slightly varies, depending on the dataset country of origin. While we chose to use conditional alignment, we compare our method with different alignment strategies: marginal and joint distribution alignment in ablation.

Fig. 2 provides a schematic overview of our method UADA3D. In each iteration, a batch of samples $Q$ from source $\mathcal{D}_s$ and target $\mathcal{D}_t$ domain is fed to the feature extractor $f_{\theta_f}$. Next, for each sample, features are extracted, and fed to the detection head $h_{\theta_y}$ that predicts 3D bounding boxes (lines 3-4 in Algorithm 1). The object detection loss is calculated only for the labeled samples from source domain (lines 6-7). The probability distribution alignment branch uses the domain discriminator $g_{\theta_D}$ (line 9) to predict from which domain samples came from, based on the extracted features $X$ and predicted labels $\hat{Y}$. The domain loss $\mathcal{L}_\mathcal{C}$ is calculated for all samples (line 9). Next, the $\mathcal{L}_\mathcal{C}$ is backpropagated through the discriminators (line 10), and through the gradient reversal layer (GRL) with the coefficient $\lambda$, that reverses the gradient during backpropagation, to detection head and feature extractor (lines 11-12). This adversarial training scheme works towards creating domain invariant features. Thus, our network learns how to extract features that will be domain invariant but also how to provide accurate predictions. Therefore, we seek for the optimal parameters $\theta_f^*$,$\theta_y^*$, and $\theta_D^*$, that satisfy:

$$\begin{aligned} \theta_D^* &= \arg\min_{\theta_D} \mathcal{L}_C \\ (\theta_f^*, \theta_y^*) &= \arg\min_{\theta_f, \theta_y} \mathcal{L}_{det} - \lambda \mathcal{L}_C|_{\theta_D^*} \end{aligned} \tag{1}$$

where $\mathcal{L}_{det}$ is the detection loss calculated only for the la-

**Algorithm 1** UADA3D

**Input** Labeled source dataset $\mathcal{D}_s : \{(Q^s, Y^s)\}^{N_s}$, unlabeled target dataset $\mathcal{D}_t : \{(Q^t)\}^{N_t}$

**Output** Weights: backbone $\theta_f$, detection head $\theta_y$, discriminators $\theta_D$

1: $\theta_f, \theta_y, \theta_D \leftarrow$ *Weight Intialization*
2: **for** $Q \in \mathcal{D}_s, \mathcal{D}_t$ **do**
3: $\quad X \leftarrow f_{\theta_f}(Q)$
4: $\quad \hat{Y} \leftarrow h_{\theta_y}(X)$ $\quad \triangleright\ \hat{Y} = (\hat{y}, \hat{b})$
5: $\quad$ **if** source domain **then**
6: $\quad\quad \theta_y \leftarrow UpdateWeights(\theta_y, \frac{\partial \theta_y}{\partial \mathcal{L}_{det}})$
7: $\quad\quad \theta_f \leftarrow UpdateWeights(\theta_f, \frac{\partial \theta_f}{\partial \mathcal{L}_{det}})$
8: $\quad$ **end if**
9: $\quad \mathcal{L}_C \leftarrow \frac{1}{N} \sum^N \hat{y}_{k,n} \odot (g_{\theta_D,k}(x_n, \hat{b_n}) - d)^2$
10: $\quad \theta_D \leftarrow UpdateWeights(\theta_D, \frac{\partial \theta_D}{\partial \mathcal{L}_C})$
11: $\quad \theta_y \leftarrow UpdateWeights(\theta_y, -\lambda \frac{\partial \theta_y}{\partial \mathcal{L}_C})$
12: $\quad \theta_f \leftarrow UpdateWeights(\theta_f, -\lambda \frac{\partial \theta_f}{\partial \mathcal{L}_C})$
13: **end for**
14: **return** $\theta_f^*, \theta_y^*, \theta_D^*$

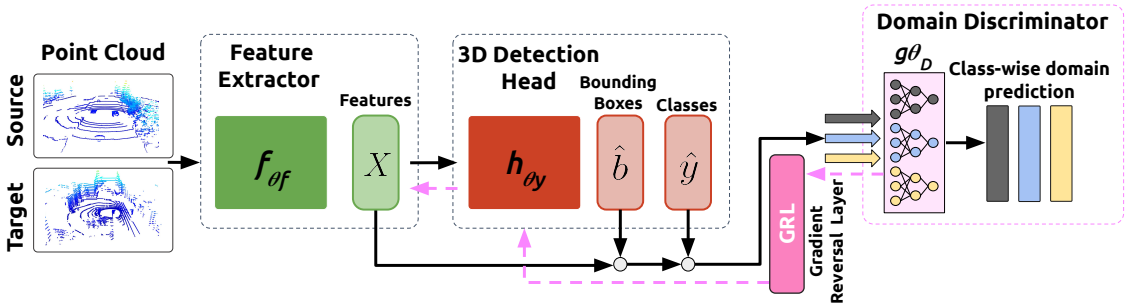


Figure 2. An overview of **UADA3D** (**black** arrows show forwards, and **pink** backwards pass). While the primary task of $f_{\theta_f}$ and $h_{\theta_y}$ is 3D object detection, the discriminator $g_{\theta_D}$ aims to classify the domain of each detected instance. Discriminator's loss, reversed by GRL, encourages the detector to learn features that are not only effective for object detection but also invariant across domains.

beled source domain, $\mathcal{L}_C$ is the domain loss calculated for both source and target domains. $\lambda = 0.1$ is the GRL coefficient.

### 2.3. Feature Masking

*Feature masking* plays a crucial role in predicting the domain based on specific object features. Masking enables the model to focus solely on the features corresponding to each instance, thus enhancing the relevance and accuracy of the domain prediction. In Fig. 3, we show how features extracted from a point cloud $Q$ are masked and used for domain prediction. The input to the class-wise domain discriminators $g_{\theta_{D,k}}$ is $(x, \hat{b})$, where $x$ are masked features, $\hat{b}$ are predcited bounding boxes, and $(a, b)$ denotes a concatenation. To obtain masked features $x$, we mask the feature map $X = f_{\theta f}(Q)$ with each predicted bounding box $\hat{b}_n$ creating corresponding masked features $x_n$. Finally, we concatenate $x_n$ with the bounding box $\hat{b}_n$ and feed to the corresponding class-wise discriminator $g_{\theta_{D,k}}$.

### 2.4. Conditional Distribution Alignment

The conditional distribution alignment module shown in Figs. 2 and 3, has the task of reducing the discrepancy between the conditional distribution $P(Y_s|X_s)$ of the source and $P(Y_t|X_t)$ of the target. As we highlighted in Section 2.2, we can see a large difference between how objects from each category appear in different domains. Thus, instead of having one discriminator, we use $K = 3$ class-wise domain discriminators $g_{\theta_{D,k}}$, corresponding to vehicle, pedestrian, and cyclist classes. The conditional distribution alignment module is trained using the least-squares loss function:

$$\mathcal{L}_C = \frac{1}{N} \sum_{n=1}^{N} \hat{y}_{k,n} \odot (g_{\theta_{D,k}}(x_n, \hat{b}_n) - d)^2 \qquad (2)$$

where $N$ is the number of labels, $\hat{y}_{k,n}$ corresponding class confidence of instance $n$ and $\odot$ is element-wise multiplication. The loss is backpropagated to the discriminators (line 9 in Algorithm 1). Next, $\mathcal{L}_C$ is backpropagated through GRL to the detection head $h_{\theta_y}$ and the feature extractor $f_{\theta_f}$.

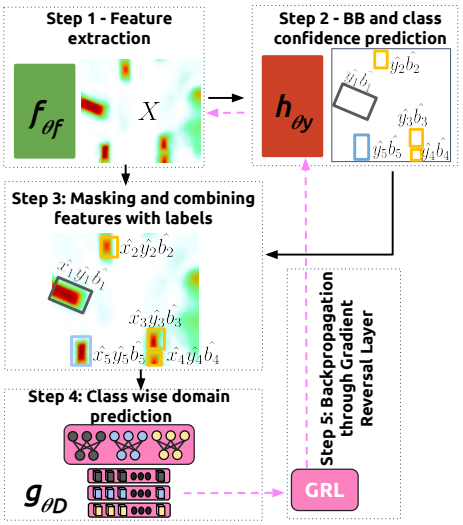


Figure 3. Flowchart showing feature masking and domain prediction in UADA3D.

## 3. Experiments

In Table 1, we compare our method with SOTA on a large number of adaptation scenarios between source and target (S → T) data. In Table 1 we focus on 9 different dense to

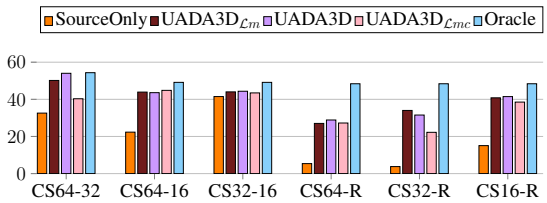


Figure 4. Comparison UADA3D performance with different probability distribution alignments on CenterPoint.

Table 1. Comparison of performance of different adaptation methods on source to target (S→T) domain adaptation tasks. We report $mAP_{3D}$ and $mAP_{BEV}$ over Vehicle, Pedestrian and Cyclist. The best score is **bold** and the second best is underline.

| S→T | Methods | Models | | | | | |
|---|---|---|---|---|---|---|---|
| | | IA-SSD [27] | | | Centerpoint [26] | | |
| | | 3D/BEV | Change | Closed Gap | 3D/BEV | Change | Closed Gap |
| CS64 ↓ R | Source Only | 7.28 / 12.93 | -/- | -/- | 5.38 / 25.97 | -/- | -/- |
| | ST3D++ [25] | 31.79 / 42.19 | 24.50 / 29.26 | 66.53 % / 74.69 % | 23.30 / 33.99 | 17.92 / 8.01 | 41.67 % / 23.88 % |
| | DTS [8] | **33.68 / 46.19** | **26.40 / 33.26** | **71.69 % / 84.90 %** | 16.87 / 32.13 | 11.50 / 6.16 | 26.73 % / 18.34 % |
| | L.D. [21] | 20.70 / 32.10 | 13.41 / 19.17 | 36.42 % / 48.94 % | 28.13 / 38.90 | 22.75 / 12.93 | 52.89 % / 38.51 % |
| | MS3D++ [18] | × | × | × | 14.44 / 37.88 | 9.06 / 11.90 | 21.06 % / 35.47 % |
| | **UADA3D (ours)** | 33.00 / 43.02 | 25.72 / 30.09 | 69.83 % / 76.81 % | **28.87 / 40.09** | **23.49 / 14.12** | **54.62 % / 42.07 %** |
| CS32 ↓ R | Source Only | 8.33 / 15.93 | -/- | -/- | 3.78 / 6.00 | -/- | -/- |
| | ST3D++ [25] | 21.57 / 31.04 | 13.25 / 15.11 | 37.02 % / 41.77 % | 26.57 / 40.68 | 22.80 / 34.69 | 51.09 % / 64.79 % |
| | DTS [8] | 15.37 / 26.94 | 7.05 / 11.01 | 19.69 % / 30.45 % | 18.68 / 29.65 | 14.91 / 23.65 | 33.41 % / 44.18 % |
| | L.D. [21] | 19.43 / 31.38 | 11.10 / 15.45 | 31.02 % / 42.70 % | 26.89 / 36.85 | 23.12 / 30.85 | 51.82 % / 57.62 % |
| | MS3D++ [18] | × | × | × | 4.45 / 42.46 | 0.68 / 36.46 | 1.52 % / 68.09 % |
| | **UADA3D (ours)** | **22.15/ 31.90** | **13.82 / 15.97** | **38.62 % / 44.14 %** | **31.54/ 42.82** | **27.76 / 36.83** | **62.23 % / 68.78 %** |
| CS16 ↓ R | Source Only | 10.28 / 15.26 | -/- | -/- | 15.07 / 33.97 | -/- | -/- |
| | ST3D++ [25] | 9.58 / 29.83 | -0.7 / 14.56 | -2.07 % / 39.53 % | 33.22 / 42.62 | 18.15 / 8.65 | 54.46 % / 33.84 % |
| | DTS [8] | 11.95 / 27.59 | 1.67 / 12.32 | 4.93 % / 33.45 % | 17.35 / 34.34 | 2.28 / 0.37 | 6.84 % / 1.45 % |
| | L.D. [21] | 9.48 / 23.27 | -0.81 / 8.01 | -2.38 % / 33.45 % | 27.13 / 40.19 | 12.06 / 6.22 | 36.19 % / 24.32 % |
| | MS3D++ [18] | × | × | × | 2.16 / 32.30 | -12.91 / -1.67 | -38.75 % / -6.52 % |
| | **UADA3D (ours)** | **19.71 / 35.69** | **19.71 / 20.42** | **27.87 % / 55.44 %** | **41.47 / 46.86** | **26.40 / 12.89** | **79.24 % / 50.43 %** |
| Robot | Oracle (R) | 44.11 / 52.10 | -/- | -/- | 48.39 / 59.54 | -/- | -/- |

sparse scenarios for the two models: Centerpoint [26] and IA-SSD [27]. Our method, UADA3D, outperforms other SOTA approaches in most cases achieving much higher improvements, especially when it comes to the larger domain gaps adapting models towards mobile R. By analyzing *Change* and *Closed Gap*, we can observe that Centerpoint often shows higher adaptability and substantially higher improvement than IA-SSD. That may come from the fact that for IA-SSD, we have to fix a specific number of sampling points, which makes the model less flexible when adapting across different LiDAR densities.

We observe that our method handles diverse domain shifts effectively when adapting between simulation autonomous driving datasets dataset (LiDAR-CS), and robot data (R). As mentioned in Section 2, UADA3D does not need a pre-trained teacher model, as all the other approaches do. Instead, we can directly train the domain-adapted model, leveraging the GRL functionality which creates domain-invariant features. This allows our method to successfully train high-performing models on unlabeled target data, without depending on pseudo labels. We can observe in Table 1 that some of the methods perform even worse than the source-only approaches when tested on adaptation towards more challenging scenarios, failing to generate accurate pseudo labels or distill teacher knowledge. Additionally, it is important to note that UADA3D never achieves lower performance than source-only models, on the adaptation task. Notably, CS16→R adaptations appear particularly difficult, as many methods perform worse than source-only.

**Probability Distribution Alignments:** In these ablation studies, we explore the effectiveness of other probability distribution alignment strategies, shown in Fig. 4. In UADA3D with the marginal distribution alignment, UADA3D$_{\mathcal{L}m}$, the discriminator gradient is backpropagated only to the feature extractor with loss $\mathcal{L}_m$, where $\mathcal{L}_m = \frac{1}{N}\sum_{n=1}^{N} d \cdot \log(g_{\theta_m}(X_n)) + (1-d) \cdot \log(1 - g_{\theta_m}(X_n))$, where $d$ is 0 for source and 1 for target domains and $g_{\theta_M}$ denotes the marginal discriminator network. In UADA3D (i.e., UADA3D$_{\mathcal{L}c}$) the discriminator gradient is backpropagated to the detection head through the whole model. UADA3D$_{\mathcal{L}mc}$ combines marginal and conditional alignment with $\mathcal{L}_{mc} = (\mathcal{L}_m + \mathcal{L}_C)$, where $\mathcal{L}_C$ is given in Eq. (2). UADA3D$_{\mathcal{L}m}$ uses feature maps directly to predict the domain and calculate the loss, while UADA3D employs masked features with class labels and bounding boxes. Conditional probability distribution alignment consistently yields high-quality outcomes, but there are cases where it is not the best one. UADA3D$_{\mathcal{L}m}$ delivers comparable results, especially in cross-sensor adaptation for self-driving cars, due to significant differences in the marginal probability distribution $P(X)$. While UADA3D$_{\mathcal{L}m}$ and UADA3D$_{\mathcal{L}mc}$ sometimes outperform UADA3D, they underperform in some scenarios such as IA-SSD adaptation to *R*. Consequently, we selected conditional alignment as our preferred method.

## 4. Conclusions

We introduced UADA3D that successfully adapt models trained on simulation autonomous driving datasets, towards challenging real world scenarios. Further experiments will be presented in the full article and can be found in arxiv version [23].

## References


[1] 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.

[2] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Domain adaptive faster r-cnn for object detection in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3339–3348, 2018.

[3] 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 International Conference on Computer Vision*, pages 3714–3726, 2023.

[4] Robert DeBortoli, Li Fuxin, Ashish Kapoor, and Geoffrey A Hollinger. Adversarial training on point clouds for sim-to-real 3d object detection. *IEEE Robotics and Automation Letters*, 6(4):6662–6669, 2021.

[5] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189. PMLR, 2015.

[6] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 2013.

[7] Deepti Hegde, Vishwanath Sindagi, Velat Kilic, A Brinton Cooper, Mark Foster, and Vishal Patel. Uncertainty-aware mean teacher for source-free unsupervised domain adaptive 3d object detection. *arXiv preprint arXiv:2109.14651*, 2021.

[8] Qianjiang Hu, Daizong Liu, and Wei Hu. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17556–17566, 2023.

[9] 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.

[10] Jinlong Li, Runsheng Xu, Jin Ma, Qin Zou, Jiaqi Ma, and Hongkai Yu. Domain adaptive object detection for autonomous driving under foggy weather. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 612–622, 2023.

[11] Ziyu Li, Jingming Guo, Tongtong Cao, Liu Bingbing, and Wankou Yang. Gpa-3d: Geometry-aware prototype alignment for unsupervised domain adaptive 3d object detection from point clouds. In *Proceedings of the International Conference on Computer Vision*, pages 6394–6403, 2023.

[12] 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 International Conference on Computer Vision*, pages 8866–8875, 2021.

[13] Thien-Minh Nguyen, Shenghai Yuan, Thien Hoang Nguyen, Pengyu Yin, Haozhi Cao, Lihua Xie, Maciej Wozniak, Patric Jensfelt, Marko Thiel, Justin Ziegenbein, et al. Mcd: Diverse large-scale multi-campus dataset for robot perception. *arXiv preprint arXiv:2403.11496*, 2024.

[14] Xidong Peng, Xinge Zhu, and Yuexin Ma. Cl3d: Unsupervised domain adaptation for cross-lidar 3d detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2047–2055, 2023.

[15] Mark Schutera, Mostafa Hussein, Jochen Abhau, Ralf Mikut, and Markus Reischl. Night-to-day: Online image-to-image translation for object detection within autonomous driving by night. *IEEE Transactions on Intelligent Vehicles*, 6(3):480–489, 2021.

[16] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020.

[17] Chi-Yi Tsai, Humaira Nisar, and Yu-Chen Hu. Mapless lidar navigation control of wheeled mobile robots based on deep imitation learning. *IEEE Access*, 9:117527–117541, 2021.

[18] Darren Tsai, Julie Stephany Berrio, Mao Shan, Eduardo Nebot, and Stewart Worrall. Ms3d: Leveraging multiple detectors for unsupervised domain adaptation in 3d object detection. In *International Conference on Intelligent Transportation Systems*. IEEE, 2023.

[19] Vidit Vidit and Mathieu Salzmann. Attention-based domain adaptation for single-stage detectors. *Machine Vision and Applications*, 33(5):1–14, 2022.

[20] 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 11710–11720, 2020.

[21] 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. In *Proceedings of the European Conference on Computer Vision*, pages 179–195. Springer, 2022.

[22] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS*, 2021.

[23] Maciej K Wozniak, Mattias Hansson, Marko Thiel, and Patric Jensfelt. Uada3d: Unsupervised adversarial domain adaptation for 3d object detection with sparse lidar and large domain gaps. *arXiv preprint arXiv:2403.17633*, 2024.

[24] Maciej K Wozniak, Viktor Kårefjärd, Marko Thiel, and Patric Jensfelt. Towards a robust sensor fusion step for 3d object detection on corrupted data. *IEEE Robotics and Automation Letters*, 2023.

[25] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: Denoised self-training for unsu-

pervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5):6354–6371, 2022.

[26] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021.

[27] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. 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*, pages 18953–18962, 2022.