ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240472

Detect Closer Surfaces That Can Be Seen: New Modeling and Evaluation in Cross-Domain 3D Object Detection

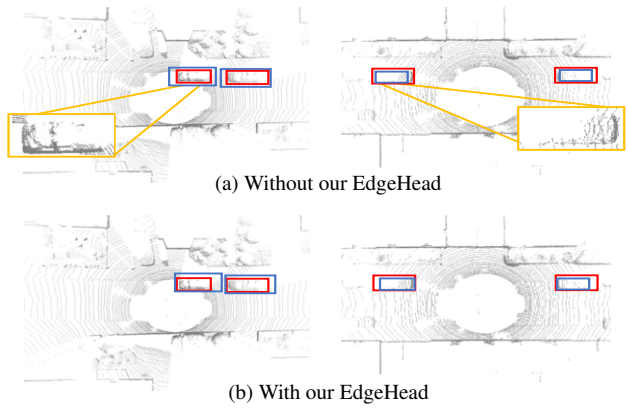
Ruixiao Zhang [a,*], **Yihong Wu** [a], **Juheon Lee**[b,1], **Xiaohao Cai**[a] **and Adam Prugel-Bennett**[a]

[a]Department of Electronics and Computer Science, University of Southampton
[b]Meta Inc.

Abstract. The performance of domain adaptation technologies has not yet reached an ideal level in the current 3D object detection field for autonomous driving, which is mainly due to significant differences in the size of vehicles, as well as the environments they operate in when applied across domains. These factors together hinder the effective transfer and application of knowledge learned from specific datasets. Since the existing evaluation metrics are initially designed for evaluation on a single domain by calculating the 2D or 3D overlap between the prediction and ground-truth bounding boxes, they often suffer from the overfitting problem caused by the size differences among datasets. This raises a fundamental question related to the evaluation of the 3D object detection models' cross-domain performance: Do we really need models to maintain excellent performance in their original 3D bounding boxes after being applied across domains? From a practical application perspective, one of our main focuses is actually on preventing collisions between vehicles and other obstacles, especially in cross-domain scenarios where correctly predicting the size of vehicles is much more difficult. In other words, as long as a model can accurately identify the closest surfaces to the ego vehicle, it is sufficient to effectively avoid obstacles. In this paper, we propose two metrics to measure 3D object detection models' ability of detecting the closer surfaces to the sensor on the ego vehicle, which can be used to evaluate their cross-domain performance more comprehensively and reasonably. Furthermore, we propose a refinement head, named EdgeHead, to guide models to focus more on the learnable closer surfaces, which can greatly improve the cross-domain performance of existing models not only under our new metrics, but even also under the original BEV/3D metrics. Our code is available at https://github.com/Galaxy-ZRX/EdgeHead.

1 Introduction

3D object detection aims to localize and categorize different types of objects in specific 3D space described by 3D sensor data (*e.g.*, LiDAR point clouds). Recently, the application of this technology has achieved significant improvement due to the development of deep neural networks, especially in the field of autonomous driving. Current 3D object detection methods mainly focus on specific datasets, *i.e.*, models will be trained and tested independently on a specific dataset. In doing so, a number of models achieved high performances on public benchmarks including nuScenes [2], Waymo [21], and KITTI [10]. However, if the application of the model on a new dataset is needed, the training on the new dataset as well as modifications of some training hyper-parameters are usually necessary. In other words, it is hard for models trained on one dataset to adapt directly to another. These domain shifts may arise from different sensor types, weather conditions [24], and object sizes [30] between different datasets or domains. This domain adaptation problem is therefore a big challenge for real-world applications of existing 3D object detection methods, as their retraining steps can be very slow and resource-consuming. It is thus significant to understand the reasons for the drop in cross-domain performance and propose efficient methods to raise the cross-domain performance to the same level as within-domain tasks. By comparing the performance of two models [19, 29] on several different datasets, the work in Wang et al. [24] showed that the difference in car size across geographic locations is one of the main challenges for domain adaptation problems. Other factors such as the difference in LiDAR point cloud densities [25, 14] and weather conditions [24] have also been investigated.



(a) Without our EdgeHead

(b) With our EdgeHead

Figure 1. Prediction properties of models before and after equipping with our proposed EdgeHead in cross-domain tasks. Red: Ground-truth boxes. Blue: Predictions. The left and right columns showcase the prediction properties when the training domain respectively has a larger and smaller average object size than the target domain. Object size overfitting problem occurs in both cases, while our proposed EdgeHead can help better detect the closer surfaces to the ego vehicle.

Some generic training methods have tried to overcome the domain adaptation problem. Among these, some methods explore the possibility of adding information about domain gaps into the models by training or using additional prior knowledge [24, 30, 33, 7] to adapt the trained models from the training domain to new domains. Some other methods tried to improve the models' generalization ability to enhance their performance on multiple domains without fine-tuning

* Corresponding Author. Email: rz6u20@soton.ac.uk.
[1] This study was conducted before joining Meta.

and additional training operations [14, 34]. It is common practice for methods to evaluate their cross-domain performance using established 3D object detection metrics designed for single-domain tasks, such as bird's-eye view average precision (BEV AP) and 3D average precision (3D AP). Since these metrics are initially designed for the performance measurement on a single domain, they do not consider the differences across domains and therefore easily suffer from the overfitting problem in object size, making them difficult to evaluate models' cross-domain performance comprehensively.

From another point of view, existing metrics are usually designed to evaluate models' ability to predict the complete shapes of objects. However, as shown in Figure 1, most objects' point cloud is incomplete due to physical obstructions in the capturing process of LiDAR sensors. As a result, it is difficult to predict the full size and entire box of objects. In within-domain tasks, we may be able to "guess" the prediction through training with large amounts of samples. However, the guessing easily becomes incorrect in cross-domain tasks since the average object size has changed. Therefore, correctly predicting the location (usually represented by the center of the box) along with the full box size becomes a much more difficult task. Instead, we argue that what we can better guide the model to learn from the incomplete object point cloud is to predict the closer surfaces of the objects to the ego vehicle, since there are plenty of points over there as shown in Figure 1. Among the four surfaces perpendicular to the ground, capturing and describing these two is more reasonable and also more important, since our main purpose is to avoid potential collisions with surrounding objects during driving.

In this work, we approach the problem of cross-domain 3D object detection from a different perspective. Our main contributions are as follows: We first propose two novel evaluation metrics that aim to evaluate models' ability to detect the closer surfaces to the sensor. By using the defined absolute gap of the closer surfaces (*i.e.*, G_{cs}) to penalize the original BEV AP, we propose the *closer-surfaces penalized BEV* (*i.e.*, the CS-BEV) AP and achieve a balance between the detection quality of the entire box and the closer surfaces to the ego vehicle. Additionally, to avoid the interference of different BEV AP, we introduce the *absolute closer-surfaces* (*i.e.*, the CS-ABS) AP metric for a more precise analysis of improvements in closer-surfaces detection contributed by different methods. Furthermore, we design a second-stage refinement head, *i.e.*, *EdgeHead*, specifically for the closer-surfaces detection task. It can directly be combined with various existing models. Our EdgeHead can guide models to make predictions with a smaller gap between the closer surfaces and the closest vertex to the ego vehicle. To the best of our knowledge, this is the first work to improve cross-domain performance with minimal modifications to existing models by adding a novel refinement module.

Thorough experiments show that our proposed metrics can find a balance between evaluating the detection quality of the entire boxes and the closer surfaces to the ego vehicle, and provide a brand new way to describe and measure models' remaining detection ability after adapting to new domains. We also experimentally prove that by simply equipping existing models with the proposed EdgeHead, their detection ability on the closer surfaces can be greatly improved, which not only leads to higher performance under the new proposed metrics but also improves the performance under the original BEV and 3D metrics. The results indicate that by guiding the models to focus more on the unobstructed closer surfaces that can be scanned and surrounded by more points, our proposed strategies can help the models learn more robustly from the point cloud data, and therefore predict both the closer surfaces and the entire boxes better.

In brief, our findings suggest that 3D vision and autonomous driving researchers pay more attention to the detection quality of the objects' closer surfaces to the ego vehicle in cross-domain 3D object detection tasks, which can help models learn more robustly and perform better across different domains.

2 Related Work

3D object detection with point clouds. A common way of representing real-world 3D space is using LiDAR point cloud data, i.e., using 3D points to record the 3D environment. Although it benefits from accurate point locations, the main challenge for LiDAR-based 3D object detection methods is finding the best way to process the point cloud data. CNNs have been widely used in 2D object detection problems; however, due to the sparsity and spatial disorder of the point cloud data, they cannot be directly fed into the CNNs. Therefore, current LiDAR-based methods either transform the point clouds into spatial invariant formats for CNN models or learn features directly from the geometry of 3D points. For example, VoxelNet [36] and SECOND [28] encoded point clouds into voxels so that features can be extracted by CNNs designed for 3D inputs. MV3D [4] projected point clouds into 2D spaces (*i.e.*, the front view and bird's-eye view) and then used 2D CNNs to extract features. PointRCNN [19] applied PointNet++ [18] to obtain 3D point-wise features and directly learned the 3D proposals from the points. PV-RCNN [20] voxelized the point cloud data first and then used key-point-wise features to keep more semantic features, in which the combination of point-based and voxel-based methods greatly improved the performance with acceptable computation cost. More recently, some methods [6, 17] further explored the potential of convolution-based models and have achieved state-of-the-art performance. With the development of transformers [9, 3] in the computer vision field, there are also methods attempting to use transformer-based models to predict 3D objects. For example, TransFusion [1] used the transformer decoder to combine the image features and LiDAR point cloud features for prediction quality enhancement.

Domain adaptation. Domain adaptation has been widely used in 2D object detection [13, 16, 23] and 2D semantic segmentation [5, 12, 15]. However, there are limited approaches specifically designed for 3D object detection with domain adaptation. Wang et al. [24] first proposed the overfitting problem in object sizes and provided a simple but effective solution by normalizing the object size of different datasets based on additional prior knowledge. ST3D [30] and ST3D++ [31] proposed the random object scaling (ROS) algorithm to solve the overfitting problem in car size and used self-training algorithms to generate pseudo labels and train models on the target domain without ground truth. Wei et al. [25] proposed a distillation method for LiDAR point clouds in order to overcome the beam difference between datasets, which enables models to adapt to point cloud data with lower densities and fewer beams.

Model generalization. Besides the adaptation methods described above, there are some explorations on improving models' generalization ability, *i.e.*, training models only once and enabling them to achieve acceptable performance on more domains. Uni3D [34] aligned the unavoidable differences between datasets via a data-level correction operation and a semantic-level coupling-and-recoupling module. Wu et al. [26] proposed a multi-domain knowledge transfer framework to leverage spatial-wise and channel-wise knowledge across domains, which helps extract universal feature representations for models. Hu et al. [14] proposed the random beam re-sampling (RBRS) method to improve the models' beam-density robustness and used a teacher-student framework to generate pseudo labels on unseen target domains.

3 Method

Motivation. We first point out the weakness of existing metrics such as BEV and 3D AP in cross-domain tasks. Given two predictions of the same car ground truth as shown in Figure 1, exactly the same BEV and 3D AP will be obtained as they have the same overlaps with the ground-truth bounding box. However, there is a big difference when comparing their detection quality of the surfaces that are closer to the LiDAR sensor, which are not occluded by other surfaces and therefore have a bigger chance of being captured by the sensor. This detection quality indicates how correctly a model can estimate the distance from our car to the surfaces of other objects that could collide with us, which is directly related to driving safety and should be paid more attention to than the other two surfaces of detected objects. It is therefore essential to develop new metrics that can accurately assess this detection quality of models, thereby enabling a more reasonable and comprehensive evaluation of performance, particularly for cross-domain tasks.

From another point of view, existing models are easy to overfit on the training domain, on which they are designed and trained to perform well. This limits their detection ability across domains. One of the main factors that causes this overfitting problem is that these models usually have a regression module to learn the offsets of box dimensions and locations between the prediction and ground truth. Such a module results in the overfitting on object sizes, especially for anchor-based methods. It is thus critical to explore the performance of a specifically designed model with the consistent aim of the new metric we usher in here, *i.e.*, the one that focuses more on the detection quality of the closer surfaces of the bounding box.

The aim of our endeavor is therefore twofold. Firstly, we aim to measure models' cross-domain 3D object detection performance with fairer and more reasonable metrics. Specifically, we will design new evaluation metrics that will be less influenced by the cross-domain factors (*e.g.*, object sizes and point cloud densities). Secondly, we attempt to improve the models' detection ability that can be preserved across different domains by guiding them with the newly proposed metrics. We will achieve this by equipping existing models with an additional proposed refinement head, called EdgeHead, together with novel proposed loss functions.

3.1 Problem statement

The purpose of 3D object detection is to predict the 3D bounding boxes of objects that are parameterized by the center locations (x_c, y_c, z_c), the sizes (l, w, h) and the rotation angle θ. In cross-domain tasks, we train models on the source domain $\{(X_i^s, Y_i^s)\}_{i=1}^{N_s}$, but focus on their performance on the target domain $\{X_i^t\}_{i=1}^{N_t}$, where N_s and N_t are respectively the number of samples in the source domain and target domain, X_i^s and Y_i^s respectively denote the i-th source domain point cloud data and its corresponding label, and X_i^t denotes the i-th target domain point cloud data.

3.2 Closer-surfaces-based evaluation metrics

We now propose new evaluation metrics to measure the models' ability to detect the closest corner of an object and its two surfaces that are closer to the LiDAR sensor.

We first define the absolute gap between the closer surfaces of predictions and ground truth. Given a prediction box with its vertices $\{V_{\text{pred}}^i\}_{i=1}^4$ on the BEV plane and the related ground-truth box with its vertices $\{V_{\text{gt}}^i\}_{i=1}^4$, we first sort their vertices by their distance to the origin (*i.e.*, the location of the LiDAR sensor), and then further sort the second and third vertices by their absolute x-coordinate. After sorting, prediction and ground truth boxes should follow the same indexing rule for their vertices, *i.e.*, V^1 and V^4 are respectively the vertices closest and furthest to the origin, and V^2 is the vertex having a smaller absolute x-coordinate compared with V^3. We can then define the absolute gap, say G_{cs}, of the closer surfaces between the prediction and the ground truth, *i.e.*,

$$G_{\text{cs}} = |V_{\text{pred}}^1 - V_{\text{gt}}^1| + \text{Dist}(V_{\text{pred}}^2, E_{\text{gt}}^{1,2}) + \text{Dist}(V_{\text{pred}}^3, E_{\text{gt}}^{1,3}), \quad (1)$$

where $E^{i,j}$ is the edge connecting V^i and V^j, and $\text{Dist}(V, E)$ calculates the perpendicular distance from vertex V to edge E.

The defined absolute gap G_{cs} in Eq. (1) can be used to measure the detection quality of the closer surfaces; however, it will fluctuate with the sizes of the object boxes. In other words, it is not a scaled metric that can be used to calculate the AP with pre-determined thresholds. To solve this problem, we propose the ***absolute closer-surfaces AP*** (*i.e.*, the CS-ABS AP) to directly measure the detection quality of the closer surfaces by using

$$\Gamma_{\text{ABS}}^{\text{CS}} = 1/(1 + \alpha G_{\text{cs}}), \quad (2)$$

where $\alpha \geq 0$ is the penalty ratio set to 1 by default.

The proposed CS-ABS AP by Eq. (2) can also be utilized to combine with existing popular metrics and thus form new metrics with hybrid effectiveness for more powerful and fairer evaluation of models' performance. In particular, we combine the CS-ABS AP with the BEV AP and propose the ***closer-surfaces penalized BEV AP*** (*i.e.*, the CS-BEV AP) to measure the detection quality by using the penalized IoU, say $\Gamma_{\text{BEV}}^{\text{CS}}$, *i.e.*,

$$\Gamma_{\text{BEV}}^{\text{CS}} = \Gamma_{\text{BEV}}/(1 + \alpha G_{\text{cs}}) \quad (3)$$

where Γ_{BEV} is the original BEV IoU and α is the penalty ratio set to 1 by default based on experimental experience (see more discussions in the Supplementary Material [35]). Our proposed CS-BEV metric in Eq. (3) not only retains the robustness of the original BEV metric but also better distinguishes the detection quality of the closer surfaces. It finds an evaluation balance between the quality of the entire 3D box and the quality of the closer surfaces. Taking the same examples in Figure 1, the newly proposed metric will return a higher AP when the prediction matches the closer surfaces of the ground truth better.

In sum, we in this section proposed two evaluation metrics: CS-ABS AP and CS-BEV AP. The CS-ABS AP can directly tell the detection quality of the closer surfaces without considering the ability to evaluate the quality of the entire 3D box, which can be specifically used when analyzing the detection quality gain regarding closer surfaces. The CS-BEV AP can find a balance between the quality of the entire 3D box and the closer surfaces, which is more comprehensive and can be used to measure the overall cross-domain performance for different models and tasks.

3.3 EdgeHead for closer-surfaces localization

To improve models' closer-surfaces localization ability, we propose a refinement head, *i.e.*, the ***EdgeHead***, by modifying the models' training purpose. Similar to other refinement heads, the proposed EdgeHead takes the predictions of a model's first stage as the regions of interest (RoIs). It then aggregates the features from earlier backbones of the model (*e.g.*, the 3D convolution backbones) into the RoI features for prediction refinement. During the refinement process of

EdgeHead, we modify the loss function to guide the model to learn the closer-surfaces offsets between the predictions and ground truth.

The voxel RoI pooling [8] is used to aggregate the RoI features. In detail, we extract the 3D voxel features from the last two layers in the 3D sparse convolution backbone, which is available for most voxel-based 3D object detection models. Afterwards, the feature of each RoI is assigned by aggregating the 3D features from its neighbor voxels via the voxel query operation. Since features from the 3D backbones usually contain more spatial and structural information, the aggregated RoI features can help improve the detection quality of the closer surfaces of bounding boxes.

The loss of a typical RoI refinement module consists of two parts, *i.e.*, the IoU-based classification loss [20] and the regression loss. In detail, the original regression loss, say $\mathcal{L}_{\text{reg}}$, uses the smooth ℓ_1 loss [11] to learn the 7 parameters of the bounding boxes, *i.e.*,

$$\mathcal{L}_{\text{reg}} = \sum_{r \in \{x_{\text{c}}, y_{\text{c}}, z_{\text{c}}, l, h, w, \theta\}} \mathcal{L}_{\text{smooth}-\ell_1}(\widehat{\Delta r^a}, \Delta r^a) \tag{4}$$

where $\widehat{\Delta r^a}$ and Δr^a are the predicted residual and the regression target, respectively. In our EdgeHead, we use the original classification loss and modify the regression loss in Eq. (4), which will guide the model to learn the closer-surfaces offsets between the predictions and ground truth.

Given the closest vertex of the anchor box and the ground-truth box respectively as $(x^a_{\text{cv}}, y^a_{\text{cv}}, z^a_{\text{cv}})$ and $(x^{\text{gt}}_{\text{cv}}, y^{\text{gt}}_{\text{cv}}, z^{\text{gt}}_{\text{cv}})$, we first rotate the anchor box by the rotation angle θ_{gt} of the ground-truth box as shown in Figure 2(c), and denote the rotated box's closest vertex by $(x^{a'}_{\text{cv}}, y^{a'}_{\text{cv}}, z^{a'}_{\text{cv}})$. We then calculate the residuals of x_{cv} and y_{cv} between the rotated anchor box and ground truth as follows

$$\Delta x_{\text{cv}} = x^{\text{gt}}_{\text{cv}} - x^{a'}_{\text{cv}}, \quad \Delta y_{\text{cv}} = y^{\text{gt}}_{\text{cv}} - y^{a'}_{\text{cv}}, \tag{5}$$

and modify the ℓ_1 loss by replacing the residual of center locations with the residuals of the rotated closest vertex to the origin (*i.e.*, our ego vehicle) as calculated in Eq. (5). Since the Z-axis of bounding boxes is always set to be perpendicular to the horizontal plane, the distances of the closer surfaces between the predictions and ground truth are only related to the X and Y coordinates. We therefore remove the regression for z_{cv} and focus on x_{cv} and y_{cv}. To avoid the overfitting problem on object sizes, we also remove the parts of regression loss for the residuals related to object sizes (*i.e.*, l, w, h). As a result, we only refine x_{cv}, y_{cv}, and θ in our EdgeHead and keep the z_{cv}, l, w, and h as predicted by the model's first stage. The new regression loss of our EdgeHead is therefore defined as

$$\mathcal{L}'_{\text{reg}} = \sum_{r \in \{x_{\text{cv}}, y_{\text{cv}}, \theta\}} \mathcal{L}_{\text{smooth}-\ell_1}(\widehat{\Delta r^a}, \Delta r^a). \tag{6}$$

Remark. The rotation of the anchor box used in Eq. (5) is important in the modification of the regression process to realize our real purpose, *i.e.*, to guide the model to learn the closer-surfaces offsets between the predictions and ground truth. Considering an example of the model's regression process without the rotation as shown in Figure 2(a), the regression target related to x and y will guide the model to predict the residual so that the predicted box can coincide with the ground-truth box at the vertex closest to the origin. However, since we are also regressing the rotation angle θ, we will finally get a predicted box as shown in Figure 2(b), whose closest vertex to the origin does not coincide with the ground truth's anymore (see the red arrow). To consider the rotation regression as well, we first rotate the anchor box by the rotation angle θ_{gt} of the ground-truth box as shown in Figure 2(c), and then calculate the residuals of x_{cv} and y_{cv} between the rotated anchor box and ground truth as the new regression target. Such a modified regression process makes the prediction box's closest vertex finally coincide with the ground-truth box's.

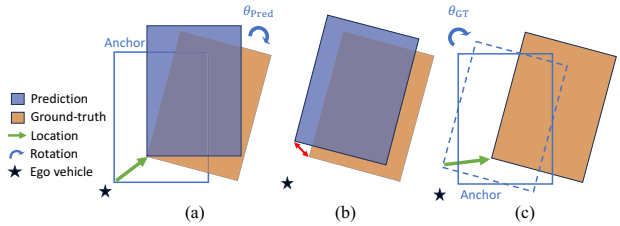


Figure 2. Illustration of different regression processes. (a) The process that directly regresses the closest vertex and rotations without rotating the anchor box first. (b) The prediction obtained using the process in (a), in which the red arrow shows that the prediction does not learn the closest vertex as expected. (c) The regression process guided by Eq. (5) and Eq. (6) in our EdgeHead, which first rotates the anchor by θ_{gt} and then calculate the regression target of x and y locations.

3.4 *Use of point-wise features in EdgeHead*

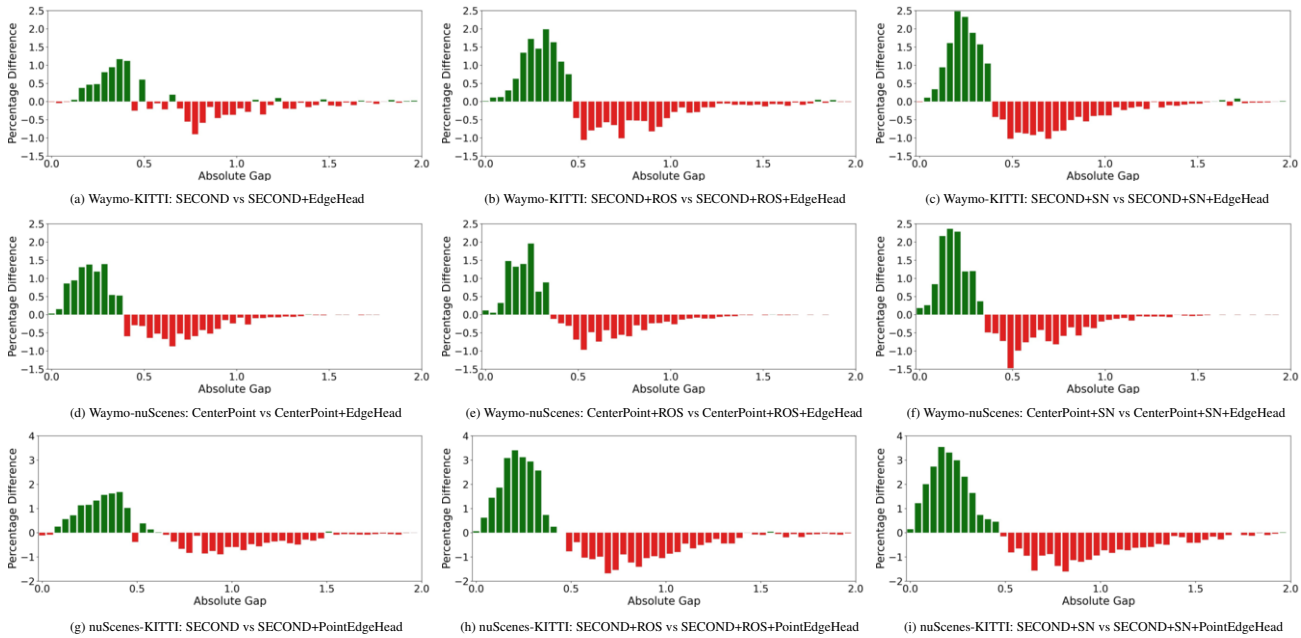
We notice that some models use the additional raw point features as part of the input of the RoI refinement head to aggregate structural and spatial information into the RoI features to help improve object localization accuracy. This inspires us to investigate whether the point-wise features can also help improve the detection quality of the closer surfaces. To do so, we further include the point feature aggregation into the RoI pooling module of our EdgeHead. Specifically, we follow the idea of PV-RCNN [20] to extract the point-wise features. The keypoints are sampled from the original point cloud by the furthest-point-sampling algorithm, and the predicted-keypoint-weighting module is used to re-assign the weights of each point feature, which consists of a three-layer MLP network and a sigmoid function to predict the confidence that each point belongs to the foreground (*i.e.*, inside an object box). Afterwards, the weighted keypoint features are aggregated into the related RoIs via the set-abstraction-based RoI grid pooling, together with the 3D convolution features described in Section 3.3 above. We name this extended EdgeHead the ***point-enhanced EdgeHead*** – shortened as ***PointEdgeHead***.

4 Experiments

4.1 *Datasets and models*

Datasets. We conduct our main experiments on three datasets that have been widely used in 3D object detection tasks: KITTI [10], nuScenes [2], and Waymo [21]. Following existing works [24, 30, 25], we use the KITTI evaluation metric for all datasets on the car category (*i.e.*, the vehicle category in Waymo). As mentioned in Yang et al. [30], KITTI only provides annotations in the front view, which makes it much more difficult to adapt models from KITTI to the other two datasets that provide ring view point cloud data and annotations. We therefore evaluate the models' cross-domain performance via the following tasks: nuScenes → KITTI, Waymo → KITTI and Waymo → nuScenes.

Data integration. Most existing 3D object detection methods [28, 20, 19] aimed to achieve higher performance within each specific domain/dataset, and they often fine-tuned the models for different datasets independently (*e.g.*, adjusting the hyper-parameters related to data preprocessing and using different voxel sizes) without considering the influence of domain gaps. However, to investigate the



(a) Waymo-KITTI: SECOND vs SECOND+EdgeHead (b) Waymo-KITTI: SECOND+ROS vs SECOND+ROS+EdgeHead (c) Waymo-KITTI: SECOND+SN vs SECOND+SN+EdgeHead

(d) Waymo-nuScenes: CenterPoint vs CenterPoint+EdgeHead (e) Waymo-nuScenes: CenterPoint+ROS vs CenterPoint+ROS+EdgeHead (f) Waymo-nuScenes: CenterPoint+SN vs CenterPoint+SN+EdgeHead

(g) nuScenes-KITTI: SECOND vs SECOND+PointEdgeHead (h) nuScenes-KITTI: SECOND+ROS vs SECOND+ROS+PointEdgeHead (i) nuScenes-KITTI: SECOND+SN vs SECOND+SN+PointEdgeHead

Figure 3. Proportion difference of the absolute gap of the closer surfaces. (a)–(c): SECOND in the Waymo → KITTI task. (d)–(f): CenterPoint in the Waymo → nuScenes task. (g)–(i): SECOND with our PointEdgeHead in the nuScenes → KITTI task. Columns one to three show the comparisons of models without domain adaptation methods, with the ROS, and with the SN, respectively. More results are given in the Supplementary Material [35].

cross-domain performance of these models, we must find a way to merge these datasets. We note that the following differences between datasets have a significant influence on the cross-domain experiments, *i.e.*, (i) the point cloud range; (ii) the origin of coordinates; and (iii) the unit for preprocessing the point cloud data, such as voxel sizes in voxel-based methods. Following the ideas in previous works [24, 30], some preprocessing methods are adopted. We set the point cloud range of all datasets to $[-75.2, -75.2, -2, 75.2, 75.2, 4]$ meters and shift the whole point cloud space of different datasets vertically so that the X-Y plane always coincides with the horizontal plane. Following Yang et al. [30], we set the voxel size of voxel-based methods to $(0.1, 0.1, 0.15)$ meters for all datasets.

Baseline models. To better investigate the influence of model structures on cross-domain performance, we train SECOND [28] and CenterPoint [32] on KITTI, Waymo, and nuScenes using the OpenPCDet [22] toolbox with suggested numbers of epochs and learning rates. In detail, we train them on KITTI for 80 epochs with learning rate 1×10^{-3} and batch size of 8. Epochs of 50 and batch size of 16 are used for the training of the same models on nuScenes, and epochs of 30 and batch size of 8 for Waymo. For the models equipped with our EdgeHead, we train them based on the pre-trained original models for the same epochs as above, during which the parameters of the original models are frozen and only the heads are being trained. We also train the above models with two domain adaptation methods – ROS [30] and SN [24] – and combine them with our EdgeHead. We use the same training settings as their original reproductions on OpenPCDet. Following other works [27, 30] based on OpenPCDet, we adopt random horizontal flip, rotation, and scale transforms during the training process. All models are trained on RTX 8000.

Evaluation metrics. We follow KITTI to evaluate the models' performance under the original BEV and 3D metrics. The evaluation is focused on the *Car* category (*i.e.*, the *Vehicle* in Waymo) which has the most samples in all the datasets and has been the main focus in existing works. AP (average precision) for BEV and 3D metrics (*i.e.*, the AP_{BEV} and AP_{3D}) with the IoU threshold at 0.7 is reported, *i.e.*, a car is marked as correctly detected if the IoU between the prediction and the ground truth is larger than 0.7. Our proposed metrics, *i.e.*, the CS-ABS AP and the CS-BEV AP, are denoted by $AP_{CS\text{-}ABS}$ and $AP_{CS\text{-}BEV}$, respectively; and we report the results with the IoU threshold at 0.7 for $AP_{CS\text{-}ABS}$ and 0.5 for $AP_{CS\text{-}BEV}$ (given that $AP_{CS\text{-}BEV}$ is more difficult to reach 0.7).

Table 1. Performance comparison of original models under four metrics in both cross-domain and within-domain tasks. W, K, and N represent the Waymo, KITTI, and nuScenes datasets, respectively.

Task	Method	AP_{BEV}	AP_{3D}	$AP_{CS\text{-}BEV}$	$AP_{CS\text{-}ABS}$
W → K	SECOND	49.2	9.3	19.0	10.9
	CenterPoint	51.3	13.1	18.2	9.5
W → N	SECOND	27.8	16.1	15.6	6.7
	CenterPoint	30.4	16.7	19.9	12.3
N → K	SECOND	35.7	11.8	16.4	9.8
	CenterPoint	34.6	8.3	13.1	5.8
K → K	SECOND	84.3	72.1	71.4	53.3
	CenterPoint	84.4	73.6	72.8	56.8

4.2 *The absolute gap of the closer surfaces*

We first compare the proposed EdgeHead with existing methods by measuring their absolute gaps of the closer surfaces (*i.e.*, G_{cs}). As shown in Figure 3, we calculate the distributions of G_{cs} for each comparison pair of methods and draw the proportion difference between them. Specifically, we quantify the G_{cs} distribution of two models within an identical interval I (set to $[0, 2]$ by default), and then calculate the proportion difference as $\text{Diff}_{AB}(i) = P_B^i - P_A^i$, where P_A^i and P_B^i denote the proportion of G_{cs} in the i-th sub-interval of I for models A and B, respectively. Therefore, if the left part of the proportion difference graph is above the X-axis and the right half is vice versa, we can tell that model B predicts the closer surfaces better than model A and thus has a G_{cs} distribution closer to zero. For example, Figure 3(a) shows that SECOND+EdgeHead (*i.e.*, the SECOND model combined with our proposed EdgeHead)

Table 2. Main comparisons for SECOND and CenterPoint across different tasks. We report AP_{BEV}, AP_{3D} and $AP_{CS\text{-}ABS}$ of the car category at IoU = 0.7 and $AP_{CS\text{-}BEV}$ at IoU = 0.5. The reported performance is the moderate case when KITTI is the target domain, and is the overall result for other cross-domain tasks. Improvement (*i.e.*, fifth column) is calculated by the relative difference between each used method and the original model (*i.e.*, the first row of each task).

Task	Method	SECOND			CenterPoint		
		AP_{BEV}/ AP_{3D}	$AP_{CS\text{-}BEV}$/ $AP_{CS\text{-}ABS}$	Improvement (%)	AP_{BEV}/ AP_{3D}	$AP_{CS\text{-}BEV}$/ $AP_{CS\text{-}ABS}$	Improvement (%)
W → K	Original	49.2 / 9.3	19.0 / 10.9	-	51.3 / 13.1	18.2 / 9.5	-
	+ EdgeHead	52.3 / 10.7	23.7 / 14.7	24.7% / 34.9%	53.9 / 14.5	22.0 / 13.3	20.9% / 40.0%
	+ ROS	73.0 / 38.3	33.7 / 12.6	77.4% / 15.6%	75.1 / 44.2	41.1 / 19.1	126.1% / 101.1%
	+ EdgeHead & ROS	76.4 / 41.5	42.9 / 20.4	125.8% / 87.2%	77.3 / 47.4	46.2 / 23.2	154.4% / 144.2%
	+ SN	73.0 / 55.5	49.3 / 20.5	159.5% / 87.9%	72.5 / 56.7	51.4 / 24.9	182.4% / 162.1%
	+ EdgeHead & SN	79.7 / 64.2	62.3 / 34.2	227.9% / 213.8%	77.8 / 63.5	59.8 / 30.1	228.6% / 216.8%
W → N	Original	27.8 / 16.1	15.6 / 6.7	-	30.4 / 16.7	19.9 / 12.3	-
	+ EdgeHead	29.9 / 18.0	20.9 / 13.0	34.0% / 94.0%	29.7 / 17.6	21.3 / 13.8	7.0% / 12.2%
	+ ROS	26.7 / 15.4	15.8 / 6.5	1.3% / -3.0%	28.8 / 16.2	19.5 / 11.7	-2.0% / -4.9%
	+ EdgeHead & ROS	28.3 / 17.1	19.9 / 11.9	27.6% / 77.6%	29.2 / 17.4	21.3 / 13.4	7.0% / 9.3%
	+ SN	26.4 / 16.4	16.7 / 8.7	7.1% / 29.9%	29.4 / 18.0	20.5 / 12.7	3.0% / 3.3%
	+ EdgeHead & SN	28.4 / 18.6	20.7 / 13.4	32.7% / 100.0%	29.3 / 19.2	22.1 / 18.9	11.1% / 53.7%
N → K	Original	35.7 / 11.8	16.4 / 9.8	-	34.6 / 8.3	13.1 / 5.8	-
	+ EdgeHead	53.6 / 15.9	33.3 / 19.6	103.0% / 100.0%	37.0 / 10.4	19.6 / 11.5	49.6% / 98.3%
	+ ROS	43.4 / 20.0	20.2 / 8.1	23.2% / -17.3%	43.8 / 20.6	27.6 / 13.1	110.8% / 125.9%
	+ EdgeHead & ROS	52.7 / 33.1	39.9 / 24.6	143.3% / 151.0%	60.3 / 31.3	43.2 / 21.3	229.8% / 267.2%
	+ SN	29.6 / 14.3	15.7 / 8.2	-4.3% / -16.3%	33.5 / 18.1	22.0 / 11.6	67.9% / 100.0%
	+ EdgeHead & SN	45.7 / 30.4	35.1 / 23.5	114.0% / 139.8%	58.4 / 34.7	44.8 / 26.8	241.2% / 362.1%

predicts the closer surfaces better than the original SECOND model when trained on Waymo and tested on KITTI. Consistent results are observed for the other tasks and models, which demonstrates that our EdgeHead can stably shift the G_{cs} distribution to the left, *i.e.*, improve the detection quality regarding the closer surfaces.

We also plot the proportion difference to analyze models using domain adaptation methods (*i.e.*, ROS and SN; see the last two columns in Figure 3), and using our PointEdgeHead (see the last row in Figure 3), which will be further discussed in Sections 4.3.3 and 4.3.4.

4.3 Main results of the proposed EdgeHead

In this section, we extensively evaluate the models' performance before and after equipping with our EdgeHead and under different types of metrics including our proposed CS-ABS and CS-BEV metrics. We first compare the results of different models under the original BEV and 3D metrics with our proposed CS-ABS and CS-BEV metrics in Table 1 to analyze the robustness of our new metrics. Then we analyze the performance of models before and after equipping with our proposed EdgeHead, and investigate the influence of using EdgeHead and two domain adaptation methods simultaneously in Table 2. We also calculate the ***Improvement*** value as the relative difference between the used methods (*e.g.*, + EdgeHead & ROS) and the original models under the CS-ABS and CS-BEV metrics. Due to page limit, additional results are attached in *Supplementary Material* [35].

4.3.1 Evaluation on the original models

First of all, we compare the performance of two existing models, *i.e.*, SECOND and CenterPoint under four metrics as shown in Table 1. We evaluate them on three cross-domain tasks and one within-domain task on KITTI. Table 1 shows that our proposed CS-ABS and CS-BEV metrics provide results of a similar quantity level with the original BEV and 3D metrics for various models and tasks. Our metrics also show different characteristics compared with the original metrics. Taking the W → K (*i.e.*, Waymo → KITTI) task as an example, the BEV AP and 3D AP of SECOND are both lower than CenterPoint, but both the CS-ABS AP and CS-BEV AP of SECOND are higher. Therefore, when trained on Waymo and tested on KITTI, the SECOND model predicts the closer surfaces better than CenterPoint. This advantage, however, is obscured by its lower BEV AP and 3D AP scores before. Such results highlight the distinction and importance of our closer-surfaces-based evaluation metrics against the traditional ones.

4.3.2 Improvement by our EdgeHead

Table 2 presents the quantitative comparison of the performance between diffffernt models before and after equipping with our proposed EdgeHead. It shows that models equipped with EdgeHead can achieve better CS-ABS AP and CS-BEV AP than the original models in all cross-domain tasks. As described in Section 3.2, the CS-ABS AP directly measures the improvement in the detection quality of the closer surfaces, and the consistently improved performance under this metric shows that EdgeHead can stably improve the closer-surfaces detection ability of existing models across various domains. The results of the CS-ABS AP are also consistent with the proportion difference of the closer-surfaces absolute gap shown in Figure 3. Meanwhile, the improvement under the CS-BEV metric shows that EdgeHead also works well when evaluating models with a balance between the accuracy of the entire box and the closer surfaces.

Table 2 also shows that the models' performance changes much less or even remains at the original value level when evaluated under the original BEV and 3D metrics. Taking the SECOND model in the Waymo → KITTI task as an example, the BEV AP and 3D AP respectively improved by 6.3% (from 49.2 to 52.3) and 15.1% (from 9.3 to 10.7) when equipped with EdgeHead, while the CS-ABS AP and CS-BEV AP represent the improvement by 24.7% and 34.9%, respectively. We also summarize the gaps between each original model before and after equipping with EdgeHead in Table 2, which shows that similar phenomena can also be observed in other comparisons. The larger improvement shown in the CS-ABS AP and CS-BEV AP supports two important conclusions: (i) the newly proposed metrics are truly more sensitive to the closer-surfaces detection ability, and therefore can evaluate the model's cross-domain performance from a different point of view; and (ii) our EdgeHead can effectively improve the model's ability to detect the closer surfaces, which is truly helpful for applications in cross-domain tasks.

4.3.3 Combination with ROS and SN

ROS [30] and SN [24] are two domain adaptation methods that aim to solve the overfitting problems in object sizes. ROS randomly scales

the size of object boxes in both the annotations and the point cloud data to make the model more robust to object sizes. SN uses the average object size of each dataset as additional information and normalizes the source domain's object size by using the target domain's size statistics. It is therefore worth investigating the influence of such methods on the models' closer-surfaces detection ability.

We below first evaluate the CS-ABS AP and CS-BEV AP of different models equipped with these two methods (*i.e.*, ROS and SN) including further combining them with our EdgeHead. We denote these combinations as +ROS, +SN, +EdgeHead & ROS and +EdgeHead & SN in Table 2. The comparisons of the absolute gap are shown in Figure 3 as well. For most tasks, ROS and SN can help the models achieve higher BEV AP and 3D AP, but cannot stably improve the CS-ABS AP and CS-BEV AP by a similar margin. Taking SECOND in the Waymo → KITTI task as an example, ROS increases the BEV AP and 3D AP respectively by 48.4% and 311.8% (*i.e.*, from 49.2 / 9.3 to 73.0 / 38.3) but only increases the CS-ABS AP by 15.6%. In comparison, the additional use of our EdgeHead increases the performance under all four metrics, especially for the CS-ABS AP and CS-BEV AP. Taking the above example, the performance under the new metrics increases by 125.8% and 87.2% when equipping SECOND with ROS and EdgeHead simultaneously. Consistent results can also be observed for the other tasks and models in Table 2 and Figure 3. We also noticed that for both models in the Waymo → nuScenes task, ROS and SN increase the performance much less or even decrease it due to the minor object size difference between these two datasets, which is also mentioned in Yang et al. [30]. However, the performance can still be greatly improved by using our EdgeHead and ROS / SN together.

The above results demonstrate that our proposed EdgeHead can be effectively used with the existing domain adaptation methods designed for the size overfitting problem, which not only further improves the models' detection ability for the entire box but also helps achieve much better closer-surfaces detection quality compared with only using the existing domain adaptation methods.

4.3.4 *Influence of additional point-wise features*

We now compare the performance between our proposed EdgeHead and its extended version PointEdgeHead taking additional point-wise features as described in Section 3.4, see the results in Table 3 and the third column of Figure 3 where the SECOND model is utilized. Using additional point-wise features further improves the models' performance under the CS-ABS and CS-BEV metrics for most tasks, showing the point features' structural information can indeed be helpful for the closer surfaces detection. The improvement is more obvious in the nuScenes → KITTI task, which indicates that the point-wise features may play an important role when adapting models from a sparser domain to a denser domain. The improvement is however not that obvious for the Waymo → KITTI and Waymo → nuScenes tasks, and sometimes using PointEdgeHead could lead to lower performance (*e.g.*, Waymo → nuScenes without ROS or SN). Considering the extra time and resources consumed by the point feature aggregation process, it is thus unnecessary to always consider PointEdgeHead. In particular, these results show that our EdgeHead has already effectively enhanced the models' closer-surfaces detection ability with high utilization of current input information.

4.4 *Ablation study*

To further analyze our EdgeHead's refinement performance, below we propose another control-group head for us to conduct ablation study. Specifically, we maintain the module structure and the loss function design of EdgeHead, while replacing the closest vertex in EdgeHead with the center coordinates for the calculation of the location regression target. Therefore, the loss function of the control-group head reads

$$\mathcal{L}''_{\text{reg}} = \sum_{r\in\{x_c,y_c,\theta\}} \mathcal{L}_{\text{smooth}-\ell_1}(\widehat{\Delta r^a}, \Delta r^a). \tag{7}$$

In other words, the above control-group head is a simplified version of the typical refinement module as described in Eq. (4), which only refines the (BEV) location x, y, and the rotation angle θ. By comparing the performance of EdgeHead and this control-group head, we can better understand the contribution of modifying the training purpose to the closer surfaces. As shown in Table 4, although the performance of the control-group head is better than the original model that does not use any refinement head, there is a rather significant gap in terms of detection performance improvement when comparing to the excellent results of our EdgeHead. The results in Table 4 indicate that the regression target in our EdgeHead truly helps models achieve better closer-surfaces detection ability.

Table 3. Comparison between our EdgeHead and PointEdgeHead under the SECOND model.

Task	Method	+ EdgeHead $AP_{CS\text{-}BEV}$/ $AP_{CS\text{-}ABS}$	+ PointEdgeHead $AP_{CS\text{-}BEV}$/ $AP_{CS\text{-}ABS}$
W → K	Original	23.7 / 14.7	24.4 / 16.9
	+ ROS	42.9 / 20.4	47.2 / 25.0
	+ SN	62.3 / 34.2	59.0 / 33.3
W → N	Original	20.9 / 13.0	20.8 / 12.2
	+ ROS	19.9 / 11.9	21.0 / 12.7
	+ SN	20.7 / 13.4	22.0 / 14.3
N → K	Original	33.3 / 19.6	36.5 / 21.0
	+ ROS	39.9 / 24.6	46.2 / 27.3
	+ SN	35.1 / 23.5	43.6 / 28.3

Table 4. Ablation study of our EdgeHead under the SECOND model. Refine: Using a second stage refinement head. Corner: Replacing the center locations with the locations of the closest vertex in the refinement head.

Method	Refine	Corner	W → K	W → N	N → K
Original	✗	✗	19.0 / 10.9	15.6 / 6.7	16.4 / 9.8
Control-group	✓	✗	21.6 / 12.1	18.7 / 11.4	19.2 / 10.1
EdgeHead	✓	✓	23.7 / 14.7	20.9 / 13.0	33.3 / 19.6

5 Conclusion

In this paper, we novelly view the cross-domain 3D object detection problem from the detection quality of the closer surfaces to the ego vehicle. We proposed two evaluation metrics, *i.e.*, the CS-ABS AP and CS-BEV AP, to measure this detection quality and achieve a balance between the entire boxes and the closer surfaces of the objects. The proposed metrics are less sensitive to the object size difference among datasets and thus can evaluate the models' performance across domains more reasonably. Meanwhile, we equipped the existing models with our proposed EdgeHead to guide them to focus more on the closer-surfaces gaps during training. Extensive experiments show that EdgeHead can effectively help models detect better the closer surfaces and perform better under both the existing metrics and our proposed metrics. The results indicate that by guiding models to focus more on the surfaces with more points captured by the LiDAR sensor, the models can learn more robust knowledge from the training domain and perform better in cross-domain tasks.

Acknowledgements

We express sincere gratitude to Xiangyu Chen and Runwei Guan for their inspiration at the early stage of this project.

References

[1] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1090–1099, June 2022.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020. doi: 10.1109/CVPR42600.2020.01164.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia. Multi-view 3d object detection network for autonomous driving. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6526–6534, 2017. doi: 10.1109/CVPR.2017.691.

[5] Y. Chen, W. Li, and L. V. Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018. doi: 10.1109/CVPR.2018.00823.

[6] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[7] Z. Chen, Y. Luo, Z. Wang, M. Baktashmotlagh, and Z. 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.

[8] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv:2012.15712*, 2020.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

[10] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

[11] R. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. doi: 10.1109/ICCV.2015.169.

[12] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-consistent adversarial domain adaptation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/hoffman18a.html.

[13] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang. Progressive domain adaptation for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020.

[14] Q. Hu, D. Liu, and W. Hu. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[15] H. Huang, Q. Huang, and P. Krahenbuhl. Domain transfer through deep activation matching. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

[16] M. Khodabandeh, A. Vahdat, M. Ranjbar, and W. Macready. A robust learning approach to domain adaptive object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 480–490, 2019. doi: 10.1109/ICCV.2019.00057.

[17] J. Li, C. Luo, and X. Yang. Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[18] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

[19] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. doi: 10.1109/CVPR.2019.00086.

[20] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. 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 (CVPR)*, June 2020.

[21] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

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

[23] T. Wang, X. Zhang, L. Yuan, and J. Feng. Few-shot adaptive faster r-cnn. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7166–7175, 2019. doi: 10.1109/CVPR.2019.00734.

[24] Y. Wang, X. Chen, Y. You, L. E. Li, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11710–11720, 2020. doi: 10.1109/CVPR42600.2020.01173.

[25] Y. Wei, Z. Wei, Y. Rao, J. Li, J. Zhou, and J. Lu. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *European Conference on Computer Vision*, pages 179–195. Springer, 2022.

[26] G. Wu, T. Cao, B. Liu, X. Chen, and Y. Ren. Towards universal lidar-based 3d object detection by multi-domain knowledge transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8669–8678, 2023.

[27] B. Xuyang, H. Zeyu, Z. Xinge, H. Qingqiu, C. Yilun, F. Hongbo, and C.-L. Tai. TransFusion: Robust Lidar-Camera Fusion for 3d Object Detection with Transformers. *CVPR*, 2022.

[28] Y. Yan, Y. Mao, and B. Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10), 2018. ISSN 1424-8220. doi: 10.3390/s18103337. URL https://www.mdpi.com/1424-8220/18/10/3337.

[29] B. Yang, W. Luo, and R. Urtasun. Pixor: Real-time 3d object detection from point clouds. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. doi: 10.1109/CVPR.2018.00798.

[30] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10363–10373, 2021. doi: 10.1109/CVPR46437.2021.01023.

[31] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[32] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11784–11793, June 2021.

[33] J. Yuan, B. Zhang, X. Yan, T. Chen, B. Shi, Y. Li, and Y. 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.

[34] B. Zhang, J. Yuan, B. Shi, T. Chen, Y. Li, and Y. Qiao. Uni3d: A unified baseline for multi-dataset 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9253–9262, 2023.

[35] R. Zhang, Y. Wu, J. Lee, A. Prugel-Bennett, and X. Cai. Detect closer surfaces that can be seen: New modeling and evaluation in cross-domain 3d object detection. *arXiv preprint arXiv:2407.04061*, 2024.

[36] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. doi: 10.1109/CVPR.2018.00472.