

Multi-Modal Assistance for Unsupervised Domain Adaptation on Point Cloud 3D Object Detection

Shenao Zhao*, Pengpeng Liang*[†], Zhoufan Yang

School of Computer Science and Artificial Intelligence, Zhengzhou University, China
1722236579@qq.com, liangpcs@gmail.com, 2829149697@qq.com

Abstract

Unsupervised domain adaptation for LiDAR-based 3D object detection (3D UDA) based on the teacher-student architecture with pseudo labels has achieved notable improvements in recent years. Although it is quite popular to collect point clouds and images simultaneously, little attention has been paid to the usefulness of image data in 3D UDA when training the models. In this paper, we propose an approach named MMAssist that improves the performance of 3D UDA with multi-modal assistance. A method is designed to align 3D features between the source domain and the target domain by using image and text features as bridges. More specifically, we project the ground truth labels or pseudo labels to the images to get a set of 2D bounding boxes. For each 2D box, we extract its image feature from a pre-trained vision backbone. A large vision-language model (LVLM) is adopted to extract the box’s text description, and a pre-trained text encoder is used to obtain its text feature. During the training of the model in the source domain and the student model in the target domain, we align the 3D features of the predicted boxes with their corresponding image and text features, and the 3D features and the aligned features are fused with learned weights for the final prediction. The features between the student branch and the teacher branch in the target domain are aligned as well. To enhance the pseudo labels, we use an off-the-shelf 2D object detector to generate 2D bounding boxes from images and estimate their corresponding 3D boxes with the aid of point cloud, and these 3D boxes are combined with the pseudo labels generated by the teacher model. Experimental results show that our approach achieves promising performance compared with state-of-the-art methods in three domain adaptation tasks on three popular 3D object detection datasets.

Code — <https://github.com/liangp/MMAssist>

Introduction

3D object detection from point clouds is an important task in autonomous driving, and it has made significant progress in recent years with the availability of large-scale annotated data (Caesar et al. 2020; Sun et al. 2020). However, due to

*These authors contributed equally.

[†]Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

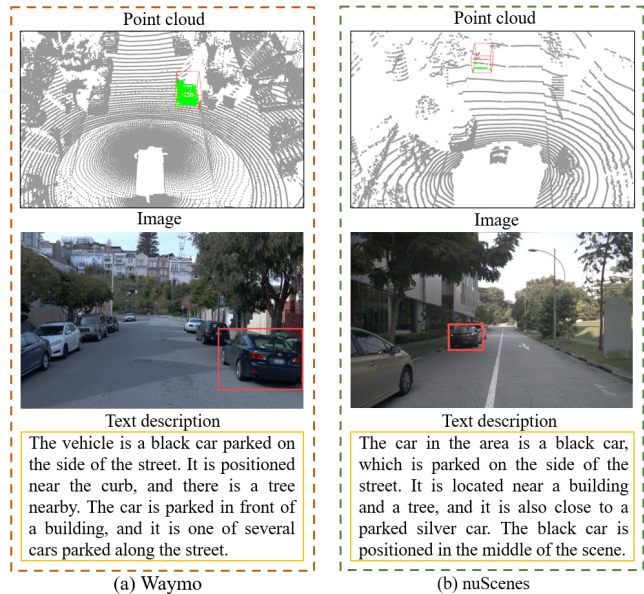


Figure 1: Comparison of point cloud, image, and object’s text description generated by LLaVA (Liu et al. 2023) of two objects having similar appearances in the images. (a) is from Waymo (Sun et al. 2020), and (b) is from nuScenes (Caesar et al. 2020). Best viewed in color.

the discrepancies between domains caused by varying LiDAR beams, diverse environments, etc, 3D object detectors trained in the source domain often encounter difficulties when applied to the target domain (Chang et al. 2024; Wang et al. 2020; Wei et al. 2022; Zhang et al. 2024c).

Unsupervised domain adaptation aims at transferring the model trained in the source domain to the target domain, in which no data is annotated. To improve the performance of LiDAR-based 3D UDA, one strategy is to reduce the data or feature distribution gap between the two domains (Wang et al. 2020; Hu, Liu, and Hu 2023). SN (Wang et al. 2020) normalizes the source domain data based on the difference in mean car sizes between the two domains. DTS (Hu, Liu, and Hu 2023) uses random beam re-sampling to augment the data of both domains to reduce the beam density gap. Another strategy is to enhance the pseudo label quality under the self-training framework, which uses the source domain model or a teacher model to generate pseudo labels for

the target domain (Yang et al. 2022; Zhang et al. 2024c). ST3D++ (Yang et al. 2022) designs a hybrid quality-aware triplet memory to improve the quality of pseudo labels. PERE (Zhang et al. 2024c) mitigates the impact of unreliable pseudo labels with box replacement and point removal.

It is a common practice to collect LiDAR point cloud and image data simultaneously in autonomous driving (Caesar et al. 2020; Sun et al. 2020; Geiger, Lenz, and Urtasun 2012), but most of the current state-of-the-art approaches (Hu, Liu, and Hu 2023; Shin et al. 2024; Yihan et al. 2021; Zhang et al. 2024c; Yang et al. 2022) solely depend on the LiDAR data without exploring the assistance of image data. CMDA (Chang et al. 2024) transfers semantic knowledge from images to the 3D detector of the source domain, but the usefulness of images is not directly exploited in the self-training of the target domain model. Besides image data, the recent progress of large vision-language model (LVLM) (Liu et al. 2023; Wang et al. 2024) enables the extraction of accurate text descriptions of objects from images. Although LVLM has been successfully adopted by point cloud pre-training (Xue et al. 2024), its effectiveness for LiDAR-based 3D UDA has not been well studied.

In this paper, we present MMAssist that assists LiDAR-based 3D UDA with multi-modal information (image and text), and our approach is based on the teacher-student self-training method DTS (Hu, Liu, and Hu 2023). As vision models trained with large-scale data show strong generalization ability (Lavoie, Mahmoud, and Waslander 2025), compared with the large discrepancy with regard to point clouds, image features of similar objects in different domains are expected to have a smaller domain gap. Meanwhile, LVLM generates similar text descriptions for similar objects in two domains as shown in Fig. 1. So, instead of directly aligning features between two domains, which is difficult when the training processes of two domains are separate, we propose to use image and text features as bridges to align features across two domains. Given the ground truth labels in the source domain and pseudo labels in the target domain, we project the 3D bounding boxes to images to generate a set of 2D bounding boxes. For each 2D bounding box, we use RoIAlign (He et al. 2017) to extract its image feature from a pre-trained vision backbone and use an LVLM to generate its text description, then the text description is encoded into the text feature with a pre-trained text encoder. During the training in the source domain and the training of the student model in the target domain, when a predicted 3D bounding box matches a ground truth label (or pseudo label), we extract its 3D feature from the detector backbone and align it with its corresponding image and text features by minimizing losses based on cosine similarity. We fuse the 3D feature and its aligned features in a weighted manner to predict the final 3D bounding box. In the target domain, when a detected 3D box of the student branch matches a detected 3D box of the teacher branch, their 3D features are aligned as well.

To improve the quality of pseudo labels, following (Zhang et al. 2024b), we use the off-the-shelf 2D object detector GroundingDINO (Liu et al. 2024) to generate 2D bounding boxes from images and lift them to 3D space with geometric reasoning (Wei et al. 2021). Based on the observation that

the 3D object detector trained in the source domain, in general, generates more accurate pseudo labels than the pseudo labels reasoned from 2D bounding boxes, but it has difficulty in detecting long-range 3D objects, we add 3D pseudo labels from images in the long-range area where the distance is larger than a threshold. And when there are two pseudo labels generated by the above two approaches with overlap larger than a threshold, we keep the pseudo labels generated by the 3D detector while discarding the other.

We follow (Hu, Liu, and Hu 2023) and integrate our approach into three popular LiDAR-based object detectors (PV-RCNN (Shi et al. 2020), SECOND-IoU (Yang et al. 2021), and PointPillars (Lang et al. 2019)), and their performances are evaluated on three domain adaptation tasks using nuScenes (Caesar et al. 2020), KITTI (Geiger, Lenz, and Urtasun 2012), and Waymo (Sun et al. 2020). The combination of three detectors and the three domain adaptation tasks results in nine subtasks, and our approach performs best in seven of them in general.

The contributions of our paper are summarized as follows:

- We propose to assist in LiDAR-based 3D UDA with multi-modal information with respect to both cross-domain feature alignment and pseudo label quality.
- We design a method that uses image and text features as bridges for feature alignment between the source domain and the target domain.
- The performance of MMAssist is inspiring, and the effectiveness of each component of our approach is validated via ablation study.

Related Work

Unsupervised Domain Adaptation for 3D Object Detection

A popular unsupervised domain adaptation approach for point cloud-based 3D object detection is pseudo label-based self-training, and current research efforts are mainly focused on improving the quality of pseudo labels and aligning the data distributions or feature representations across domains. ST3D++ (Yang et al. 2022) applies random object scaling during the pre-training phase and designs a voting method based on historical pseudo labels to reduce pseudo label noise. To reduce the sensitivity regarding the beam density variation, DTS (Hu, Liu, and Hu 2023) presents a random beam re-sampling module and aligns features between the teacher and student branches with an object graph alignment module. PERE (Zhang et al. 2024c) refines pseudo labels by randomly removing or replacing them based on prediction confidences and aligns instance features across domains with a triplet loss. MLC-Net (Luo et al. 2021) trains the student branch with both source domain labels and target domain pseudo labels, and it enhances the consistency between the teacher and student branches at the point, instance, and batch normalization levels. ReDB (Chen et al. 2023) generates reliable pseudo labels with cross-domain examination and downsampling based on overlapped box counting. CMDA (Chang et al. 2024) mixes the source point cloud and target point cloud during self-training and

learns domain-invariant features via instance-level adversarial learning. DALI (Lu and Radha 2024) enhances the quality of pseudo labels generated by the source domain model via scaling the coordinates of target domain point clouds, and synthetic point clouds are used to improve the consistency between pseudo labels and their corresponding point clouds.

In addition to the above self-training methods, (Wei et al. 2022) generates low-beam pseudo point clouds through downsampling and transfers the model trained on the high-beam source domain to the low-beam target domain using knowledge distillation. 3D-CoCo (Yihan et al. 2021) trains the target domain model with pseudo labels and updates the source domain model with ground-truth labels simultaneously, and cross-domain instance features are aligned through contrastive learning during the co-training process. CMT (Chen et al. 2024) constructs a hybrid domain that leverages both the source and target domains to align features. GPA-3D (Li et al. 2023) aligns the BEV features of the source and target domains to the same set of geometry-aware prototypes to reduce domain discrepancy. GroupEXPDA (Shin et al. 2024) clusters the labels into groups that are shared between domains and learns group-equivariant spatial features. GBlobs (Malić et al. 2025) encodes point cloud neighborhoods with Gaussian blobs to improve the generalization ability of detectors.

LiDAR-based 3D Object Detection

LiDAR-based 3D object detection aims to localize and classify objects from sparse and unordered point clouds. Point-based methods (Shi, Wang, and Li 2019; Yang et al. 2020; Shi and Rajkumar 2020) directly operate on raw point clouds, and they typically employ hierarchical networks such as PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b) to extract point-wise features. Voxel-based methods (Zhou and Tuzel 2018; Deng et al. 2021; Yan, Mao, and Li 2018; Lang et al. 2019; Fan et al. 2022; Zhang et al. 2024a) convert unstructured point clouds into regularized 2D/3D grids and carry out detection based on the voxel representation. Meanwhile, PV-RCNN (Shi et al. 2020) leverages both point-based and voxel-based representations by combining keypoints with nearby voxels.

Method

Problem Statement and Overview

Problem statement. Unsupervised domain adaptation for 3D object detection aims to adapt a model trained on a labeled source domain dataset $D^S = \{(P_i^S, Y_i^S)\}_{i=1}^{N_S}$ to an unlabeled target domain dataset $D^T = \{P_i^T\}_{i=1}^{N_T}$. P_i^S and Y_i^S represent the i -th point cloud sample from the source domain and its corresponding annotation, respectively. The label Y_i^S includes object category information and 3D bounding box parameters. Each 3D bounding box is defined by its length l , width w , height h , center coordinate (x, y, z) , and heading angle θ . P_i^T denotes the i -th point cloud sample from the target domain, which does not have ground truth labels.

Overview. As shown in Fig. 2, our approach MMAssist is based on the teacher-student framework. The source point cloud and the target point cloud of the student branch are augmented with beam re-sampling (Hu, Liu, and Hu 2023) so that they have similar densities. In the pre-training stage, we use D^S to train the source model M^S . In the self-training stage, M^S is used to initialize the teacher model M^{teacher} and the student model M^{student} . We use pseudo labels to train M^{student} , and the parameters of M^{teacher} are updated with the parameters of M^{student} using exponential moving average (EMA), i.e., $M^{\text{teacher}} = \epsilon M^{\text{teacher}} + (1 - \epsilon) M^{\text{student}}$. During both the pre-training stage and the self-training stage, we align the 3D features of objects with their corresponding image and text features to learn domain-invariant feature representations, which are further fused with the 3D features in a weighted manner to make the final prediction. In the self-training stage, the pseudo labels generated by the teacher model and pseudo labels generated based on 2D detection results from images associated with D^T are combined to train the student model. Features between the student branch and the teacher branch are aligned as well. It is worth noting that image and text information is only used in the training stage, and the test stage solely uses point clouds as input.

Cross-Domain Feature Alignment with Image and Text Features as Bridges

In this section, we introduce the method for aligning features between two domains with image and text features as bridges. The alignment is performed in two domains individually.

Image and text features generation. Given a point cloud sample P from the source domain (or target domain) and its corresponding 3D ground truth labels (or pseudo labels) $B = \{\mathbf{b}_i = (x_i, y_i, z_i, l_i, w_i, h_i, \theta_i)\}_{i=1}^{N^{3D}}$, we use a camera’s intrinsic matrix K and extrinsic matrix E to project each \mathbf{b}_i to its 2D image plane. We filter out the 2D projected bounding boxes outside the image. For the multi-camera setting, after projecting \mathbf{b}_i to multiple image planes, we calculate the truncation rates of 2D projected bounding boxes and only retain the 2D box with the minimum truncation rate. We finally obtain a set of 2D bounding boxes $B^{2D} = \{\mathbf{b}_i^{2D} = (x_i^{\min}, y_i^{\min}, x_i^{\max}, y_i^{\max})\}_{i=1}^{N^{2D}}$, where $(x_i^{\min}, y_i^{\min}, x_i^{\max}, y_i^{\max})$ indicates the coordinates of bounding box \mathbf{b}_i^{2D} ’s top-left and bottom-right corners.

For each \mathbf{b}_i in B that has a corresponding \mathbf{b}_i^{2D} in B^{2D} , we first use RoIAlign (He et al. 2017) with \mathbf{b}_i^{2D} to extract its image feature $\mathbf{f}_i^{\text{img}} \in \mathbb{R}^{C^{\text{img}}}$ from the feature maps of \mathbf{b}_i^{2D} ’s corresponding image I , the feature maps of which are generated by a pre-trained vision backbone. Then, we use a large vision-language model (LVLM) to extract its text description. More specifically, we construct a text prompt $p_i = \text{“There is a \{class\} in the area } (x_i^{\min}, y_i^{\min}, x_i^{\max}, y_i^{\max}), \text{ please describe the characteristics of this \{class\}.”}$, where class is the category of \mathbf{b}_i . Then, p_i and the image I in which \mathbf{b}_i^{2D} is located are fed into the LVLM to produce the text description d_i . Next, we use a pre-trained text encoder to extract the text feature $\mathbf{f}_i^{\text{text}} \in \mathbb{R}^{C^{\text{text}}}$

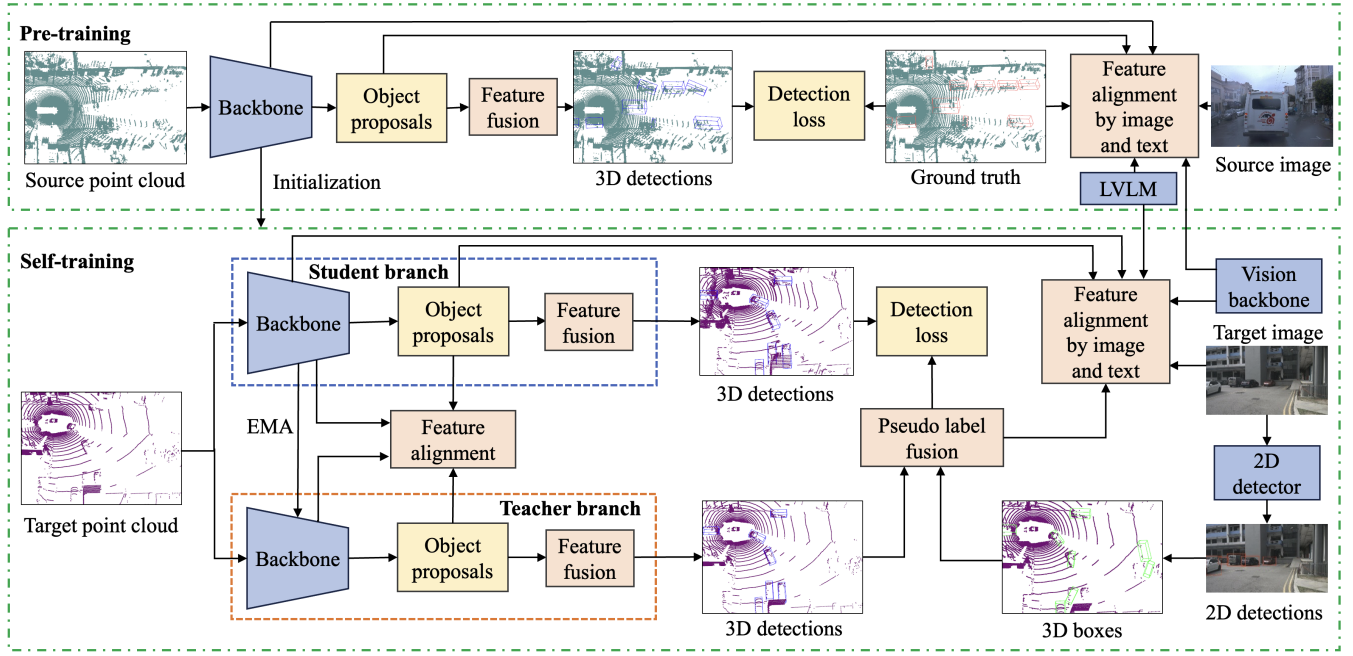


Figure 2: Overall approach pipeline. Features of the source domain model and features of the student model in the target domain are aligned with the help of their corresponding image and text features. The aligned features are fused with the 3D features to make the final prediction (the input features to the feature fusion module are not shown). 3D bounding boxes estimated from the 2D detection results are used to enhance the pseudo labels. The teacher model is updated by the student model via EMA.

for d_i . We choose the backbone of GroundingDINO (Liu et al. 2024), LLaVA (Liu et al. 2023), and SLIP (Mu et al. 2022) as the vision backbone, LVLm, and text encoder, respectively.

Aligning 3D feature with image and text features. For both the training of the source model in the pre-training stage and the training of the student model in the self-training stage, given a point cloud sample P , we first obtain the initial predicted bounding boxes $\hat{B} = \{\hat{\mathbf{b}}_i = (x_i, y_i, z_i, l_i, w_i, h_i, \theta_i)\}_{i=1}^L$. For each $\hat{\mathbf{b}}_i$, we extract its 3D feature $\hat{\mathbf{f}}_i^{3D} \in \mathbb{R}^{C^{3D}}$ from the backbone of the detector (detail about the feature extraction is in the supplementary material of the arxiv version). Then, $\hat{\mathbf{f}}_i^{3D}$ is converted to $\hat{\mathbf{f}}_i^{img} \in \mathbb{R}^{C^{img}}$ and $\hat{\mathbf{f}}_i^{text} \in \mathbb{R}^{C^{text}}$ by MLP, and they are used for feature alignment with image and text features, respectively.

To find the corresponding image and text features of $\hat{\mathbf{b}}_i$, which we align $\hat{\mathbf{f}}_i^{img}$ and $\hat{\mathbf{f}}_i^{text}$ with, we match $\hat{\mathbf{b}}_i$ to a 3D ground truth bounding box using the following formula:

$$\mathbf{b}_m = \arg \max_{\mathbf{b}_j \in B} \text{IoU}(\hat{\mathbf{b}}_i, \mathbf{b}_j) \quad (1)$$

\mathbf{b}_m is the 3D ground truth bounding box in B that has the highest IoU with $\hat{\mathbf{b}}_i$ (ground truth is the pseudo labels in the self-training stage). If $\text{IoU}(\hat{\mathbf{b}}_i, \mathbf{b}_m) \geq \mu$, we consider $(\hat{\mathbf{b}}_i, \mathbf{b}_m)$ a matched pair. And if \mathbf{b}_m has a matched 2D bounding box \mathbf{b}_m^{2D} , we use \mathbf{b}_m^{2D} 's image feature \mathbf{f}_m^{img} and text feature \mathbf{f}_m^{text} as $\hat{\mathbf{b}}_i$'s image and text features for alignment, respectively, i.e., $\hat{\mathbf{g}}_i^{img} = \mathbf{f}_m^{img}$ and

$\hat{\mathbf{g}}_i^{text} = \mathbf{f}_m^{text}$. Finally, we construct a subset $\hat{B}' = \{\hat{\mathbf{b}}_i = (x_i, y_i, z_i, l_i, w_i, h_i, \theta_i)\}_{i=1}^{L'}$ of \hat{B} , where $L' \leq L$, and each $\hat{\mathbf{b}}_i$ in \hat{B}' has its corresponding image feature $\hat{\mathbf{g}}_i^{img}$ and text feature $\hat{\mathbf{g}}_i^{text}$.

To align with image features, we first randomly generate a set of background 2D bounding boxes $B_{bg}^{2D} = \{\mathbf{b}_i^{bg} = (x_i^{min}, y_i^{min}, x_i^{max}, y_i^{max})\}_{i=1}^{N^{bg}}$ which don't intersect with any 2D box in B^{2D} , and the image feature \mathbf{g}_i^{bg} of each background box is extracted with RoAlign as well. Then, for each $\hat{\mathbf{b}}_i$ in \hat{B}' , we pull its $\hat{\mathbf{f}}_i^{img}$ derived from its 3D feature $\hat{\mathbf{f}}_i^{3D}$ closer to its image feature $\hat{\mathbf{g}}_i^{img}$ and push it away from image features of background boxes. The corresponding loss is formulated as:

$$\mathcal{L}_{align}^{img} = \frac{1}{L'} \sum_{i=1}^{L'} \max \left(\frac{1}{N^{bg}} \sum_{j=1}^{N^{bg}} \text{sim}(\hat{\mathbf{f}}_i^{img}, \mathbf{g}_j^{bg}) - \text{sim}(\hat{\mathbf{f}}_i^{img}, \hat{\mathbf{g}}_i^{img}) + \sigma, 0 \right) \quad (2)$$

where $\text{sim}(\mathbf{a}, \mathbf{b}) = \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$ is the cosine similarity, and σ is the margin.

To align with text feature, for each $\hat{\mathbf{b}}_i$, we minimize the distance between $\hat{\mathbf{f}}_i^{text}$ and its text feature $\hat{\mathbf{g}}_i^{text}$, where $\hat{\mathbf{f}}_i^{text}$ is derived from $\hat{\mathbf{b}}_i$'s 3D feature. Below is the loss based on cosine similarity:

$$\mathcal{L}_{align}^{text} = \frac{1}{L'} \sum_{i=1}^{L'} \left(1 - \text{sim}(\hat{\mathbf{f}}_i^{text}, \hat{\mathbf{g}}_i^{text}) \right) \quad (3)$$

The final loss function of the pre-training stage is a combination of the detector’s original loss \mathcal{L}_{det} , $\mathcal{L}_{\text{align}}^{\text{text}}$, and $\mathcal{L}_{\text{align}}^{\text{img}}$:

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{det}} + \alpha \mathcal{L}_{\text{align}}^{\text{text}} + \beta \mathcal{L}_{\text{align}}^{\text{img}} \quad (4)$$

Fusing 3D features with aligned features. Given a 3D bounding box $\hat{\mathbf{b}}^s$ ’s 3D feature $\mathbf{f}^{3D} \in \mathbb{R}^{C^{3D}}$, image-aligned feature $\mathbf{f}^{\text{img}} \in \mathbb{R}^{C^{\text{img}}}$, and text-aligned feature $\mathbf{f}^{\text{text}} \in \mathbb{R}^{C^{\text{text}}}$, we fuse them in a weighted manner. More specifically, we first use two MLPs to change the dimensionality of \mathbf{f}^{img} and \mathbf{f}^{text} to C^{3D} , respectively, then we concatenate the three feature vectors and input it to another MLP to learn a weight vector $\mathbf{w} \in \mathbb{R}^3$. The final fused feature is obtained by:

$$\mathbf{f}^{\text{fused}} = \mathbf{w}_0 \mathbf{f}^{3D} + \mathbf{w}_1 \mathbf{f}^{\text{img}} + \mathbf{w}_2 \mathbf{f}^{\text{text}} \quad (5)$$

After getting $\mathbf{f}^{\text{fused}}$, for PV-RCNN (Shi et al. 2020), we use it to refine the proposal from the first stage; for PointPillar (Lang et al. 2019), we use it to refine the initial detection result; for SECOND-IoU (Yang et al. 2021), we use it to predict the IoU score in the second stage.

Aligning 3D Features between Student and Teacher Branches

In the self-training stage, we follow (Hu, Liu, and Hu 2023) and further align the 3D features extracted from the student model and the teacher model. Given the predicted bounding boxes $\hat{B}^S = \{\hat{\mathbf{b}}_i^S\}_{i=1}^{L^S}$ from the student model and $\hat{B}^T = \{\hat{\mathbf{b}}_j^T\}_{j=1}^{L^T}$ from the teacher model, we extract their 3D RoI features $\hat{F}^S = \{\hat{\mathbf{f}}_i^S\}_{i=1}^{L^S}$ and $\hat{F}^T = \{\hat{\mathbf{f}}_j^T\}_{j=1}^{L^T}$ from the backbones of their detectors. For each $\hat{\mathbf{b}}_i^S$, we follow Eq. 1 to find $\hat{\mathbf{b}}_m^T$ in \hat{B}^T that has the largest IoU with $\hat{\mathbf{b}}_i^S$. If $\text{IoU}(\hat{\mathbf{b}}_i^S, \hat{\mathbf{b}}_m^T) \geq \eta$, we obtain a matched pair $(\hat{\mathbf{b}}_i^S, \hat{\mathbf{b}}_m^T)$. Finally, we obtain a set of matched pairs $\hat{B}^{\text{match}} = \{(\hat{\mathbf{b}}_i^S, \hat{\mathbf{b}}_i^T)\}_{i=1}^G$ of \hat{B}^S , where $G \leq L^S$. For each pair in \hat{B}^{match} , we align their 3D features via:

$$\mathcal{L}_{\text{ST}} = \frac{1}{G} \sum_{i=1}^G (1 - \text{sim}(\hat{\mathbf{f}}_i^S, \hat{\mathbf{f}}_i^T)) \quad (6)$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

The loss for training the student model in the self-training stage is a combination of the detector’s original loss \mathcal{L}_{det} , $\mathcal{L}_{\text{align}}^{\text{img}}$ in Eq. 2, $\mathcal{L}_{\text{align}}^{\text{text}}$ in Eq. 3, and the above \mathcal{L}_{ST} :

$$\mathcal{L}_{\text{student}} = \mathcal{L}_{\text{det}} + \alpha \mathcal{L}_{\text{align}}^{\text{text}} + \beta \mathcal{L}_{\text{align}}^{\text{img}} + \gamma \mathcal{L}_{\text{ST}} \quad (7)$$

Enhancing 3D Pseudo Labels with 3D Boxes from Images

To generate 3D pseudo labels from images, we follow (Zhang et al. 2024b) and obtain a set of 3D bounding boxes $\hat{B}_{\text{img}}^{3D} = \{\hat{\mathbf{b}}_i^{3D}\}_{i=1}^H$ based on 2D detection bounding boxes from the off-the-shelf open-vocabulary 2D detector GroundingDINO (Liu et al. 2024). During the self-training stage, we select 3D pseudo labels from $\hat{B}_{\text{img}}^{3D}$ based on their

distances and overlaps with pseudo labels generated by the teacher model. The selection can be formulated as:

$$\hat{B}_{\text{img}}^{\text{sel}} = \left\{ \hat{\mathbf{b}}_i^{3D} \mid \hat{\mathbf{b}}_i^{3D} \in \hat{B}_{\text{img}}^{3D}, \text{dist}(\hat{\mathbf{b}}_i^{3D}) \geq \tau, \max_{\hat{\mathbf{b}}_j \in B_{\text{teacher}}} \left(\text{IoU}(\hat{\mathbf{b}}_i^{3D}, \hat{\mathbf{b}}_j) \right) \leq \xi \right\}. \quad (8)$$

where B_{teacher} contains the pseudo labels generated by the teacher model, $\text{dist}(\cdot)$ obtains the distance of a 3D bounding box. The final set of pseudo labels is $B_{\text{pseudo}} = \hat{B}_{\text{img}}^{\text{sel}} \cup B_{\text{teacher}}$.

Experiment

Experimental Settings

Datasets. We evaluate our proposed MMAassist on three widely used 3D object detection datasets for autonomous driving: Waymo (Sun et al. 2020), nuScenes (Caesar et al. 2020), and KITTI (Geiger, Lenz, and Urtasun 2012). The point cloud data of Waymo dataset is captured by five LiDAR sensors, i.e., one 64-beam LiDAR and four 200-beam LiDARs, and the image data of Waymo is collected with five surrounding cameras. The nuScenes dataset provides point clouds captured by a 32-beam roof LiDAR, and its image data is collected by six surrounding cameras. KITTI uses a 64-beam LiDAR to collect point clouds, and it provides a pair of images captured by stereo cameras. Each dataset has its unique characteristics in sensor configuration, environment, etc. We consider each dataset as a separate domain and evaluate our MMAassist by adapting detectors with it across domains. We follow (Hu, Liu, and Hu 2023) and evaluate our MMAassist with the following domain adaptation tasks: Waymo \rightarrow nuScenes (W \rightarrow N), Waymo \rightarrow KITTI (W \rightarrow K), and nuScenes \rightarrow KITTI (N \rightarrow K).

Evaluation metric. Following (Hu, Liu, and Hu 2023), we adopt the KITTI evaluation metric and perform evaluations on the commonly used car category (the vehicle category in Waymo). We report the average precision (AP) with respect to BEV IoUs and 3D IoUs over 40 recall positions with IoU threshold of 0.7. The domain adaptation metric Closed Gap (Yang et al. 2021) is reported as well, and it is defined as $\text{Closed Gap} = \frac{\text{AP}_{\text{model}} - \text{AP}_{\text{source}}}{\text{AP}_{\text{oracle}} - \text{AP}_{\text{source}}} \times 100\%$.

Implementation details. The effectiveness of MMAassist is validated with SECOND-IoU (Yang et al. 2021), PV-RCNN (Shi et al. 2020), and PointPillars (Lang et al. 2019). We adopt the training settings of the popular point cloud detection codebase OpenPCDet (Team 2020) to pre-train the detectors in the source domain with feature alignment. The weights α and β for the losses of feature alignment with image and text features are both set to 0.3 in the pre-training stage and both are set to 0.03 in the self-training stage. The weight γ of the feature alignment loss between teacher and student branches is set to 0.1. The IoU threshold μ used to match the 3D predictions and labels is set to 0.5. For the self-training stage in the target domain, we use Adam (Kingma and Ba 2014) and one cycle scheduler to fine-tune the detectors for 30 epochs, and the initial learning rate is set to 1.5×10^{-3} . The smoothing coefficient ϵ of EMA is set to 0.999. The IoU threshold η for matching the predictions of

Task	Method	PV-RCNN (Shi et al. 2020)		PointPillars (Lang et al. 2019)		SECOND-IoU (Yang et al. 2021)	
		AP _{BEV} /AP _{3D}	Closed Gap	AP _{BEV} /AP _{3D}	Closed Gap	AP _{BEV} /AP _{3D}	Closed Gap
N→K	Source Only	68.2/37.2	-/-	22.8/0.5	-/-	51.8/17.9	-/-
	SN [†] (Wang et al. 2020)	60.5/49.5	+36.8%/+27.1%	39.3/2.0	+26.6%/+2.1%	59.7/37.6	+25.1%/+35.4%
	ST3D (Yang et al. 2021)	78.4/70.9	+49.0%/+74.3%	60.4/11.1	+60.6%/+14.9%	75.9/54.1	+76.6%/+59.5%
	DTS (Hu, Liu, and Hu 2023)	83.9/71.8	+75.8%/+76.4%	79.5/51.8	+91.5%/+72.2%	81.4/66.6	+94.0%/+87.6%
	CMDA (Chang et al. 2024)	84.9/75.0	+80.3%/+83.4%	-/-	-/-	82.1/69.0	+96.2%/+92.0%
	CMT (Chen et al. 2024)	85.8/75.5	+84.6%/+84.5%	81.4/57.6	+94.5%/+80.3%	83.0/68.1	+99.0%/+90.3%
	GroupEXP-DA (Shin et al. 2024)	-/-	-/-	81.9/52.8	+95.3%/+73.6%	81.5/68.2	+94.3%/+90.5%
	Ours	86.8/78.1	+89.4%/+90.3%	81.9/60.4	+95.3%/+84.2%	84.8/69.8	+104.8%/+93.3%
Oracle	89.0/82.5	-/-	84.8/71.6	-/-	83.3/73.5	-/-	
W→K	Source Only	61.2/22.0	-/-	47.8/11.5	-/-	67.6/27.5	-/-
	SN [†] (Wang et al. 2020)	79.8/63.6	+66.9%/+68.7%	27.4/6.4	-55.1%/-8.5%	79.0/59.2	+72.3%/+69.0%
	ST3D (Yang et al. 2021)	84.1/64.8	+82.4%/+70.7%	58.1/23.2	+27.8%/+19.5%	82.2/61.8	+93.0%/+74.7%
	DTS (Hu, Liu, and Hu 2023)	86.4/68.1	+90.6%/+76.2%	76.1/50.2	+76.5%/+64.4%	85.8/71.5	+115.9%/+95.7%
	CMT (Chen et al. 2024)	85.9/ 74.5	+88.8%/+ 86.8%	77.4/51.5	+80.0%/+66.6%	85.2/72.1	+112.1%/+97.0%
	GroupEXP-DA (Shin et al. 2024)	-/-	-/-	78.4/54.1	+82.7%/+70.9%	86.9/73.7	+122.9%/+100.4%
	Ours	87.6/72.7	+95.0%/+83.8%	81.4/56.8	+90.8%/+75.4%	86.9/70.4	+122.9%/+93.3%
	Oracle	89.0/82.5	-/-	84.8/71.6	-/-	83.3/73.5	-/-
W→N	Source Only	34.5/21.5	-/-	27.8/12.1	-/-	32.9/17.2	-/-
	SN [†] (Wang et al. 2020)	34.2/22.3	-1.5%/+4.8%	28.3/13.0	+2.4%/+4.7%	33.2/18.6	+1.7%/+7.5%
	ST3D (Yang et al. 2021)	36.4/23.0	+10.3%/+8.8%	30.6/15.6	+13.2%/+18.2%	35.9/20.2	+15.9%/+16.7%
	L. D. (Wei et al. 2022)	43.3/25.6	+47.3%/+24.0%	40.2/19.1	+58.4%/+36.5%	40.7/22.9	+41.1%/+32.2%
	DTS (Hu, Liu, and Hu 2023)	44.0/26.2	+51.1%/+27.5%	42.2/21.5	+67.9%/+49.0%	41.2/23.0	+43.7%/+32.8%
	CMDA (Chang et al. 2024)	44.4/26.4	+53.2%/+28.7%	-/-	-/-	42.8/24.6	+52.1%/+41.8%
	CMT (Chen et al. 2024)	41.7/26.4	+38.7%/+28.7%	38.0/21.1	+48.1%/+46.9%	40.8/24.3	+41.6%/+40.1%
	GroupEXP-DA (Shin et al. 2024)	-/-	-/-	44.3/22.2	+77.8%/+52.6%	43.8/24.4	+57.4%/+40.7%
Ours	45.5/27.0	+59.1%/+32.2%	42.5/21.5	+69.3%/+49.0%	45.1/24.6	+64.2%/+41.8%	
Oracle	53.1/38.6	-/-	49.0/31.3	-/-	51.9/34.9	-/-	

Table 1: Comparison with state-of-the-art approaches. “Source Only” means the pre-trained model in the source domain is directly applied to the target domain. “Oracle” indicates that the detector is trained with the labeled target domain data. † means SN is weakly supervised with target domain statistics. The best performances are in bold.

the teacher and student models is set to 0.5. The IoU threshold ξ and distance threshold τ used for selecting 3D labels generated from images are set to 0.5 and 30, respectively.

Speed comparison. Since the feature alignment is only performed in the training stage, our approach adds little extra computation burden to the test stage. With a single RTX 4090 GPU, the test speeds of the baseline SECOND-IoU, PV-RCNN, PointPillars, and their MMAssist-enhanced versions (in the parentheses) are 52.04 FPS (51.40 FPS), 6.67 FPS (6.65 FPS), and 82.55 FPS (79.43 FPS), respectively, where FPS is short for frames per second.

Comparison with State-of-the-Art

We compare our approach MMAssist with SN (Wang et al. 2020), ST3D (Yang et al. 2021), L.D (Wei et al. 2022), DTS (Hu, Liu, and Hu 2023), CMDA (Chang et al. 2024), CMT (Chen et al. 2024) and GroupEXP-DA (Shin et al. 2024). As shown in Table 1, when combined with PV-RCNN, on N → K, MMAssist surpasses the second-best method CMT by 1.0% AP_{BEV} and 2.6% AP_{3D}. On W → K, MMAssist is 1.7% AP_{BEV} better than CMT while CMT is 1.8% AP_{3D} better than ours. On W → N, our approach outperforms the second-best CMDA by 1.1% AP_{BEV} and 0.6% AP_{3D}. The performance of MMAssist with PointPillars is also remarkable. On N → K, MMAssist and GroupEXP-DA both achieve the best 81.9% AP_{BEV}, but our approach

is 7.6% AP_{3D} better than GroupEXP-DA. Ours also outperforms second-best method GroupEXP-DA by 3.0% AP_{BEV} and 2.7% AP_{3D} on W → K. On W → N, GroupEXP-DA performs best and excels our approach by 1.8% AP_{BEV} and 0.7% AP_{3D}. In terms of SECOND-IoU, on N → K, our approach outperforms the overall second best CMT by 1.8% AP_{BEV} and 1.7% AP_{3D}. On W → K, both MMAssist and GroupEXP-DA show the best performance regarding AP_{BEV}, but GroupEXP-DA outperforms our approach by 3.3% AP_{3D}. On W → N, our approach surpasses the second-best AP_{BEV} of GroupEXP-DA by 1.3% and shares the best AP_{3D} with CMDA.

Ablation Studies

In this section, we conduct ablation studies on the Waymo → KITTI task with PointPillars (Lang et al. 2019).

Main ablation. We analyze the impact of each component of our MMAssist with the results in Table 2. “IA” represents aligning 3D features with image features, “TA” represents aligning 3D features with text features, “CAM” means incorporating pseudo labels generated from 2D images, and “STA” indicates aligning the 3D features between the student and teacher branches. (a) is the baseline, which is the basic student-teacher approach with only beam resampling (Hu, Liu, and Hu 2023). (b) incorporates new pseudo labels generated from images during self-training in

Method	Pre-Training		Self-Training				AP _{BEV}	AP _{3D}
	IA	TA	CAM	IA	TA	STA		
(a)							76.7	52.7
(b)			✓				79.1	53.2
(c)	✓		✓	✓			80.5	54.4
(d)		✓	✓		✓		80.0	54.5
(e)	✓	✓	✓	✓	✓		80.9	55.7
(f)	✓	✓	✓	✓	✓	✓	81.4	56.8

Table 2: Validation of the effectiveness of each component of MMAssist. Please refer to the text for the meanings of “IA”, “TA”, “CAM”, and “STA”.

the target domain. It can be observed that the newly added pseudo labels improve both AP_{3D} and AP_{BEV}. (c) and (d) further align the 3D features with either image features or text features. The results show that compared with (b), both “IA” and “TA” bring improvements in AP_{BEV} and AP_{3D}. (e) aligns the 3D features with both image and text features, and the performance is better than both (c) and (d). (f) is our full MMAssist, and we can see that compared with (e), “STA” can further improve the performance. These experiments effectively demonstrate the usefulness of each component. It is worth noting that compared with the vanilla PointPillars, which is (a), we add a second stage to make use of the aligned features, but simply adding a second stage does not bring about notable performance gain, the AP_{BEV} increases to 78.3% while AP_{3D} decreases to 51.9%.

Pre-Training	Self-Training	AP _{BEV}	AP _{3D}
		79.1	53.2
IA+TA		79.7	55.3
	IA+TA	79.8	54.6
IA+TA	IA+TA	81.4	56.8

Table 3: Analysis of the bridge effect of image and text features. “IA” and “TA” have the same meaning as Table 2.

Bridge effect of image and text features for 3D feature alignment. To further verify that the usefulness of image and text features mainly comes from them serving as bridges between the 3D features of the source domain and the target domain, on top of (b) in Table 2, we add “IA” and “TA” only to the pre-training stage and only to the self-training stage, respectively. The results in Table 3 show that when “IA” and “TA” are added to both the pre-training and the self-training stages, the performance gain is notably larger than only adding them to one of the two training stages.

Comparison of different feature fusion methods. To validate the effectiveness of our weighted element-wise sum approach (WSum) to fuse 3D features with aligned features, we compare our approach with element-wise sum (Sum) and concatenation (Concat) in Table 4. “Concat” concatenates the 3D features, image-aligned feature, and text-aligned feature and processes it with MLP. “Sum” directly sums the 3D features and the aligned features after changing their di-

Method	AP _{BEV} /AP _{3D}	Method		AP _{BEV} /AP _{3D}
Concat	80.5/55.4	Qwen2-VL + SLIP		80.2/55.7
Sum	78.8/50.8	LLaVA + LLaMA		80.6/52.7
WSum	81.4/56.8	LLaVA + SLIP		81.4/56.8

Table 4: Comparison of text feature generation methods. Table 5: Comparison of text feature of different feature fusion methods.

Method	AP _{BEV} / AP _{3D}		
	0-30m	30-60m	60-150m
Baseline	81.0 / 58.5	50.7 / 24.6	3.2 / 0.2
Baseline+CAM	81.4 / 58.4	51.9 / 27.9	5.0 / 0.7

Table 6: Effect of adding pseudo labels generated from images. “CAM” has the same meaning as Table 2.

mensionalities. The result shows that our approach is more effective.

Comparison of different text feature generation methods. For choice of pre-trained LLM, we test LLaVA (v1.5-13B) (Liu et al. 2023) and Qwen2-VL (Qwen2-VL-2B) (Wang et al. 2024), the results in Table 5 show that LLaVA is a better choice for MMAssist. Regarding the choice of the pre-trained text encoder, we test SLIP (Mu et al. 2022) and LLaMA (Llama-2-7B) (Touvron et al. 2023). The results reveal that SLIP is a better choice for our approach.

Effect of incorporating 3D pseudo labels generated from images at different ranges. To demonstrate the effectiveness of combining the pseudo labels generated from images and the pseudo labels generated by the teacher model, we evaluate “Baseline” and “Baseline+CAM” at different distance ranges. “Baseline” is (a) in Table 2, and “Baseline+CAM” is (b) in Table 2, which uses the pseudo labels generated from images during the self-training. The results in Table 6 show pseudo labels generated from images just have a very small impact in the range of 0-30m, but they improve the performance in the range of 30-60m by 3.3% AP_{3D} and 1.2% AP_{BEV}. The performance in the range of 60-150m can also be improved by 0.5% AP_{3D} and 1.8% AP_{BEV}. This is consistent with our pseudo label combination strategy.

Conclusion and Limitation

In this paper, we propose using multi-modal information to assist in the task of unsupervised domain adaptation on point cloud 3D object detection (3D UDA). Image and text features are used as bridges to align the 3D features between the source and target domains. The pseudo labels for self-training are enhanced with pseudo labels estimated from 2D detection results as well. Comprehensive evaluations demonstrate the effectiveness of our approach. A limitation of our work is that features are aligned at the instance level that is short of semantic information. We will investigate the usefulness of global semantic information for assisting in 3D UDA in our future work.

Acknowledgments

This work is in part supported by Natural Science Foundation of Henan Province (No. 252300421503 and 242300420270).

References

- Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11621–11631.
- Chang, G.; Roh, W.; Jang, S.; Lee, D.; Ji, D.; Oh, G.; Park, J.; Kim, J.; and Kim, S. 2024. Cnda: Cross-modal and domain adversarial adaptation for lidar-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 972–980.
- Chen, S.; Zhuo, J.; Li, X.; Liu, H.; Wang, R.; Chen, J.; and Ma, H. 2024. CMT: Co-training Mean-Teacher for Unsupervised Domain Adaptation on 3D Object Detection. In *Proceedings of the ACM International Conference on Multimedia*, 4738–4747.
- Chen, Z.; Luo, Y.; Wang, Z.; Baktashmotlagh, M.; and Huang, Z. 2023. 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*, 3714–3726.
- Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2021. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 1201–1209.
- Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.-X.; Zhao, H.; Wang, F.; Wang, N.; and Zhang, Z. 2022. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8458–8468.
- Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361.
- He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. 2017. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2961–2969.
- Hu, Q.; Liu, D.; and Hu, W. 2023. Density-insensitive unsupervised domain adaption on 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17556–17566.
- Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.
- Lavoie, M.-A.; Mahmoud, A.; and Waslander, S. L. 2025. Large Self-Supervised Models Bridge the Gap in Domain Adaptive Object Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 4692–4702.
- Li, Z.; Guo, J.; Cao, T.; Bingbing, L.; and Yang, W. 2023. Gpa-3d: Geometry-aware prototype alignment for unsupervised domain adaptive 3d object detection from point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6394–6403.
- Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual instruction tuning. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 36, 34892–34916.
- Liu, S.; Zeng, Z.; Ren, T.; Li, F.; Zhang, H.; Yang, J.; Jiang, Q.; Li, C.; Yang, J.; Su, H.; et al. 2024. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *Proceedings of the European Conference on Computer Vision*, 38–55.
- Lu, X.; and Radha, H. 2024. DALI: Domain Adaptive LiDAR Object Detection via Distribution-level and Instance-level Pseudo Label Denoising. *IEEE Transactions on Robotics*.
- Luo, Z.; Cai, Z.; Zhou, C.; Zhang, G.; Zhao, H.; Yi, S.; Lu, S.; Li, H.; Zhang, S.; and Liu, Z. 2021. Unsupervised domain adaptive 3d detection with multi-level consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8866–8875.
- Malić, D.; Fruhwirth-Reisinger, C.; Schuster, S.; and Possegger, H. 2025. GBlobs: Explicit Local Structure via Gaussian Blobs for Improved Cross-Domain LiDAR-based 3D Object Detection. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 27357–27367.
- Mu, N.; Kirillov, A.; Wagner, D.; and Xie, S. 2022. SLIP: Self-supervision meets language-image pretraining. In *Proceedings of the European Conference on Computer Vision*, 529–544.
- Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.
- Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 30.
- Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10529–10538.
- Shi, S.; Wang, X.; and Li, H. 2019. Pointcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 770–779.
- Shi, W.; and Rajkumar, R. 2020. Point-gnn: Graph neural network for 3d object detection in a point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1711–1719.
- Shin, S.; He, Y.; Vankadari, M.; Cheng, T.-Y.; Xie, Q.; Markham, A.; and Trigoni, N. 2024. Towards learning group-equivariant features for domain adaptive 3D detection. *Proceedings of the Advances in Neural Information Processing Systems*, 18019–18035.

- Sun, P.; Kretschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.
- Team, O. D. 2020. OpenPCDet: An Open-source toolbox for 3D object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>.
- Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint*, arXiv:2302.13971.
- Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; et al. 2024. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Wang, Y.; Chen, X.; You, Y.; Li, L. E.; Hariharan, B.; Campbell, M.; Weinberger, K. Q.; and Chao, W.-L. 2020. 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*, 11713–11723.
- Wei, Y.; Su, S.; Lu, J.; and Zhou, J. 2021. Fgr: Frustum-aware geometric reasoning for weakly supervised 3d vehicle detection. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 4348–4354.
- Wei, Y.; Wei, Z.; Rao, Y.; Li, J.; Zhou, J.; and Lu, J. 2022. Lidar distillation: Bridging the beam-induced domain gap for 3d object detection. In *Proceedings of the European Conference on Computer Vision*, 179–195.
- Xue, L.; Yu, N.; Zhang, S.; Panagopoulou, A.; Li, J.; Martín-Martín, R.; Wu, J.; Xiong, C.; Xu, R.; Niebles, J. C.; et al. 2024. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 27091–27101.
- Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.
- Yang, J.; Shi, S.; Wang, Z.; Li, H.; and Qi, X. 2021. St3d: Self-training for unsupervised domain adaptation on 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10368–10378.
- Yang, J.; Shi, S.; Wang, Z.; Li, H.; and Qi, X. 2022. St3d++: Denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(5): 6354–6371.
- Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11040–11048.
- Yihan, Z.; Wang, C.; Wang, Y.; Xu, H.; Ye, C.; Yang, Z.; and Ma, C. 2021. Learning transferable features for point cloud detection via 3d contrastive co-training. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 34, 21493–21504.
- Zhang, G.; Chen, J.; Gao, G.; Li, J.; Liu, S.; and Hu, X. 2024a. Safdnet: A simple and effective network for fully sparse 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14477–14486.
- Zhang, R.; Zhang, H.; Yu, H.; and Zheng, Z. 2024b. Approaching outside: scaling unsupervised 3D object detection from 2D scene. In *Proceedings of the European Conference on Computer Vision*, 249–266.
- Zhang, Z.; Chen, M.; Xiao, S.; Peng, L.; Li, H.; Lin, B.; Li, P.; Wang, W.; Wu, B.; and Cai, D. 2024c. Pseudo label refinery for unsupervised domain adaptation on cross-dataset 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15291–15300.
- Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4490–4499.