Weakly Supervised Monocular 3D Object Detection using Multi-View Projection and Direction Consistency

Runzhou Tao[1,3*], Wencheng Han[2*], Zhongying Qiu[3], Cheng-zhong Xu[2], Jianbing Shen[2†]
[1] Beijing Institute of Technology, [2] SKL-IOTSC, CIS, University of Macau, [3] QCraft
{wencheng256, shenjiangbingcg}@gmail.com
https://github.com/weakmono3d/weakmono3d

Abstract

Monocular 3D object detection has become a mainstream approach in automatic driving for its easy application. A prominent advantage is that it does not need LiDAR point clouds during the inference. However, most current methods still rely on 3D point cloud data for labeling the ground truths used in the training phase. This inconsistency between the training and inference makes it hard to utilize the large-scale feedback data and increases the data collection expenses. To bridge this gap, we propose a new weakly supervised monocular 3D objection detection method, which can train the model with only 2D labels marked on images. To be specific, we explore three types of consistency in this task, i.e. *the projection, multi-view and direction consistency, and design a weakly-supervised architecture based on these consistencies. Moreover, we propose a new 2D direction labeling method in this task to guide the model for accurate rotation direction prediction. Experiments show that our weakly-supervised method achieves comparable performance with some fully supervised methods. When used as a pre-training method, our model can significantly outperform the corresponding fully-supervised baseline with only 1/3 3D labels.*

1. Introduction

Monocular 3D object detection is a foundational research area in computer vision and plays an important role in autonomous driving systems. It aims to identify the objects and estimate the 3D bounding boxes of the corresponding targets with a single image as input. Different from 3D point clouds detection methods like [42, 43, 47, 53, 61], monocular 3D detection models [28, 34, 48, 49, 65] alleviate the need of LiDAR sensors, making the self-driving system easier to be applied.

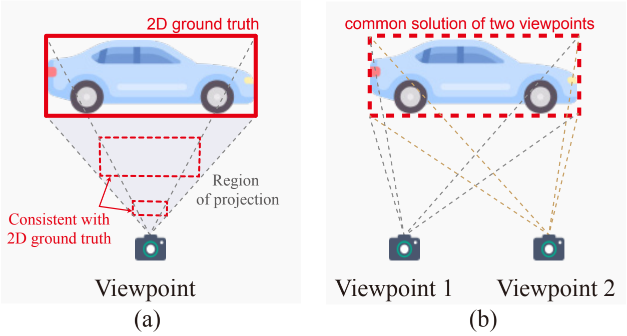


Figure 1. **Illustration of the projection and multi-view consistency.** (a) Only projection consistency cannot determine the accurate position of the target because projection loss has more than one optimal solution in the 3D space. For example, the two dashed boxes in 3D space produce the same projection loss because they have the same projection in 2D space. (b) Constrained by the multi-view consistency, the optimal solution must be the common solution for two viewpoints, that is, the target location.

However, there is still a challenging problem that limits the application of 3D object detection with pure camera vision data. That is, the ground truth 3D boxes used in the training phases are usually labeled with 3D point clouds [2, 10, 45]. Recently, self-driving systems with pure camera vision inputs have become a new trend. But the feedback video clips captured by the production cars cannot be utilized to improve the 3D object detection models because of the lack of training labels. Compared with the data from the data-collection cars, the feedback images from production cars have a larger scale diversity and contain more corner cases, which are crucial for improving the robustness of models.

In this paper, we propose a new weakly supervised training method, which can train the 3D object detection models with only camera images and 2D labels, making it possible to utilize the feedback data from the production cars. To

*Equal contribution. †Corresponding author: *Jianbing Shen*. This work was supported in part by the FDCT grant SKL-IOTSC(UM)-2021-2023, the FDCT Grant 0123/2022/AFJ, the Grant MYRG-CRG2022-00013-IOTSC-ICI, and the Grant SRG2022-00023-IOTSC.

achieve this goal, we exploit three types of consistency between the 3D boxes and 2D images and fully utilize them for training the object detection models. The first is the projection consistency. With the intrinsic matrix of a camera, a 3D box predicted by the models can be projected into the 2D image space, and the projected boxes should be consistent with the corresponding 2D boxes. Based on this, we propose a projection loss by minimizing the difference between the projected boxes and 2D ground truths. This criterion can guide the predicted 3D boxes into the projection regions. However, only projection consistency cannot provide enough information to correct the errors in the 3D space, especially for the depth dimension, as shown in Fig. 1 (a). According to the perspective principle, multiple boxes in the 3D space can be projected into the same 2D box in the image. Thus, there is more than one optimal solution for the projection loss, and the errors caused by these boxes cannot be optimized by the projection loss.

Aiming at solving this limitation, we incorporate multi-view consistency into our method to minimize the errors in 3D space. The same object captured from different viewpoints would show different positions and shapes in the corresponding 2D images. But in the 3D space, 3D bounding boxes belonging to the same object should be consistent, *i.e*. they should be of the same position, size and rotation angle in a certain coordinate system. Based on this, we construct the multi-view consistency by minimizing the discrepancy between the predicted bounding boxes of the same object from a different point of view. As shown in Fig. 1(b), projection losses on the two viewpoints will constrain the predictions into their projection regions, and the multi-view consistency will further guide the predictions to the common optimal solutions of the two views, which are where the objects located. Notably, in our work, images paired from different viewpoints are only used for calculating the losses, and the models still only take monocular inputs in the evaluation phase.

The last consistency presented in this paper is the direction consistency for guiding the prediction of the direction scale. In previous works [2] [45] [10], 3D rotation direction is labeled on point clouds by a vector from the center to the front of objects. To avoid the need for 3D LiDAR data, we propose a new labeling method named 2D direction label directly on pure camera images, indicating the 2D direction of the object in the images. The predicted 3D box rotation should be consistent with the direction in 2D space when they are projected, *i.e*. the direction consistency. Based on this consistency, we further design a 2D rotation loss for optimizing the rotation-scale estimation.

The proposed weakly supervision method is a general framework that can be integrated with most monocular 3D detection models. To show the efficiency of our method, we incorporate it with a representative model - DD3D [34] in this task, and evaluate it on the KITTI benchmark. Results show that our method can achieve comparable performance with some fully supervised methods. Also, to demonstrate the application in real scenes, we collect a new dataset named **ProdCars** from production cars and evaluate the performance of our method on it.

In short, in this work, we propose a novel weakly supervised method for monocular 3D object detection, which only utilizes 2D labels as ground truth without depending on 3D point clouds for labeling, making us the first to do so. Our approach incorporates projection consistency and multi-view consistency, which are used to design two consistency losses guiding the prediction of accurate 3D bounding boxes. Additionally, we introduce a new labeling method called 2D direction label, replacing the 3D rotation label in point clouds data and a direction consistency loss based on the new labels. Our experiments show that our proposed weakly supervised method achieves comparable performance with some fully supervised methods, and even with only 1/3 of the ground truth labels, our method outperforms corresponding fully supervised baselines, demonstrating its potential for improving models based on feedback production data.

2. Related Works

2.1. Monocular 3D Object Detection

Monocular 3D object detection is a challenging task because of the lack of object information in 3D space. Previously, lots of monocular 3D object detection methods [5, 24, 26, 33, 38, 50] aim to find the clues benefited for 3D object detection from 2D images. Deep3DBox [33] detected 2D object MS-CNN [3] by 2D object detector and estimated the dimension and orientation of the object by a 3D regression head. Afterwards, the network combines geometric projection constraints to obtain a 3D pose. FCOS3D [50] is an improved 3D objetc detection method based on FCOS [46], which decoupled the 7-DoF 3D object into 2D and 3D attributes, and assigns an object to different feature levels by the 2D dimension of the object. DD3D [34] noticed that the depth information is highly critical to the monocular 3D object detection task. It learned the dense depth estimation in advance, and significantly improved the 3D object detection performance. According to the geometric relationship of 3D objects, MonoDDE [21] generated multiple estimations for each object, and formulated the depth selection with combination strategies to make the final depth more accurate.

Recently, more well-known algorithms [8, 18, 19] have been proposed to explore the useful information in 2D space to further constrain the 3D detection results. Based on the original 3D regression head, MonoCon [25] added some additional auxiliary tasks to predict the object center point and corner point in the 2D image to better learn the correction information. MonoJSG [22] formulated the depth estimation as a progressive refinement procedure and proposed a joint semantic and geometric cost volume to evaluate the depth error. It obtained the depth of the object more accurately by segmenting the object and estimating the depth of the dense points in the object. In addition, to directly feed monocular images into the network, [51,58] used the depth estimation network [4,9] to predict the depth value of each pixel for the monocular image, and represented the image in the form of the 3D point cloud. Thus, the 3D object detector can be used to detect it; however, due to the disadvantage of high delay, it is hard to meet the real-time requirement of automatic driving scenes.

2.2. LiDAR 3D Object Detection

Compared with images, LiDAR point clouds have disorder characteristics, rotation invariance and uneven distribution, which leads to the 2D convolutional network cannot be directly applied for 3D point cloud data. Therefore, it is particularly important to extract information from the 3D point cloud data directly [20,32,52,54–57]. PointNet [36] directly extracted high-level features from the coordinate of the 3D point clouds through MLP, and aggregated the global features by max-pooling layers to solve the disorder problem of the point clouds data. Based on PointNet [36], PointNet++ [37] recursively aggregated the 3D point clouds in a layered manner and proposed a multi-scale and multi-resolution grouping scheme. Unlike the methods of directly using 3D point clouds, VoxelNet [64] divided the 3D point clouds into equidistant 3D voxels and encoded each voxel to output the 3D object detection result through the region proposal network. MV3D [7] projected the 3D point clouds into a 2D bird's-eye grid, and set a large number of 3D anchor boxes in advance to generate the 3D bounding box. PointPillar [16] encoded the 3D point clouds into a vertical column as a special voxel. Then, the 3D point clouds in the cylinder are encoded through PointNet [36] to generate a pseudo-2D image and finally use the 2D object detector to detect 3D objects.

2.3. Weakly Supervised 3D Object Detection

For 3D object detection task, data annotations for 3D point clouds are very expensive, and so some works explore how to use more straightforward annotations to train 3D object detection task. WS3D [31] presented a weakly supervised method for 3D LiDAR object detection with two stages. In the first stage, the cylindrical object proposal is generated by clicking annotations in the bird's-eye-view. Afterwards, the network uses a few precisely labeled object instances to optimize the cylindrical proposal to generate the final 3D object bounding box. VS3D [39] learned from the point cloud to generate the 3D proposal and uses a 2D classification network to identify the 3D proposal. WeakM3D [35] first detected the image and combined the 3D point clouds with the detection results to obtain the object-LiDAR-points. In addition, it proposed a method to estimate object orientation θ, which obtained the orientation of each pair of points in the object-LiDAR-points. On the contrary, our method not only does not rely on the 3D point clouds data, but also achieve comparable results with some fully-supervised methods.

3. The Proposed Method

3.1. Architecture

Fig. 2 shows an overview of the proposed architecture, in which the proposed method is general and can be incorporated with most previous monocular 3D detection methods. To show the efficiency, we employ the representative DD3D [34] as our base detection model, keeping all the settings the same as our baseline except for the training method. In the training phase, image pairs taken from two different viewpoints and their relative positions are needed. According to the available data, image pairs can be retrieved from different sources. For data from multiple cameras, the pairs could be the images from different cameras with some overlap areas. For data captured by only a single camera, the pairs could be adjacent frames from video sequences when the camera is moving, and the relative positions could be predicted by a separated network like [62]. Firstly, the image pairs $(\boldsymbol{I_1},\ \boldsymbol{I_2})$ are sent into the backbone model $\mathcal{F}$ for feature extraction and produce the corresponding feature maps $\boldsymbol{f_1},\ \boldsymbol{f_2}$:

$$\boldsymbol{f_1} = \mathcal{F}(\boldsymbol{I_1}, W); \boldsymbol{f_2} = \mathcal{F}(\boldsymbol{I_2}, W), \tag{1}$$

where W is the weight of the backbone model. Then, different heads are employed to generate predictions for the objects from the feature maps $\boldsymbol{f_1},\ \boldsymbol{f_2}$:

$$cls_i = \mathcal{H}_{cls}(\boldsymbol{f_i}); B_i = \mathcal{H}_{3d}(\boldsymbol{f_i}), i \in \{1, 2\}, \tag{2}$$

where $\mathcal{H}_{cls}$ is the classification head and $\mathcal{H}_{3d}$ is the 3D detection heads. B is the predicted 3D boxes for the targets.

The predictions and 2D labels are used to calculate the losses. The projection loss L_{proj} and direction consistency loss L_{rot} are applied on the results of each image, and the multi-view consistency loss L_{con} is calculated between the results of the pairs:

$$\begin{aligned} \mathcal{L}_{proj} &= E_{proj}(B_1, y_{box1}) + E_{proj}(B_2, y_{box2}); \\ \mathcal{L}_{rot} &= E_{rot}(B_1, y_{dir1}) + E_{rot}(B_2, y_{dir2}); \\ \mathcal{L}_{con} &= E_{con}(B_1, B_2) \end{aligned} \tag{3}$$

where y_{box} denotes the 2D boxes labels on the image and y_{dir} represents the 2D direction labels introduced in § 3.4.

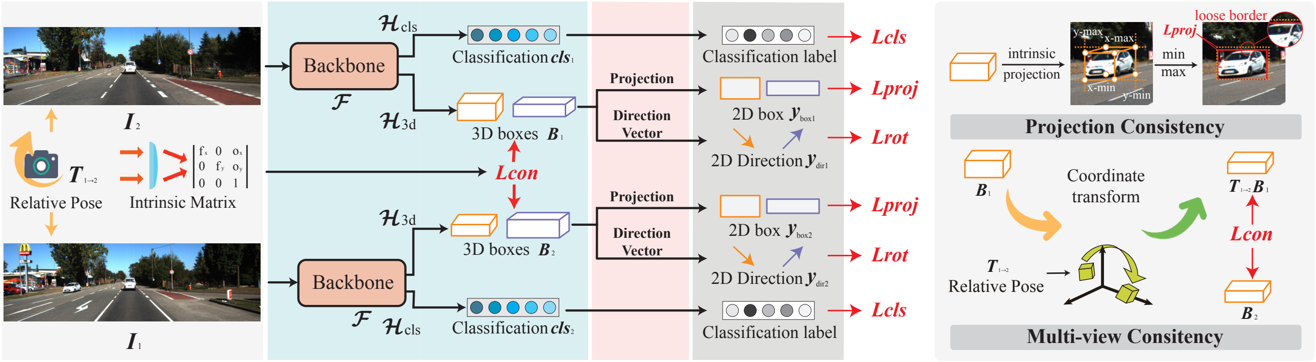


Figure 2. **The architecture of the proposed method.** The left column shows that during the training phase, image pairs from different viewpoints are sent into the detection model, and 4 losses are computed between the predictions and the 2D ground truth. The right column shows the details of the projection consistency and the multi-view consistency. To calculate the projection consistency loss, we project the predicted box into the 2D image and convert it into a 2D box and finally compute the differences between the 2D box and 2D box labels. To calculate the consistency loss, we first transform the predicted 3D boxes from viewpoint 1 into the coordinate system of viewpoint 2 and then compute the difference between the converted boxes and the predicted boxes of viewpoint 2.

E are the criterion of corresponding losses. Finally, the three losses and classification loss $\mathcal{L}_{cls}$ are summed together to be the final loss $\boldsymbol{\mathcal{L}}$:

$$\boldsymbol{\mathcal{L}} = \mathcal{L}_{proj} + \mathcal{L}_{con} + \mathcal{L}_{rot} + \mathcal{L}_{cls}. \tag{4}$$

During the inference, the model only takes one image from a single viewpoint and produces the 3D boxes for the targets. In the rest of this section, we will introduce three consistencies between the 2D labels and 3D predictions and how we use them for guiding the optimization of models.

3.2. Projection Consistency

To avoid the need of the 3D point clouds annotations in the labeling procedure, we only employ 2D ground truths of images in our work. An intuitive idea is that the predicted 3D boxes can be projected into the 2D image space, and the projected 2D boxes should be consistent with the labeled 2D boxes. We define this property as the projection consistency.

Given the intrinsic matrix of a camera, a point $\boldsymbol{Q}(x, y, z)$ in the real world 3D space can be converted into a point $\boldsymbol{q}$ in the 2D image space by multiplying with the matrix:

$$z\boldsymbol{q} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{Q} \tag{5}$$

where f_x and f_y are the pixel focal lengths and o_x, o_y are the offsets of the principal point. Therefore, the 3D boxes B with 8 corner points can be transformed into the corresponding points in the 2D space. Then a 2D bounding box b can be generated by calculating the minimum and maximum values of the x,y coordinates, as shown in Fig. 2. Finally, the loss values are computed between the b and the 2D ground truth y_{box}.

When estimating the error, there are some significant differences between our method and the traditional 2D box loss. The first is that there exists a gap between the 2D loss and the 3D evaluation metrics. We only compute the 2D box difference in the training phase, but the 3D box difference is the target we want to minimize in the evaluation. The 3D box of an object is usually translation invariant, *i.e.* no matter whether the object is near to the camera or far from it, the 3D box is still the same. But the 2D box will change according to the perspective principle. This gap would introduce some undesired problems. For example, objects far from the camera have small bounding boxes in the 2D space, but their evaluation is conducted on the original size. That means even a small difference between the 2D boxes would cause large errors in the 3D space. To balance the performance of near and far objects, we employ a size-independent $GIoU$ loss [40] as our main criterion. As the discrepancy of near boxes is enlarged by the perspective, we employ an extra L_1 loss to improve the detection performance on near targets.

Secondly, since the targets are not exact cubes, the projected boxes cannot perfectly align with the 2D boxes. As shown in Fig. 2, the converted boxes will introduce the loose boundaries on the images, while the labeled 2D boxes do not. Considering this, we replace the traditional L_1 by SmoothL1 [11] in our loss function, and leave a soft margin when computing the difference between the converted boxes and 2D ground truth. If the difference between the boxes is smaller than a given threshold, the gradient will be reduced as:

$$SmoothL_1 = \begin{cases} |a-b| - 0.5 \times \gamma, & if\ |a-b| > \gamma \\ 0.5(a-b)^2/\gamma, & otherwise \end{cases}, \tag{6}$$

where a, b are the input boxes and γ is the soft margin. Finally, the projection loss is defined as:

$$E_{proj} = GIoU(b, y_{box}) + \lambda SmoothL_1(b, y_{box}, \gamma), \quad (7)$$

where λ is the balance ratio.

3.3. Multi-View consistency

The projection consistency would constrain the predicted 3D boxes into the corresponding projection region, where all the 3D boxes have the same 2D projection box, as shown in Fig. 1 (a). This means that there is more than one optimal model solution with only the projection loss. However, the real and exact position of the target is the only solution we want to get, so we need to find more constrains to achieve this goal.

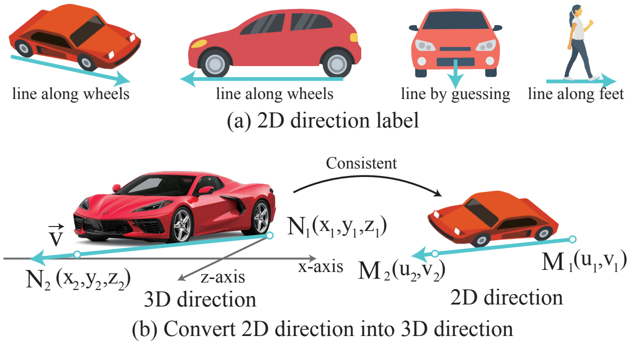
If we observe an object from two different viewpoints at the same time, an object will show different projection views in the two images, but its 3D bounding box in the real world is the same. Therefore, there exists an inner consistency between the observations from different viewpoints, and we call this multi-view consistency. As shown in Fig. 1(b), two different viewpoints have two independent projection regions, each of which has a set of optimal solutions constrained by the projection loss. But when fixing the rotation angle of objects, there is only one common solution of the two sets, which is the target box. Besides the position, all the 3D attributes of one target would be the same when observed from a different point of view, such as the size and the direction.

To utilize this consistency, we should first convert the 3D boxes from different viewpoints into the same coordinate system. So we need a relative position matrix $\boldsymbol{T}_{1\to 2}$ between the viewpoints. There are several ways to get this information. For data captured by multiple cameras, the relative position is fixed after the cameras are installed. For video sequences captured by a single camera, the relative position can be predicted by a PoseNet [62] trained by the self-supervised methods.

Given the 3D boxes B_1 and B_2 from two different viewpoints, we first convert B_1 from its coordinate system into that of B_2. Then we calculate the difference between the converted box and B_2. In this module, we employ an L_1 loss as the criterion:

$$E_{con} = L_1(B_2, \boldsymbol{T}_{1\to 2} B_1) \quad (8)$$

There is stationary a problem when the viewpoint data is extracted from video sequences. That is, some of the objects are not still when capturing the two frames, which would cause inconsistency between the two viewpoints. To solve this problem, we employ a simple but efficient strategy by labeling the objects as inconsistent if they move obviously in the frame, without computing the corresponding multi-view losses produced by them.



(a) 2D direction label

(b) Convert 2D direction into 3D direction

Figure 3. **Illustration of the Direction Consistency.** (a) Examples of how the 2D directions are labeled. For most circumstances, we can mark the 2D direction by drawing a parallel line of the wheels or feet. (b) Given a 2D direction, we can convert it into the corresponding 3D direction according to consistency.

3.4. Direction Consistency

In most previous approaches [2, 10, 45], the rotation angels are labeled with 3D point clouds. Without 3D information, it is quite difficult to indicate accurate rotation angles for humans. In this paper, instead of labeling the rotation angles, we propose a new method to get the rotation according to the direction consistency.

If we draw a vector in the 3D space to indicate the direction of the object, we will get a 2D vector in the image when projecting it into the 2D space. We call 2D vector as 2D direction, and the relationship between it and 3D rotation is called the direction consistency. The 2D directions are easier to be labeled on the images by humans. Fig. 3(a) shows some representative examples of the 2D directions. For most of the objects, the 2D direction can be labeled by drawing a line along the wheels on one side of the car or the feet of people. Even if the wheels are occluded, we can also guess the direction according to the surrounding environment, such as the lane lines.

Then, we design a new method to recover 3D rotations from the 2D labels. Given a 2D direction vector $\overrightarrow{M_1M_2}$, $M_1 = (u_1, v_1)$ and $M_2 = (u_2, v_2)$ are the start and end point of the line. There are two corresponding points in the 3D spaces named $N_1 = (x_1, y_1, z_1)$ and $N_2 = (x_2, y_2, z_2)$, and the 3D direction vector is $\overrightarrow{N_1N_2}$, as shown in Fig. 3. With the intrinsic matrix, the relationship between M_1, M_2 and N_1, N_2 is defined as:

$$\begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_i/z_i \\ y_i/z_i \\ 1 \end{bmatrix} \quad (9)$$

Then, it can be simplified as:

$$u_i = \frac{x_i f_x}{z_i} + o_x; v_i = \frac{y_i f_y}{z_i} + o_y, i \in \{1, 2\}. \quad (10)$$

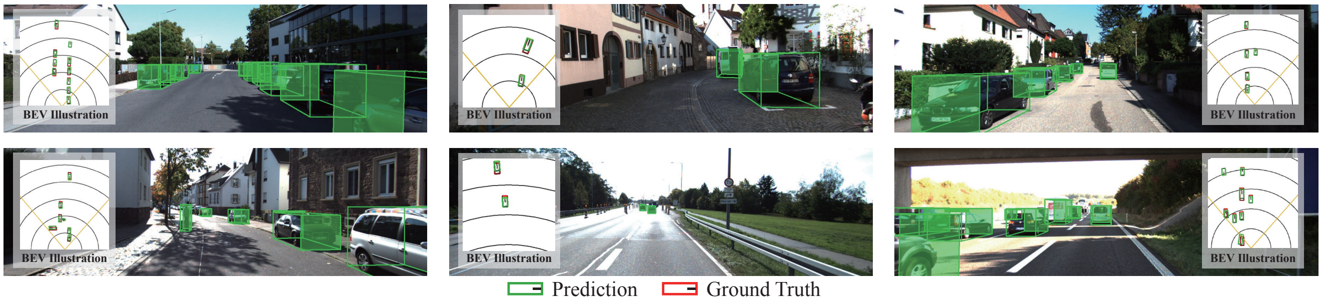


Figure 4. **Visualizations of our WeakMono3D results.** Both the solid side in the image and the short bar in the BEV-box indicate the direction of the object.

We can further represent x_i and z_i using u_i, v_i and y_i as:

$$x_i = \frac{y_i f_y (u_i - o_x)}{f_x (v_i - o_y)}; z_i = \frac{y_i f_y}{v_i - o_y}, i \in \{1, 2\}. \tag{11}$$

For most cases, the targets are in the same plane as the camera. So we mainly focus on the rotation in the XZ-plane and presume the direction vector in 3D space is parallel to the ground. Therefore, we assume that all the y will have the same values in Equation (11). Then, the direction vector in the 3D space can be calculated as:

$$\begin{aligned} \vec{n} &= \boldsymbol{N}_2 - \boldsymbol{N}_1 \\ &= (x_2 - x_1, z_2 - z_1). \end{aligned} \tag{12}$$

Finally, the rotation loss can be calculated between the recovered vector and the predicted rotation vector:

$$E_{rot} = 1 - \frac{\vec{n} \cdot \vec{n}_p}{|\vec{n}||\vec{n}_p|} \tag{13}$$

where $\vec{n}_p$ is the predicted rotation vector. In Equation (13), y is eliminated. Thus, the value of y will not influence the result, and the loss value is determined by u and v values.

4. Experiment

4.1. Experiment Setup

Implementation Details. Our experiments are based on DD3D [34], which employs a DLA-34 [59] and FPN [23] as the detector's backbone and neck. We maintain the structure of the original network and keep all hyper-parameters the same as our baseline. The learning rate is set as 2×10^{-3}, dropped by multiplying 0.1 both at 170,000 and 190,000 iterations. We adopt the SGD optimizer for training. During inference, we save the top 100 detected 3D bounding boxes and use NMS to remove the redundant predictions.

Data Augmentation. We adopt three data augmentation techniques: random horizontal flip, random resize, and random color jitter. In our implementation, we set a 50% chance to flip the image horizontally, and according to the preset sizes, the picture is randomly resized with an equal ratio. Color jitter is carried out on three types of transforms: brightness, contrast, and saturation, and the proportion of jitter is from -0.2 to 0.2.

4.2. Comparison on KITTI

KITTI [10] is a popular 3D object detection benchmark that contains 7,481 images for training and 7,518 for testing. We follow the setting in [6] and split the original training set into 3,712 images for training and 3,769 images for validation. In this benchmark, there are three classes: Car, Pedestrian, and Cyclist. For each class, there are three levels of difficulty: Easy, Moderate, and Hard. Most weakly supervised methods in the 3D object detection area employ IOU 0.5 criterion on the valid set and IOU 0.7 criterion on the test set. In our experiments, we follow the same criterion as previous works. We evaluate our method on the validation set locally and submit the detection results to the official website to obtain the metric of the test set.

To show the improvements, we compare our WeakMono3D with some representative monocular 3D object detection methods, including fully and weakly supervised ones. Table 1 shows our comparison on the validation set. We achieved AP$_{3D}$ of $49.37, 39.01, 36.34$ and AP$_{BEV}$ of $54.32, 42.83, 40.07$, in terms of the Easy, Moderate and Hard categories. Compared with some weakly supervised methods, results show that even without needing LiDAR data, our method can achieve comparable performance with previous methods and significantly outperforms them in the Moderate and Hard categories. We attribute that 3D point clouds in the distance are sparse, leading to the previous models' performance degeneration. Images do not have this problem where all the objects have a dense representation, so our method performs well in the categories mainly containing far objects. Moreover, our method performs comparably with some fully supervised methods without 3D annotation. In Table 2, we compare the metrics on the test set for cars with the prior methods. This comparison is more convincing than the validation set because the ground truth labels of the test set are not publicly available. Our method obviously outperforms WeakM3D [35] on all the metrics, and performs comparably to some fully-supervised meth-

Method	Supervision	LiDAR	AP_{BEV}/AP_{3D} (IoU=0.5)$\|_{R_{40}}$		
			Easy	Moderate	Hard
CenterNet [63]	Full	Label	34.36/20.00	27.91/17.50	24.65/15.57
MonoGRNet [38]			52.13/47.59	35.99/32.28	28.72/25.50
M3D-RPN [1]			53.35/48.53	39.60/35.94	31.76/28.59
MonoPair [8]			61.06/55.38	47.63/42.39	41.92/37.99
MonoDLE [29]			60.73/55.41	46.87/43.42	41.89/37.81
GUPNet [27]			61.78/57.62	47.06/42.33	40.88/37.59
VS3D [39]	Weak	Train+Val	31.59/22.62	20.59/14.43	16.28/10.91
Autolabels [60]		Train	50.51/38.31	30.97/19.90	23.72/14.83
WeakM3D [35]		Train	58.20/50.16	38.02/29.94	30.17/23.11
Our WeakMono3D		None	54.32/49.37	42.83/39.01	40.07/36.34

Table 1. **Comparison on KITTI validation set for the car category.** For all results, we use AP$|_{R_{40}}$ metrics with IoU threshold= 0.5.LiDAR means where the method uses LiDAR data. We highlight our results in green .

Method	Supervision	LiDAR	AP_{BEV}/AP_{3D} (IoU=0.7)$\|_{R_{40}}$		
			Easy	Moderate	Hard
FQNet [24]	Full	Label	5.40/2.77	3.23/1.51	2.46/1.01
GS3D [17]			8.41/4.47	6.08/2.90	4.94/2.47
ROI-10D [30]			9.78/4.32	4.91/2.02	3.74/1.46
MonoGRNet [38]			18.19/9.61	11.17/5.74	8.73/4.25
MonoPSR [15]			**18.33/10.76**	**12.58/7.25**	**9.91/5.85**
3D-GCK [12]			5.79/3.27	4.57/2.52	3.64/2.11
WeakM3D [35]	Weak	Train	11.82/5.03	5.66/2.26	4.08/1.63
Our WeakMono3D		None	12.31/6.98	8.80/4.85	7.81/4.45
Improvement	-	-	+0.49/+1.95	+3.14/+2.59	+3.73/+2.82

Table 2. **Comparison on KITTI test set for the car category.** For all results, we use AP$|_{R_{40}}$ metrics with IoU threshold= 0.7. The best results are highlighted in **bold** and our results are with green background. ”Imporvement” means the improvement of our method compared with WeakM3D.

ods. Table 3 shows our performance for pedestrian and cyclist class on the KITTI test set. Unlike some weakly supervised works [14, 35] mainly focus on rigid objects like Cars, our method has better generality in pedestrians and cyclists, showing comparable performance with fully-supervised methods.

Quantitive Results Fig. 4 shows more visualizations results by our method. Owe to guidance from the projection loss, the visualized 3D boxes in the 2D images fit well with the corresponding targets. ccording to the BEV illustration, the predictions have achieved accurate depth values and rotation angles, since the consistency loss provides a comprehensive 3D spatial guidance and the rotation loss corrects the direction difference between 2D ground truth and predictions.

4.3. ProdCars Dataset

To prove that our method can work well with the production data, we collect a new dataset from production cars and name it as **ProdCars**. There are about 10,000 image frames in the training set, each containing 2D boxes and 2D direction labels without point clouds. We also collect a new validation set to evaluate the performance. The validation set is collected by another car with the same cameras as production cars with one LiDAR sensor. Therefore, these images can be labeled with 3D LiDAR ground truths for evaluation purposes. Table 4 show the results of our WeakMono3D on ProdCars.

4.4. Ablation Study

To verify the significance of the proposed method, we conduct several ablation studies on KITTI validation dataset. We mainly focus on three aspects. The first one is the three weakly supervised losses proposed in this paper. Then the second one is the integration with the fully supervised method. And finally, we discuss the influence of different types of multi-view data.

Losses. In this paper, we discuss three types of consistency and propose three corresponding loss functions, L_{proj}, L_{con} and L_{rot}. Table 5 shows their ablation studies of them. The model with only projection loss shows a totally zero result because the projection loss itself cannot correct the prediction error in the 3D space. Therefore the final predictions have accurate projections in the image but wrong positions in the real world, producing zero metrics. Direction consistency loss is also critical to the performance

Method	Pedestrian			Cyclist		
	Easy	Moderate	Hard	Easy	Moderate	Hard
OFTNet [41]	1.28/0.63	0.81/0.36	0.51/0.35	0.36/0.36	0.16/0.16	0.15/0.15
SSD3D [13]	2.48/2.31	2.09/1.78	1.61/1.48	3.45/-	1.89/-	1.44/-
M3D-RPN [1]	5.65/4.92	4.05/3.48	3.29/2.94	1.25/0.94	0.81/0.65	0.78/0.47
MonoPSR [15]	7.24/6.12	4.56/4.00	4.11/3.30	**9.87/8.47**	**5.78/4.74**	**4.57/3.68**
MonoDis [44]	9.07/7.79	5.81/5.14	5.09/4.42	1.47/1.17	0.85/0.54	0.61/0.48
DD3D [34]	**15.90/13.91**	**10.85/9.30**	**8.05/8.05**	3.20/2.39	1.99/1.52	1.79/1.31
Ours	5.53/4.62	3.86/3.13	3.29/2.87	3.04/2.25	1.87/1.41	1.59/1.22

Table 3. **Comparison on KITTI test set for Pedestrian and Cyclist category.** For all results, we use AP$|_{R_{40}}$ metrics with IoU threshold= 0.7. We highlight the best results in **bold** and our results in green.

AP$_{BEV}$/AP$_{3D}$ (IoU=0.5)$\vert_{R_{40}}$		AP$_{BEV}$/AP$_{3D}$ (IoU=0.7)$\vert_{R_{40}}$	
Easy	Hard	Easy	Hard
34.85/32.80	24.03/22.07	17.58/14.30	10.37/7.32

Table 4. **Results on the proposed ProdCars dataset.**

#	L_{proj}	L_{con}	L_{rot}	AP$_{BEV}$/AP$_{3D}$ (IoU=0.5)$\vert_{R_{40}}$		
				Easy	Moderate	Hard
0	✓			0.00/0.00	0.00/0.00	0.00/0.00
1	✓	✓		17.36/11.78	15.27/9.58	14.23/9.13
2	✓	✓	✓	**54.32/49.37**	**42.83/39.01**	**40.07/36.34**

Table 5. **Ablation study for different losses.**

#	Source	AP$_{BEV}$/AP$_{3D}$ (IoU=0.5)$\vert_{R_{40}}$		
		Easy	Moderate	Hard
0	Multi-Camera	**54.32/49.37**	**42.83/39.01**	**40.07/36.34**
1	Sequence	43.04/36.68	31.57/26.80	27.18/23.33

Table 6. **Ablation study for the multi-view data sources.**

#	Ratio	AP$_{BEV}$/AP$_{3D}$ (IoU=0.5)$\vert_{R_{40}}$		
		Easy	Moderate	Hard
DD3D	1	64.09/58.08	51.84/46.78	47.52/42.70
0	0	54.32/49.37	42.83/39.01	40.07/36.34
1	1/20	59.47/53.61	48.11/43.72	43.27/39.45
2	1/10	63.72/57.21	49.95/44.60	44.54/40.50
3	1/5	63.92/59.12	50.52/46.46	46.29/41.66
4	1/3	**66.36/61.02**	**53.00/47.69**	**48.48/43.56**

Table 7. **Ablation study of integration with the fully supervised method.**

of models. The optimization of size prediction relies on minimizing the projection error. If the rotation of objects is not accurate, this loss will guide the model to provide the wrong sizes to fit the 2D ground truth, leading to a bad performance. In short, the three losses are all indispensable in our weakly supervised method. Only by employing them together can we give comprehensive guidance for the optimization of the detection models.

Employed as Pre-training Method. One of the main purposes of this paper is to enhance the detection model by exploiting the feedback data. To show the feasibility of this, we employ our method as a pre-training manner and fine-tune the model with a small proportion of fully labeled data. As shown in Table 7, even fine-tuned with an extremely small amount of data, our model can achieve comparable performance with the corresponding fully supervised model. With about 1/3 of fully labeled data, our method can significantly outperform the fully supervised baseline DD3D [34].

Different Types of Multi-view Data. As we discussed in § 3.3, multi-view images can be taken from different sources of data. Table 6 shows the comparison of two sources, the multi-camera data, or the video sequences from one single camera. We employ the left and right cameras in the KITTI dataset as multi-camera data, and frames from the left camera as the video sequences in our experiments. Experiments show that multi-camera data can achieve better performances than sequence data. In other experiments on KITTI dataset, we only report the results of the multi-camera trained model.

5. Conclusion

In this paper, we have proposed a weakly supervised method that can train the monocular 3D object detection models with only 2D labels marked on images. To achieve this, we explore three types of consistency, *i.e.* projection, multi-view, and direction consistency, and design three losses based on them. To show the feasibility, we propose a new dataset **ProdCars** collected by the production cars, and experiments show that our method can work well on this data. When employed as a pre-training method, our model can significantly outperform the corresponding fully supervised version with only 1/3 of labeled data. Experiments show that our trained models can achieve comparable performance with some fully supervised methods. We believe the proposed method can help to use the rich data from the production cars, because these data contain a larger variety of scenes than experimentally collected data, which is critical to improve the robustness and generality of models.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 9287–9296, 2019. 7, 8

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 1, 2, 5

[3] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 2

[4] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5410–5418, 2018. 3

[5] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 2

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew G Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals for accurate object class detection. *Advances in neural information processing systems*, 28, 2015. 6

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 3

[8] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020. 2, 7

[9] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 3

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 2, 5, 6

[11] Ross Girshick. Fast r-cnn. In *IEEE/CVF International Conference on Computer Vision*, pages 1440–1448, 2015. 4

[12] Nils Gählert, Jun-Jun Wan, Nicolas Jourdan, Jan Finkbeiner, Uwe Franke, and Joachim Denzler. Single-shot 3d detection of vehicles from monocular rgb images via geometrically constrained keypoints in real-time. In *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020. 7

[13] Eskil Jörgensen, Christopher Zach, and Fredrik Kahl. Monocular 3d object detection and box fitting trained end-to-end using intersection-over-union loss. *arXiv preprint arXiv:1906.08070*, 2019. 8

[14] Lukas Koestler, Nan Yang, Rui Wang, and Daniel Cremers. Learning monocular 3d vehicle detection without 3d bounding box labels. In *DAGM German Conference on Pattern Recognition*, pages 116–129. Springer, 2020. 7

[15] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11867–11876, 2019. 7, 8

[16] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 3

[17] Buyu Li, Wanli Ouyang, Lu Sheng, Xingyu Zeng, and Xiaogang Wang. Gs3d: An efficient 3d object detection framework for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1019–1028, 2019. 7

[18] Peixuan Li and Huaici Zhao. Monocular 3d detection with geometric constraint embedding and semi-supervised training. *IEEE Robotics and Automation Letters*, 6(3):5565–5572, 2021. 2

[19] Peixuan Li, Huaici Zhao, Pengfei Liu, and Feidao Cao. Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. In *European Conference on Computer Vision*, pages 644–660. Springer, 2020. 2

[20] Xiang Li, Junbo Yin, Botian Shi, Yikang Li, Ruigang Yang, and Jianbin Shen. Lwsis: Lidar-guided weakly supervised instance segmentation for autonomous driving. *arXiv preprint arXiv:2212.03504*, 2022. 3

[21] Zhuoling Li, Zhan Qu, Yang Zhou, Jianzhuang Liu, Haoqian Wang, and Lihui Jiang. Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2791–2800, 2022. 2

[22] Qing Lian, Peiliang Li, and Xiaozhi Chen. Monojsg: Joint semantic and geometric cost volume for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1070–1079, 2022. 3

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017. 6

[24] Lijie Liu, Jiwen Lu, Chunjing Xu, Qi Tian, and Jie Zhou. Deep fitting degree scoring network for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1057–1066, 2019. 2, 7

[25] Xianpeng Liu, Nan Xue, and Tianfu Wu. Learning auxiliary monocular contexts helps monocular 3d object detection. In *AAAI Conference on Artificial Intelligence*, volume 36, pages 1810–1818, 2022. 2

[26] Zechen Liu, Zizhang Wu, and Roland Tóth. Smoke: Single-stage monocular 3d object detection via keypoint estimation.

In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 996–997, 2020. 2

[27] Yan Lu, Xinzhu Ma, Lei Yang, Tianzhu Zhang, Yating Liu, Qi Chu, Junjie Yan, and Wanli Ouyang. Geometry uncertainty projection network for monocular 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 3111–3121, 2021. 7

[28] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *European Conference on Computer Vision*, pages 311–327. Springer, 2020. 1

[29] Xinzhu Ma, Yinmin Zhang, Dan Xu, Dongzhan Zhou, Shuai Yi, Haojie Li, and Wanli Ouyang. Delving into localization errors for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4721–4730, 2021. 7

[30] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2069–2078, 2019. 7

[31] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Luc Van Gool, and Dengxin Dai. Weakly supervised 3d object detection from lidar point cloud. In *European Conference on Computer Vision*, pages 515–531. Springer, 2020. 3

[32] Qinghao Meng, Wenguan Wang, Tianfei Zhou, Jianbing Shen, Yunde Jia, and Luc Van Gool. Towards a weakly supervised framework for 3d point cloud object detection and annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4454–4468, 2021. 3

[33] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2

[34] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *IEEE/CVF International Conference on Computer Vision*, pages 3142–3152, 2021. 1, 2, 3, 6, 8

[35] Liang Peng, Senbo Yan, Boxi Wu, Zheng Yang, Xiaofei He, and Deng Cai. Weakm3d: Towards weakly supervised monocular 3d object detection. *arXiv preprint arXiv:2203.08332*, 2022. 3, 6, 7

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 3

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[38] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 8851–8858, 2019. 2, 7

[39] Zengyi Qin, Jinglu Wang, and Yan Lu. Weakly supervised 3d object detection from point clouds. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4144–4152, 2020. 3, 7

[40] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 4

[41] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. 8

[42] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1

[43] Kiwoo Shin, Youngwook Paul Kwon, and Masayoshi Tomizuka. Roarnet: A robust 3d object detection based on region approximation refinement. In *2019 IEEE intelligent vehicles symposium (IV)*, pages 2510–2515. IEEE, 2019. 1

[44] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection: From single to multi-class recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 8

[45] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1, 2, 5

[46] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019. 2

[47] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4604–4612, 2020. 1

[48] Li Wang, Liang Du, Xiaoqing Ye, Yanwei Fu, Guodong Guo, Xiangyang Xue, Jianfeng Feng, and Li Zhang. Depth-conditioned dynamic message propagation for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 454–463, 2021. 1

[49] Li Wang, Li Zhang, Yi Zhu, Zhi Zhang, Tong He, Mu Li, and Xiangyang Xue. Progressive coordinate transforms for monocular 3d object detection. *Advances in Neural Information Processing Systems*, 34:13364–13377, 2021. 1

[50] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 2

[51] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 3

[52] Yan Wang, Junbo Yin, Wei Li, Pascal Frossard, Ruigang Yang, and Jianbing Shen. Ssda3d: Semi-supervised domain adaptation for 3d object detection from point cloud. *arXiv preprint arXiv:2212.02845*, 2022. 3

[53] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1742–1749. IEEE, 2019. 1

[54] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Semi-supervised 3d object detection with proficient teachers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 727–743. Springer, 2022. 3

[55] Junbo Yin, Jianbing Shen, Xin Gao, David Crandall, and Ruigang Yang. Graph neural network and spatiotemporal transformer attention for 3d video object detection from point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 3

[56] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11495–11504, 2020. 3

[57] Junbo Yin, Dingfu Zhou, Liangjun Zhang, Jin Fang, Cheng-Zhong Xu, Jianbing Shen, and Wenguan Wang. Proposal-contrast: Unsupervised pre-training for lidar-based 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*, pages 17–33. Springer, 2022. 3

[58] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. *arXiv preprint arXiv:1906.06310*, 2019. 3

[59] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2403–2412, 2018. 6

[60] Sergey Zakharov, Wadim Kehl, Arjun Bhargava, and Adrien Gaidon. Autolabeling 3d objects with differentiable rendering of sdf shape priors. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12224–12233, 2020. 7

[61] Jesus Zarzar, Silvio Giancola, and Bernard Ghanem. Pointrgcn: Graph convolution networks for 3d vehicles detection refinement. *arXiv preprint arXiv:1911.12236*, 2019. 1

[62] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017. 3, 5

[63] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 7

[64] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 3

[65] Zhikang Zou, Xiaoqing Ye, Liang Du, Xianhui Cheng, Xiao Tan, Li Zhang, Jianfeng Feng, Xiangyang Xue, and Errui Ding. The devil is in the task: Exploiting reciprocal appearance-localization features for monocular 3d object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 2713–2722, 2021. 1