Lidar Point Cloud Guided Monocular 3D Object Detection

Liang Peng[1,2], Fei Liu[3], Zhengxu Yu[1], Senbo Yan[1,2], Dan Deng[2], Zheng Yang[2], Haifeng Liu[1], and Deng Cai[1,2] ✉

[1] State Key Lab of CAD&CG, Zhejiang University, China
{pengliang, senboyan, haifengliu}@zju.edu.cn yuzxfred@gmail.com
dengcai@cad.zju.edu.cn
[2] Fabu Inc., Hangzhou, China
{dengdan, yangzheng}@fabu.ai
[3] State Key Lab of Industrial Control and Technology, Zhejiang University, China
liufei21@zju.edu.cn

Abstract. Monocular 3D object detection is a challenging task in the self-driving and computer vision community. As a common practice, most previous works use manually annotated 3D box labels, where the annotating process is expensive. In this paper, we find that the precisely and carefully annotated labels may be unnecessary in monocular 3D detection, which is an interesting and counterintuitive finding. Using rough labels that are randomly disturbed, the detector can achieve very close accuracy compared to the one using the ground-truth labels. We delve into this underlying mechanism and then empirically find that: concerning the label accuracy, the 3D location part in the label is preferred compared to other parts of labels. Motivated by the conclusions above and considering the precise LiDAR 3D measurement, we propose a simple and effective framework, dubbed LiDAR point cloud guided monocular 3D object detection (LPCG). This framework is capable of either reducing the annotation costs or considerably boosting the detection accuracy without introducing extra annotation costs. Specifically, It generates pseudo labels from unlabeled LiDAR point clouds. Thanks to accurate LiDAR 3D measurements in 3D space, such pseudo labels can replace manually annotated labels in the training of monocular 3D detectors, since their 3D location information is precise. LPCG can be applied into any monocular 3D detector to fully use massive unlabeled data in a self-driving system. As a result, in KITTI benchmark, we take the first place on both monocular 3D and BEV (bird's-eye-view) detection with a significant margin. In Waymo benchmark, our method using 10% labeled data achieves comparable accuracy to the baseline detector using 100% labeled data. The codes are released at https://github.com/SPengLiang/LPCG.

Keywords: monocular 3D detection, LiDAR point cloud, self-driving.

1 Introduction

3D object detection plays a critical role in many applications, such as self-driving. It gives cars the ability to perceive the world in 3D, avoiding collisions with other

objects on the road. Currently, the LiDAR (Light Detection and Ranging) device is typically employed to achieve this [31,12,29,46], with the main shortcomings of the high price and limited working ranges. The single camera, as an alternative, is widely available and several orders of magnitude cheaper, consequently making monocular methods [41,1,6,45] popular in both industry and academia.

To the best of our knowledge, most previous monocular-based works [1,6,28,45] employ the precisely annotated 3D box labels. The annotation process operated on LiDAR point clouds is time-consuming and costly. In this paper, we empirically find that **the perfect manually annotated 3D box labels are not essential in monocular 3D detection**. We disturb the manually annotated labels by randomly shifting their values in a range, while the detector respectively trained by disturbed labels and perfect labels show very close performance. This is a counterintuitive finding. To explore the underlying mechanism, we divide a 3D box label into different groups according to its physical nature (including 3D locations, orientations, and dimensions of objects), and disturb each group of labels, respectively. We illustrate the experiment in Figure 1. The results indicate that the precise location label plays the most important role and dominates the performance of monocular 3D detection, and the accuracy of other groups of labels is not as important as generally considered. The underlying reason lies in the ill-posed nature of monocular imagery. It brings difficulties in recovering the 3D location, which is the bottleneck for the performance.

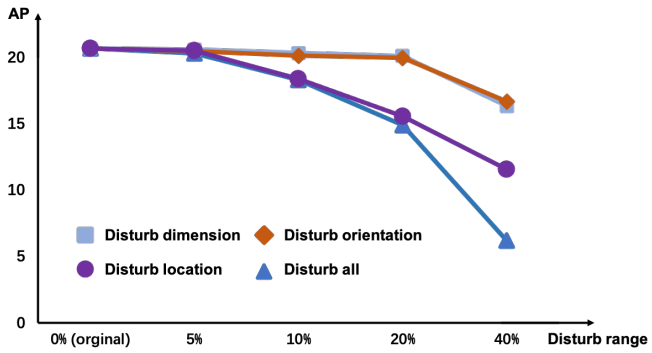

Fig. 1. We disturb the perfect manually annotated labels by randomly shifting the corresponding values within the percentage range. We can see that: 1) the disturbed labels (5%) and perfect labels lead to close accuracy; 2) the location dominates the overall accuracy (10%, 20%, 40%).

Unlike other classical computer vision tasks, manually annotating 3D boxes from monocular imagery is infeasible. It is because the depth information is lost during the camera projection process. Actually, the lost depth also is the reason why 3D location labels are the most important and difficult part for monocular

3D detection. LiDAR point clouds, which provide the crucial 3D measurements, are indispensable in the labeling procedure. As a common practice, annotators annotate 3D boxes on the LiDAR points clouds. On the other hand, concerning the data collecting process in a self-driving system, a large number of successive snippets are collected. Generally speaking, to save the high annotation costs, only some key frames in collected snippets are labeled to train networks, such as KITTI dataset [8]. Consequently, massive LiDAR point clouds holding valuable 3D information remain unlabeled.

Inspired by the 3D location label requirement and accurate LiDAR 3D measurements in 3D space, we propose a general and intuitive framework to make full use of LiDAR point clouds, dubbed **LPCG** (LiDAR point cloud guided monocular 3D object detection). Specifically, we use unlabeled LiDAR point clouds to generate pseudo labels, converting unlabeled data to training data for monocular 3D detectors. These pseudo labels are not as accurate as the manually annotated labels, but they are good enough for the training of monocular 3D detectors due to accurate LiDAR 3D measurements.

We further present two working modes in LPCG: the high accuracy mode and the low cost mode, to generate different qualities of pseudo labels according to annotation costs. The high accuracy mode requires a small amount of labeled data to train a LiDAR-based detector, and then the trained detector can produce high-quality pseudo labels on other unlabeled data. This manner can largely boost the accuracy of monocular 3D detectors. Additionally, we propose a heuristic method to produce pseudo labels without requiring any 3D box annotation. Such pseudo labels are directly obtained from the RoI LiDAR point clouds, by employing point clustering and minimum bounding box estimation. We call this manner the low cost mode. Either the high accuracy mode or the low cost mode in LPCG can be plugged into any monocular 3D detector.

Based on the above two modes, we can fully use LiDAR point clouds, allowing monocular 3D detectors to learn desired objectives on a large training set meanwhile avoiding architecture modification and removing extra annotation costs. By applying the framework (high accuracy mode), we significantly increase the 3D and BEV (bird's-eye-view) AP of prior state-of-the-art methods [27,1,13,11,45]. In summary, our contributions are two folds as below:

- **First**, we analyze requirements in terms of the label accuracy towards the training of monocular 3D detection. Based on this analysis, we introduce a general framework that can utilize massive unlabeled LiDAR point clouds, to generate new training data with valuable 3D information for monocular methods during the training.
- **Second**, experiments show that the baseline detector employing our method outperforms recent SOTA methods by a large margin, ranking 1^{st} on KITTI [8] monocular 3D and BEV detection benchmark at the time of submission (car, March. 2022). In Waymo [36] benchmark, our method achieves close accuracy compared to the baseline detector using 100% labeled data while our method requires only 10% labeled data with 90% unlabeled data.

2 Related Work

2.1 LiDAR-based 3D Object Detection

The LiDAR device can provide accurate depth measurement of the scene, thus is employed by most state-of-the-art 3D object detection methods [31,12,32,42,43,29]. These methods can be roughly divided into voxel-based methods and point-based methods. Voxel-based methods [47] first divide the point cloud into a voxel grid and then feed grouped points into fully connected layers, constructing unified feature representations. They then employ 2D CNNs to extract high-level voxel features to predict 3D boxes. By contrast, point-based methods [30,24] directly extract features on the raw point cloud via fully connected networks, such as PointNet [25] and PointNet++ [26]. SOTA 3D detection methods predominantly employ LiDAR point clouds both in training and inference, while we only use LiDAR point clouds in the training stage.

2.2 Image-only-based Monocular 3D Object Detection

As a commonly available and cheap sensor, the camera endows 3D object detection with the potential of being adopted everywhere. Thus monocular 3D object detection has become a very popular area of research and has developed quickly in recent years. Monocular works can be categorized into image-only-based methods [1,13] and depth-map-based methods [40,19,18] according to input representations. M3D-RPN [1] employs different convolution kernels in row-spaces that can explore different features in specific depth ranges and improve 3D estimates with the 2D-3D box consistency. Furthermore, RTM3D [13] predicts perspective key points and initial guesses of objects' dimensions/orientations/locations, where the key points are further utilized to refine the initial guesses by solving a constrained optimization problem. More recently, many image-only-based works utilize depth estimation embedding [45], differentiable NMS [11], and geometry properties [48,15,17], obtaining great success. There is also a related work [44] that introduces a novel autolabeling strategy of suggesting a differentiable template matching model with curriculum learning, using differentiable rendering of SDFs, while the pipeline is rather complicated.

2.3 Depth-map-based Monocular 3D Object Detection

Although monocular methods are developing quickly, a large performance gap still exists compared to LiDAR-based methods. Some prior works [40,41] argue that the improper choice of data representation is one of the main reasons, and propose to use transformed image-based depth maps. They first project LiDAR point clouds onto the image plane, to form depth map labels to train a depth estimator. Pseudo-LiDAR [40] converts the image into a point cloud by using an estimated depth map and then conducts 3D detection on it. They show promising results compared to previous image-only-based methods. Inspired by this, many later methods [41,19,6,18] also utilize off-the-shelf depth estimates to aim

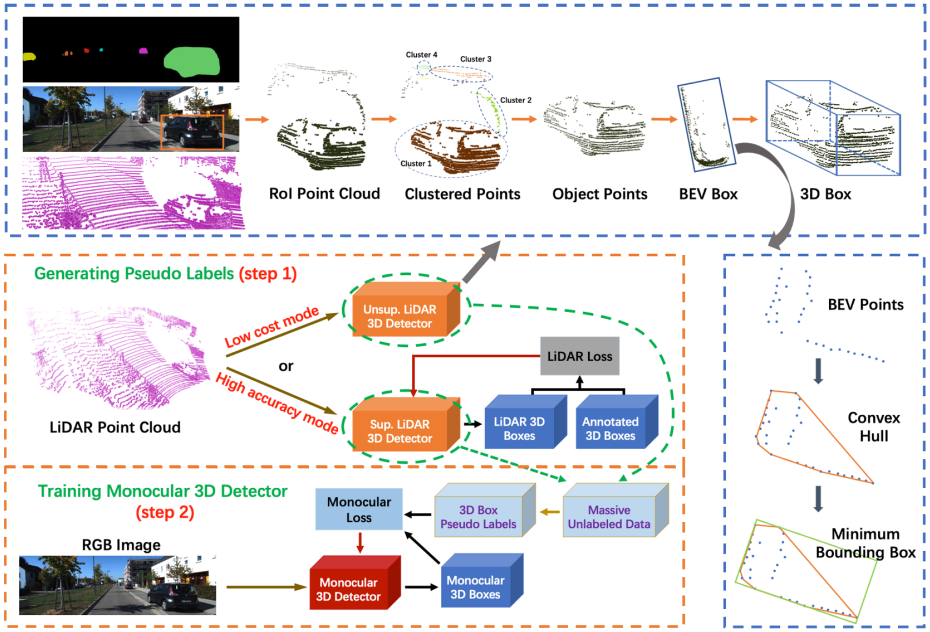

Fig. 2. Overview framework. We generate 3D box pseudo labels from unlabeled LiDAR point clouds, aiming to train the monocular 3D detector. Such 3D boxes are predicted via the well-trained LiDAR 3D detector (high accuracy mode) or obtained directly from the point cloud without training (low cost mode). "Unsup." and "Sup." in the figure denote unsupervised and supervised, respectively.

3D detection and gain performance improvements. More recently, CaDDN [28] integrates the dense depth estimation into monocular 3D detection, by using a predicted categorical depth distribution to project contextual features to the 3D space. Compared to previous depth-map-based methods, we aim to explore the potential of using LiDAR point clouds to generate pseudo labels for monocular 3D detectors.

3 LiDAR Guided Monocular 3D Detection

In this section, we detail the proposed framework, namely, LPCG (LiDAR Guided Monocular 3D Detection). First, as shown in Figure 1, the manually annotated perfect labels are unnecessary for monocular 3D detection. The accuracy led by disturbed labels (5%) is comparable to the one led by perfect labels. When enforcing large disturbances (10% and 20%), we can see that the location dominates the performance (the AP dramatically decreases only when disturbing the location). It indicates that rough pseudo 3D box labels with precise locations may replace the perfect annotated 3D box labels.

We note that LiDAR point clouds can provide valuable 3D location information. More specifically, LiDAR point clouds provide accurate depth measurement within the scene, which is crucial for 3D object detection as precise surrounding depths indicate locations of objects. Also, LiDAR point clouds can be easily captured by the LiDAR device, allowing a large amount of LiDAR point clouds to be collected offline without manual cost. Based on the analysis above, we use LiDAR point clouds to generate 3D box pseudo labels. The newly-generated labels can be used to train monocular 3D detectors. This simple and effective way allows monocular 3D detectors to learn desired objectives meanwhile eliminating annotation costs on unlabeled data. We show the overall framework in Figure 2, in which the method is able to work in two modes according to the reliance on 3D box annotations. If we use a small amount of 3D box annotations as prior, we call it the high accuracy mode since this manner leads to high performances. By contrast, we call it the low cost mode if we do not use any 3D box annotation.

3.1 High Accuracy Mode

To take advantage of available 3D box annotations, as shown in Figure 2, we first train a LiDAR-based 3D detector from scratch with LiDAR point clouds and associated 3D box annotations. The pre-trained LiDAR-based 3D detector is then utilized to infer 3D boxes on other unlabeled LiDAR point clouds. Such results are treated as pseudo labels to train monocular 3D detectors. We compare the pseudo labels with manually annotated perfect labels in Section 5.5. Due to precise 3D location measurements, pseudo labels predicted from the LiDAR-based 3D detector are rather accurate and qualified to be used directly in the training of monocular 3D detectors. We summarize the outline in Algorithm 1.

Algorithm 1: Outline of the high accuracy mode in LPCG. Both labeled and unlabeled training data contains RGB images and associated LiDAR point clouds.

Input: Labeled data $A : \{A_{data}, A_{label}\}$, unlabeled data $B : \{B_{data}\}$
Output: Well-trained monocular 3D detection model M_{mono}
$M_{lidar} \leftarrow$ Training a supervised LiDAR-based 3D detection model on labeled data $\{A_{data}, A_{label}\}$.
$\{B_{pseudo-label}\} \leftarrow$ Conducting predictions from LiDAR point clouds on unlabeled data: $M_{lidar}(B_{data})$
$C : \{C_{data}, C_{label}\} \leftarrow$ Merging training data: $\{A_{data} \cup B_{data}, A_{label} \cup B_{pseudo-label}\}$
$M_{mono} \leftarrow$ Training a supervised monocular-based model on new set $\{C_{data}, C_{label}\}$.
Return M_{mono}

Interestingly, with different training settings for the LiDAR-based 3D detector, we empirically find that monocular 3D detectors trained by resulting pseudo

labels show close performances. It indicates that monocular methods can indeed be beneficial from the guidance of the LiDAR point clouds and only a small number of 3D box annotations are sufficient to push the monocular method to achieve high performance. Thus the manual annotation cost of high accuracy mode is much lower than the one of the previous manner. Detailed experiments can be found in Section 5.6. Please note, the observations on label requirements and 3D locations are the core motivation of LPCG. The premise that LPCG can work well is that LiDAR points provide rich and precise 3D measurements, which offer accurate 3D locations.

3.2 Low Cost Mode

In this section, we describe the method of using LiDAR point clouds to eliminate the reliance on manual 3D box labels. First, an off-the-shelf 2D instance segmentation model [9] is adopted to perform segmentation on the RGB image, obtaining 2D box and mask estimates. These estimates are then used for building camera frustums in order to select associated LiDAR RoI points for every object, where those boxes without any LiDAR point inside are ignored. However, LiDAR points located in the same frustum consist of object points and mixed background or occluded points. To eliminate irrelevant points, we take advantage of DBSCAN [7] to divide the RoI point cloud into different groups according to the density. Points that are close in 3D spatial space will be aggregated into a cluster. We then regard the cluster containing most points as a target corresponding to the object. Finally, we seek the minimum 3D bounding box that covers all target points.

To simplify the problem of solving the 3D bounding box, we project points onto the bird's-eye-view map, reducing parameters since the height (h) and y coordinate (under camera coordinate system) of the object can be easily obtained. Therefore, we have:

$$L = \min_{B_{bev}}(Area(B_{bev})), \quad \textit{subject to } p \textit{ is inside } B_{bev}\textit{, where } p \in LiDAR_{RoI} \quad (1)$$

where B_{bev} refers to a bird's-eye-view (BEV) box. We solve this problem by using the convex hull of object points followed by obtaining the box by using rotating calipers [37]. Furthermore, the height h can be represented by the max spatial offset along the y-axis of points, and the center coordinate y is calculated by averaging y coordinates of points. We use a simple rule of restricting object dimensions to remove outliers. The overall training pipeline for monocular methods is summarized in Algorithm 2.

4 Applications in Real-world Self-driving System

In this section, we describe the application of LPCG to a real-world self-driving system. First, we illustrate the data collecting strategy in Figure 3. Most self-driving systems can easily collect massive unlabeled LiDAR point cloud data

Algorithm 2: Outline of the low cost mode in LPCG. Unlabeled data contains RGB images and associated LiDAR point clouds.

Input: Unlabeled data $D : \{D_{data-image}, D_{data-lidar}\}$, pre-trained Mask-RCNN model: M_{mask}
Output: Well-trained monocular 3D detection model M_{mono}
$Mask_{2D} \leftarrow$ Conducting predictions from RGB images: $M_{mask}(D_{data-image})$.
$LiDAR_{RoI} \leftarrow$ Selecting and clustering LiDAR point clouds from $D_{data-lidar}$ by $Mask_{2D}$
$D_{pseudo-label} \leftarrow$ Gernerating pseudo labels on RoI LiDAR points $LiDAR_{RoI}$.
$M_{mono} \leftarrow$ Training a supervised monocular-based model on new data $\{D_{data-image}, D_{pseudo-label}\}$.
Return M_{mono}

Table 1. Comparisons of different modes in previous works and ours.

Approaches	Modality	3D box annotations	Unlabeled LiDAR data
Previous	Image-only based	Yes	No
	Depth-map based	Yes	Yes
Ours	High accuracy mode	Yes	Yes
	Low cost mode	No	Yes

and synchronized RGB images. This data is organized with many sequences, where each sequence often refers to a specific scene and contains many successive frames. Due to the limited time and resources in the real world, only some sequences are chosen for annotating, to train the network, such as Waymo [36]. Further, to reduce the high annotation costs, only some key frames in the selected sequences are annotated, such as KITTI [8]. Therefore, there remains massive unlabeled data in real-world applications.

Considering that LPCG can fully take advantage of the unlabeled data, it is natural to be employed in a real-world self-driving system. Specifically, the high accuracy mode only requires a small amount of labeled data. Then we can generate high-quality training data from remaining unlabeled data for monocular 3D detectors, to boost the accuracy. In experiments, we quantitatively and qualitatively show that the generated 3D box pseudo labels are good enough for monocular 3D detectors. Additionally, the low cost mode does not require any 3D box annotation, still providing accurate 3D box pseudo labels. We compare LPCG with previous methods in Table 1 in terms of the data requirements.

5 Experiments

5.1 Implementation Details

We use the image-only-based monocular 3D detector M3D-RPN [1], and adopt PV-RCNN [29] as the LiDAR 3D detector for the high accuracy mode. We filter out 3D boxes generated from LiDAR point clouds with the confidence of

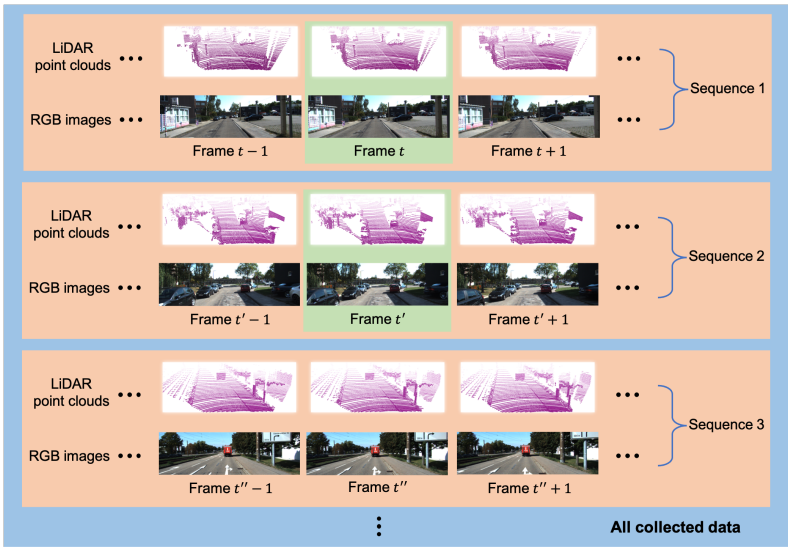

Fig. 3. Data collecting strategy in a real-world system. Only some sequences (*e.g.*, sequence 1 and 2) are chosen for annotating because of the limited time and resources in the real world, such as Waymo [36]. Further, concerning the high annotation costs, only some key frames (*e.g.*, frame t and t') in the selected sequences are annotated, such as KITTI [8].

0.7. Experiments on other methods are conducted by the official code that is publicly available, and all settings keep the same as the original paper. During the process of using LiDAR point clouds to train monocular 3D detectors, the learning iteration is scaled according to the number of training data. The high accuracy mode is employed by default. For the low cost mode, we use Mask-RCNN [9] pre-trained in the COCO dataset [14], and filter the final 3D bounding box by the width range of 1.2-1.8 meters and the length range of 3.2-4.2 meters. We filter out 2D boxes predicted from Mask-RCNN [9] with the confidence of 0.9. More details and ablations are provided in the supplementary material as the space limitation.

5.2 Dataset and Metrics

Dataset. Following prior works [27,35,1,40,13,18], experiments are conducted on the popular KITTI 3D object dataset [8], which contains 7, 481 manually annotated images for training and 7, 518 images for testing. Due to groundtruths of the test set are not available, the public training set is further split into two subsets [3]: training set (3, 712 images) and validation set (3, 769 images). Following the fashion, we report our results both on the validation set and the test set. And we use the validation set for all ablations. Also, our method and depth-map-based methods use RGB images and synchronized LiDAR point clouds from

Table 2. Comparisons on KITTI testing set. We use red to indicate the highest result and blue for the second-highest result and cyan for the third-highest result. ‡ denotes the baseline detector we employed, and the improvements are relative to the baseline detectors. We define the new state of the art. Please note, DD3D[22]⋆ employs both the large private DDAD15M dataset (containing approximately 15M frames) and the KITTI depth dataset (containing approximately 26K frames).

Approaches	Extra data	$AP_{BEV}/AP_{3D}\vert$ (IoU=0.7)$\vert_{R_{40}}$ Easy	Moderate	Hard
ROI-10D [21]	KITTI depth	9.78/4.32	4.91/2.02	3.74/1.46
MonoGRNet [27]	None	18.19/5.74	11.17/9.61	8.73/4.25
AM3D [19]	KITTI depth	25.03/16.50	17.32/10.74	14.91/9.52
MonoPair [4]	None	19.28/13.04	14.83/9.99	12.89/8.65
D4LCN [6]	KITTI depth	22.51/16.65	16.02/11.72	12.55/9.51
RTM3D [13]	None	19.17/14.41	14.20/10.34	11.99/8.77
PatchNet [18]	KITTI depth	22.97/15.68	16.86/11.12	14.97/10.17
Neighbor-Vote [5]	KITTI depth	27.39/15.57	18.65/9.90	16.54/8.89
MonoRUn [2]	None	27.94/19.65	17.34/12.30	15.24/10.58
MonoRCNN [33]	None	25.48/18.36	18.11/12.65	14.10/10.03
Monodle [20]	None	24.79/17.23	18.89/12.26	16.00/10.29
CaDDN [28]	None	27.94/19.17	18.91/13.41	17.19/11.46
Ground-Aware [15]	None	29.81/21.65	17.98/13.25	13.08/9.91
GrooMeD-NMS [11]	None	26.19/18.10	18.27/12.32	14.05/9.65
MonoEF [48]	None	29.03/21.29	19.70/13.87	**17.26**/11.71
DDMP-3D [38]	KITTI depth	28.08/19.71	17.89/12.78	13.44/9.80
PCT [39]	KITTI depth	29.65/21.00	19.03/13.37	15.92/11.31
AutoShape [16]	None	**30.66/22.47**	20.08/14.17	15.95/11.36
GUPNet [17]	None	30.29/22.26	**21.19/15.02**	**18.20/13.12**
M3D-RPN [1] ‡	None	21.02/14.76	13.67/9.71	10.23/7.42
MonoFlex [45] ‡	None	28.23/19.94	19.75/13.89	16.89/12.07
DD3D [22] ⋆	*DDAD15M...*	*32.35/23.19*	*23.41/16.87*	*20.42/14.36*
LPCG+M3D-RPN[1]	KITTI depth	**30.72/22.73**	**20.17/14.82**	16.76/**12.88**
Improvements (to baseline)		+9.70/+7.97	+6.50/+5.11	+6.53/+5.46
LPCG+MonoFlex[45]	KITTI depth	**35.96/25.56**	**24.81/17.80**	**21.86/15.38**
Improvements (to baseline)		+7.73/+5.62	+5.06/+3.91	+4.97/+3.31

KITTI raw scenes. For depth-map-based methods, note that the original depth training set overlaps KITTI 3D detection validation set. Therefore we exclude scenes that emerge in KITTI 3D validation set to avoid data leakage [34,23]. LiDAR point clouds in the remaining scenes are used. We call this extra dataset the KITTI depth dataset. It provides approximately 26K samples to train the depth estimator (for most depth-map-based methods) or to generate extra training samples for monocular 3D detectors (LPCG).

Additionally, to further validate the effectiveness of LPCG, we conduct experiments on the Waymo Open Dataset [36], which is a modern large dataset. It contains 798 training sequences and 202 validation sequences, and we adopt the same data processing strategy proposed in CaDDN [28]. The sampled training dataset includes approximately 50K training samples with manual annotations.

Metrics. Each manually annotated object is divided into easy, moderate, and hard levels according to the occlusion, truncation, and 2D box height [8]. Average precisions (AP) on the car class for bird's-eye-view (BEV) and 3D boxes with 0.5/0.7 IoU thresholds are commonly used metrics for monocular 3D detection.

Many previous methods utilize the AP_{11} metric, which has an overrated issue [35], and AP_{40} [35] is proposed to resolve it. We report AP_{40} results to make comprehensive comparisons. For Waymo dataset, we adopt the official mAP and mAPH metrics.

Table 3. Comparisons with SDFLabel [44]. Note that here we use the same number of training samples for fair comparisons.

Approaches	Data requirements in training	AP_{3D} (IoU=0.7)|$_{R_{40}}$ Easy	Moderate	Hard
SDFLabel [44]	2D masks+LiDAR+CAD models	1.23	0.54	-
LPCG (low cost mode)	2D masks+LiDAR	**5.36**	**3.07**	**2.32**

5.3 Results on KITTI

We evaluate LPCG on KITT test set using two base monocular detectors[1,45] with the high accuracy mode. Table 2 shows quantitative results in *test* set. Due to the space limitation, qualitative results are included in the supplementary material. We can observe that our method increases the current SOTA BEV/3D AP from **21.19/15.02** to **24.81/17.80** under the moderate setting, which is rather significant. Even using a monocular detector [1] proposed in 2019, our method still allows it to achieve new state-of-the-art compared to prior works. Note that our method still performs better, while DD3D [22] employs both the large private DDAD15M dataset (containing approximately 15M frames) and the KITTI depth dataset (containing approximately 20K frames). Also, we boost the performance on pedestrian and cyclist categories of the original method, and provide the results in Table 5. Such results prove the effectiveness of LPCG.

For the low cost mode, we note that there are few works exploring this area, namely, few works have explored monocular 3D detection without any 3D box annotation. The most related work is SDFLabel [44], which also does not require 3D box annotation. Thus we compare LPCG with the low cost mode with SDFLabel [44] in Table 3. Please note, in this experiment we use the same number of training samples, namely, the 3, 769 samples in KITTI3D training set. Our method outperforms it by a large margin, and our method is more generally usable as our pipeline is much simpler than SDFLabel [44].

5.4 Results on Waymo

To further prove the effectiveness of our method, we conduct experiments on the Waymo open dataset. Concerning its large scale, when enough perfect labels are available, in this dataset we aim to investigate the performance gap between the generated pseudo labels and the manual 3D box annotations. More specifically, we use the baseline detector M3D-RPN [1], training it with pseudo labels and

Table 4. Comparisons on Waymo. "lab." and "unlab." denote labeled and unlabeled.

Difficulty	w/ LPCG	Data requirements	Overall	0−30m	30−50m	50m−∞
under 3D mAP metric						
LEVEL 1 (IOU=0.5)	No	10% labeled data	4.14	14.64	1.63	0.04
	No	100% labeled data	6.42	19.50	3.04	0.17
	Yes	10% lab. + 90% unlab. data	6.23	18.39	3.44	0.19
LEVEL 2 (IOU=0.5)	No	10% labeled data	3.88	14.59	1.58	0.04
	No	100% labeled data	6.02	19.43	2.95	0.15
	Yes	10% lab. + 90% unlab. data	5.84	18.33	3.34	0.17
under 3D mAPH metric						
LEVEL 1 (IOU=0.5)	No	10% labeled data	3.94	14.07	1.51	0.04
	No	100% labeled data	6.19	18.88	2.89	0.16
	Yes	10% lab. + 90% unlab. data	6.09	18.03	3.33	0.17
LEVEL 2 (IOU=0.5)	No	10% labeled data	3.69	14.02	1.46	0.03
	No	100% labeled data	5.80	18.81	2.80	0.14
	Yes	10% lab. + 90% unlab. data	5.70	17.97	3.23	0.15

Table 5. Improvements on other categories.

Approaches	Pedestrian, AP_{3D} (IoU=0.5)|R_{40}			Cyclist, AP_{3D} (IoU=0.5)|R_{40}		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN [1]	4.75	3.55	2.79	3.10	1.49	1.17
M3D-RPN+LPCG	**7.21**	**5.53**	**4.46**	**4.83**	**2.65**	**2.62**

manual annotations, respectively. We report the results in Table 4. Pseudo labels on unlabeled data are generated by the high accuracy mode in LPCG. Interestingly, we can see that the detector using 10% labeled data and 90% unlabeled data achieves comparable accuracy to the one using 100% labeled data (*e.g.*, 6.42 *vs.* 6.23 and 6.19 *vs.* 6.09). This result demonstrates the generalization ability of LPCG, indicating that LPCG can also reduce the annotation costs for the large scale dataset with slight accuracy degradation.

5.5 Comparisons on Pseudo Labels and Manually Annotated Labels

As expected, pseudo labels are not as accurate as manually annotated labels. It is interesting to quantitatively evaluate pseudo labels using manually annotated labels. We report the results in Table 6. TP, FP, FN are calculated by matching pseudo labels and annotated labels. Regarding matched objects, we average the relative error (MRE) on each group of 3D box parameters (location, dimension, and orientation). We can see that pseudo labels from the high accuracy mode can match most real objects (91.39%), and the mean relative errors are 1%-6%. Therefore pseudo labels from the high accuracy mode are good enough for monocular 3D detectors. Actually, experiments in Table 2 and 4 also verify the effectiveness. On the other hand, for the low cost mode, we can see that many real objects are missed (11834). We note that missed objects are often truncated, occluded, or faraway. The attached LiDAR points cannot indicate the full 3D outline of objects, thus they are hard to recover by geometry-based methods.

Table 6. Performance of pseudo labels on *val* set. We evaluate pseudo labels using manually annotated labels. "MRE" refers to the mean relative error (*e.g.*, the relative error of location is $\frac{Error_{Loc}}{Loc}$). "Loc., Dim., Orient." are the location (x, y, z), dimension (h, w, l), and orientation (R_y). "TP, FP, FN" are the true positive, false positive, and false negative, which are calculated by matching pseudo labels and annotated labels. Please see Section 5.5 for detailed analysis. Note that pseudo labels on *val* set are just for the evaluation, and they are not used in the training of monocular 3D detectors.

Pseudo label types	TP	FP	FN	Loc. MRE	Dim. MRE	Orient. MRE
Low cost mode	2551	161	11834	4%/5%/2%	8%/6%/7%	8%
High accuracy mode	13146	3299	1239	4%/4%/1%	4%/4%/6%	4%

Table 7. Ablation for annotation numbers. 3712 is the total annotations in the KITTI 3D training dataset. All results are evaluated on KITTI *val* set with metric $AP|_{R_{40}}$.

Annotations	$AP_{BEV}/AP_{3D}\|$ (IoU=0.7)$_{R_{40}}$ Easy	Moderate	Hard
100	30.19/20.90	21.96/15.37	19.16/13.00
200	30.39/22.55	22.44/16.17	19.60/14.32
500	32.01/23.13	23.31/17.42	20.26/14.95
1000	33.08/25.71	24.89/19.29	21.94/16.75
3712	**33.94/26.17**	**25.20/19.61**	**22.06/16.80**

5.6 Ablation Studies

We conduct the ablation studies on KITTI *val* set. Because of the space limitation, we provide extra ablation studies in the supplementary material.

Different Monocular Detectors. We plug LPCG into different monocular 3D detectors [27,1,13,11,45], to show its extension ability. Table 8 shows the results. We can see that LPCG obviously and consistently boosts original performances, *e.g.*, 7.57 → 10.06 for MonoGRNet [27], 10.06 → 19.43 for RTM3D [13], and 14.32 → 20.46 for GrooMeD-NMS [11] under the moderate setting (AP_{3D} (IoU=0.7)). Furthermore, we explore the feasibility of using a rather simple model when large data is available. We perform this experiment on RTM3D [13] with ResNet18 [10] backbone, which achieves 46.7 FPS on a NVIDIA 1080Ti GPU. [1] To the best of our knowledge, it is the simplest and fastest model for monocular 3D detection. With employing LPCG, this simple model obtains very significant improvements. LPCG endows it (46.7 FPS) with the comparable accuracy to other state-of-the-art models (*e.g.*, GrooMeD-NMS (8.3 FPS)). These results prove that LPCG is robust to the choice of monocular 3D detectors.

The Number of Annotations. We also investigate the impact of the number of annotations. We report the results in Table 7. The results indicate that a small number of annotations in LPCG can also lead to high accuracy for monocular

[1] From RTM3D official implementation.

Table 8. Extension on different monocular detectors. LPCG can be easily plugged into other methods. * denotes that the model is re-implemented by us. All the methods are evaluated on KITTI *val* set with metric $AP|_{R_{40}}$.

Approaches	AP_{3D} (IoU=0.5)$\|_{R_{40}}$			AP_{3D} (IoU=0.7)$\|_{R_{40}}$		
	Easy	Moderate	Hard	Easy	Moderate	Hard
MonoGRNet [27]	47.34	32.32	25.54	11.93	7.57	5.74
MonoGRNet+LPCG	**53.84**	**37.24**	**29.70**	**16.30**	**10.06**	**7.86**
Improvements	+6.50	+4.92	+4.16	+4.37	+2.49	+2.12
M3D-RPN [1]	48.56	35.94	28.59	14.53	11.07	8.65
M3D-RPN+LPCG	**62.92**	**47.14**	**42.03**	**26.17**	**19.61**	**16.80**
Improvements	+14.36	+11.20	+13.44	+11.64	+8.54	+8.15
RTM3D [13]	55.44	39.24	33.82	13.40	10.06	9.07
RTM3D+LPCG	**65.44**	**49.40**	**43.55**	**25.23**	**19.43**	**16.77**
Improvements	+10.00	+10.16	+9.73	+11.83	+9.37	+7.70
RTM3D (ResNet18) [13]	47.78	33.75	28.48	10.85	7.51	6.33
RTM3D (ResNet18)+LPCG	**62.98**	**45.86**	**41.63**	**22.69**	**16.78**	**14.50**
Improvements	+15.20	+12.11	+13.15	+11.84	+9.27	+8.17
GrooMeD-NMS [11]	55.62	41.07	32.89	19.67	14.32	11.27
GrooMeD-NMS +LPCG	**68.27**	**50.80**	**45.14**	**27.79**	**20.46**	**17.75**
Improvements	+12.65	+9.73	+12.25	+8.12	+6.14	+6.48
MonoFlex [45]*	56.73	42.97	37.34	20.02	15.19	12.95
MonoFlex +LPCG	**69.16**	**54.27**	**48.37**	**31.15**	**23.42**	**20.60**
Improvements	+12.43	+11.30	+11.03	+11.13	+8.23	+7.65

3D detectors. For example, the detector using 1000 annotations performs close to the full one (24.89/19.29 *vs.* 25.20/19.61 under the moderate setting).

6 Conclusion

In this paper, we first analyze the label requirements for monocular 3D detection. Experiments show that disturbed labels and perfect labels can lead to very close performance for monocular 3D detectors. With further exploration, we empirically find that the 3D location is the most important part of 3D box labels. Additionally, a self-driving system can produce massive unlabeled LiDAR point clouds, which have precise 3D measurements. Therefore, we propose a framework (LCPG), to generate pseudo 3D box labels on unlabeled LiDAR point clouds, to enlarge the training set of monocular 3D detectors. Extensive experiments on various datasets validate the effectiveness of LCPG. Furthermore, the main limitation of LCPG is more training time due to the increased training samples.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, U1909203, 61936006, 61973271), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05).

References

1. Brazil, G., Liu, X.: M3d-rpn: Monocular 3d region proposal network for object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 9287–9296 (2019)
2. Chen, H., Huang, Y., Tian, W., Gao, Z., Xiong, L.: Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10379–10388 (2021)
3. Chen, X., Kundu, K., Zhu, Y., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals using stereo imagery for accurate object class detection. IEEE transactions on pattern analysis and machine intelligence **40**(5), 1259–1272 (2017)
4. Chen, Y., Tai, L., Sun, K., Li, M.: Monopair: Monocular 3d object detection using pairwise spatial relationships. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12093–12102 (2020)
5. Chu, X., Deng, J., Li, Y., Yuan, Z., Zhang, Y., Ji, J., Zhang, Y.: Neighbor-vote: Improving monocular 3d object detection through neighbor distance voting. arXiv preprint arXiv:2107.02493 (2021)
6. Ding, M., Huo, Y., Yi, H., Wang, Z., Shi, J., Lu, Z., Luo, P.: Learning depth-guided convolutions for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11672–11681 (2020)
7. Ester, M., Kriegel, H.P., Sander, J., Xu, X., et al.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Kdd. vol. 96, pp. 226–231 (1996)
8. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361. IEEE (2012)
9. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
11. Kumar, A., Brazil, G., Liu, X.: Groomed-nms: Grouped mathematically differentiable nms for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8973–8983 (2021)
12. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12697–12705 (2019)
13. Li, P., Zhao, H., Liu, P., Cao, F.: Rtm3d: Real-time monocular 3d detection from object keypoints for autonomous driving. arXiv preprint arXiv:2001.03343 (2020)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
15. Liu, Y., Yixuan, Y., Liu, M.: Ground-aware monocular 3d object detection for autonomous driving. IEEE Robotics and Automation Letters **6**(2), 919–926 (2021)
16. Liu, Z., Zhou, D., Lu, F., Fang, J., Zhang, L.: Autoshape: Real-time shape-aware monocular 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15641–15650 (2021)
17. Lu, Y., Ma, X., Yang, L., Zhang, T., Liu, Y., Chu, Q., Yan, J., Ouyang, W.: Geometry uncertainty projection network for monocular 3d object detection. In:

Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3111–3121 (2021)
18. Ma, X., Liu, S., Xia, Z., Zhang, H., Zeng, X., Ouyang, W.: Rethinking pseudo-lidar representation. arXiv preprint arXiv:2008.04582 (2020)
19. Ma, X., Wang, Z., Li, H., Zhang, P., Ouyang, W., Fan, X.: Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6851–6860 (2019)
20. Ma, X., Zhang, Y., Xu, D., Zhou, D., Yi, S., Li, H., Ouyang, W.: Delving into localization errors for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4721–4730 (2021)
21. Manhardt, F., Kehl, W., Gaidon, A.: Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2069–2078 (2019)
22. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)
23. Peng, L., Liu, F., Yan, S., He, X., Cai, D.: Ocm3d: Object-centric monocular 3d object detection. arXiv preprint arXiv:2104.06041 (2021)
24. Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J.: Frustum pointnets for 3d object detection from rgb-d data. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 918–927 (2018)
25. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
26. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
27. Qin, Z., Wang, J., Lu, Y.: Monogrnet: A geometric reasoning network for monocular 3d object localization. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 8851–8858 (2019)
28. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8555–8564 (2021)
29. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10529–10538 (2020)
30. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 770–779 (2019)
31. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. arXiv preprint arXiv:1907.03670 (2019)
32. Shi, W., Rajkumar, R.: 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. pp. 1711–1719 (2020)
33. Shi, X., Ye, Q., Chen, X., Chen, C., Chen, Z., Kim, T.K.: Geometry-based distance decomposition for monocular 3d object detection. arXiv preprint arXiv:2104.03775 (2021)
34. Simonelli, A., Bulo, S.R., Porzi, L., Kontschieder, P., Ricci, E.: Are we missing confidence in pseudo-lidar methods for monocular 3d object detection? In: Proceedings

of the IEEE/CVF International Conference on Computer Vision. pp. 3225–3233 (2021)
35. Simonelli, A., Bulo, S.R., Porzi, L., López-Antequera, M., Kontschieder, P.: Disentangling monocular 3d object detection. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1991–1999 (2019)
36. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2446–2454 (2020)
37. Toussaint, G.T.: Solving geometric problems with the rotating calipers. In: Proc. IEEE Melecon. vol. 83, p. A10 (1983)
38. Wang, L., Du, L., Ye, X., Fu, Y., Guo, G., Xue, X., Feng, J., Zhang, L.: Depth-conditioned dynamic message propagation for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 454–463 (2021)
39. Wang, L., Zhang, L., Zhu, Y., Zhang, Z., He, T., Li, M., Xue, X.: Progressive coordinate transforms for monocular 3d object detection. Advances in Neural Information Processing Systems **34** (2021)
40. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8445–8453 (2019)
41. Weng, X., Kitani, K.: Monocular 3d object detection with pseudo-lidar point cloud. In: Proceedings of the IEEE International Conference on Computer Vision Workshops. pp. 0–0 (2019)
42. Yang, Z., Sun, Y., Liu, S., Jia, J.: 3dssd: Point-based 3d single stage object detector. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11040–11048 (2020)
43. Ye, M., Xu, S., Cao, T.: Hvnet: Hybrid voxel network for lidar based 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1631–1640 (2020)
44. Zakharov, S., Kehl, W., Bhargava, A., Gaidon, A.: Autolabeling 3d objects with differentiable rendering of sdf shape priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12224–12233 (2020)
45. Zhang, Y., Lu, J., Zhou, J.: Objects are different: Flexible monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3289–3298 (2021)
46. Zheng, W., Tang, W., Jiang, L., Fu, C.W.: Se-ssd: Self-ensembling single-stage object detector from point cloud. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14494–14503 (2021)
47. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4490–4499 (2018)
48. Zhou, Y., He, Y., Zhu, H., Wang, C., Li, H., Jiang, Q.: Monocular 3d object detection: An extrinsic parameter free approach. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7556–7566 (2021)