PointAugmenting: Cross-Modal Augmentation for 3D Object Detection

Chunwei Wang, Chao Ma[*], Ming Zhu, and Xiaokang Yang
MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
{weiwei0224, chaoma, droplet-to-ocean, xkyang}@sjtu.edu.cn

Abstract

Camera and LiDAR are two complementary sensors for 3D object detection in the autonomous driving context. Camera provides rich texture and color cues while LiDAR specializes in relative distance sensing. The challenge of 3D object detection lies in effectively fusing 2D camera images with 3D LiDAR points. In this paper, we present a novel cross-modal 3D object detection algorithm, named PointAugmenting. On one hand, PointAugmenting decorates point clouds with corresponding point-wise CNN features extracted by pretrained 2D detection models, and then performs 3D object detection over the decorated point clouds. In comparison with highly abstract semantic segmentation scores to decorate point clouds, CNN features from detection networks adapt to object appearance variations, achieving significant improvement. On the other hand, PointAugmenting benefits from a novel cross-modal data augmentation algorithm, which consistently pastes virtual objects into images and point clouds during network training. Extensive experiments on the large-scale nuScenes and Waymo datasets demonstrate the effectiveness and efficiency of our PointAugmenting. Notably, PointAugmenting outperforms the LiDAR-only baseline detector by $+6.5\%$ *mAP and achieves the new state-of-the-art results on the nuScenes leaderboard to date.*

1. Introduction

3D object detection plays a crucial role in 3D scene understanding for autonomous driving. Existing 3D object detection algorithms mainly use LiDAR and cameras to perceive environments. LiDAR grasps depth information in the form of sparse point clouds, while cameras capture images in the form of dense intensity array with rich color and textures. The challenge of 3D object detection lies in the misalignment between images and point clouds. In this work, we aim to advance 3D object detection by means of effective cross-modal data fusion and augmentation.

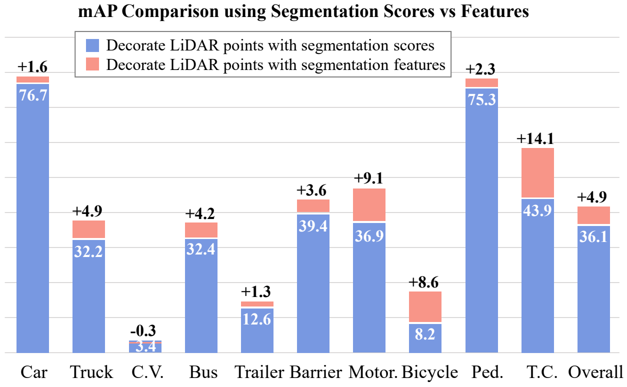

Figure 1. Performance comparison between using segmentation scores and CNN features to fuse with LiDAR points for 3D object detection. We replace the segmentation scores in PointPainting [19] with middle CNN features extracted from the same segmentation network [20]. We use the LiDAR-only detector CenterPoint [27] as baseline. The improvement of $+4.9\%$ mAP on the 1/8 nuScenes dataset suggests that CNN features from images are better at fusing with point clouds. Abbreviations stand for construction vehicle (C.V.), motorcycle (Motor.), pedestrian (Ped.), and traffic cone (T.C.).

Previous arts have explored a variety of cross-modal fusion schemes, which fall into three categories: result-level fusion, proposal-level fusion, and point-level fusion. The result-level fusion approaches [13, 21] adopt off-the-shelf 2D object detectors, thus their performances are limited by the upper bound of 2D detectors. The proposal-level fusion methods, such as MV3D [3] and AVOD [8], perform fusion at the region proposal level, resulting in heavy computation load. Recent approaches [11, 10, 29, 16, 7, 19] attempt to fetch point-wise image features by projecting point clouds onto image plane. [11, 10, 29] construct birds-eye-view (BEV) camera features before fusing with LiDAR BEV features to mitigate the viewpoint inconsistency. However, the cross-view transformation readily causes feature blurring. Instead, MVX-Net [16], EPNet [7] and PointPainting [19] directly exploit point-wise correspondence to augment each LiDAR point with CNN features or segmentation scores from image segmentation. We note that prior fusion meth-

[*] Corresponding author.

Method	Car	Truck	C.V.	Bus	Trailer	Barrier	Motor.	Bicycle	Ped.	T.C.	mAP	NDS
CenterPoint w/o GT-Paste	74.2	30.9	3.7	27.0	12.5	37.2	30.3	1.7	68.2	42.4	32.8	42.3
CenterPoint w/ GT-Paste	78.6	39.2	2.0	33.5	13.5	46.8	32.2	8.6	74.2	47.5	37.6	49.5
Gains of GT-Paste	+4.4	+8.3	-1.7	+6.5	+1.0	+9.6	+1.9	+6.9	+6.0	+5.1	+4.8	+7.2

Table 1. Effectiveness of the GT-Paste data augmentation scheme. Applying GT-Paste data augmentation for LiDAR points achieves an improvement of $+4.8\%$ mAP. We use CenterPoint as baseline with 1/8 training data on the nuScenes dataset.

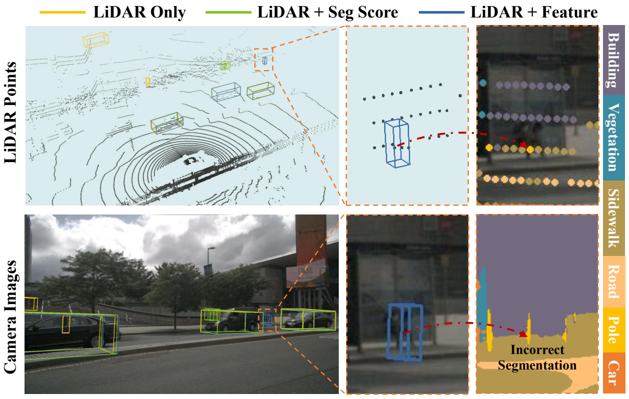

Figure 2. Comparison of different detectors. The far-away pedestrian in the scene is missed by the LiDAR-only baseline detector due to the sparse point clouds. PointPainting also fails as a result of segmentation failures on small objects. Benefiting from the abundant semantics provided by image features, our method successfully detects the pedestrian.

ods before PointPainting have limited generalization and performance, as concluded by PointPainting, “despite recent fusion research, the top methods on the popular KITTI leaderboard are lidar only”. With the help of segmentation scores, PointPainting has served as a popular baseline of fusion with large gains over the LiDAR-only detectors on large-scale datasets.

Despite the impressive improvements, segmentation scores are sub-optimal to cover color and textures in images. Intuitively, high dimensional CNN features of images contain richer appearance cues and larger receptive field than segmentation scores, therefore are more complementary to fuse with point clouds. To validate this intuition, we conduct a preliminary experiment on the basis of PointPainting. Specifically, we replace the segmentation scores with CNN features extracted by the same segmentation network [20]. Figure 1 shows that CNN features help PointPainting to achieve a significant gain of $+4.9\%$ mAP on the 1/8 nuScenes [2] dataset. This manifests the effectiveness of CNN features to fuse with point clouds for 3D detectors.

Considering the lack of ground truth segmentation labels for most 3D detection tasks, we use pretrained 2D object detection networks rather than image segmentation networks as feature extractor. Our method differs from the prior 3D detector MVX-Net [16], which utilizes the high-level semantics on the Conv5 layer of VGG-16. High-level semantics often cause blurred image features for neighboring LiDAR points. We thus take the output activation from the DLA34 layer of CenterNet [33, 32] as image features, putting emphasis on fine-grained details to strengthen the distinction between point clouds. Moreover, considering the modality gap between LiDAR and cameras, we employ a late fusion mechanism across modalities. With our fusion scheme, we achieve remarkable improvements of $+10.1\%$ and $+5.2\%$ mAP respectively over the LiDAR-only and PointPainting methods on the 1/8 nuScenes dataset. The example in Figure 2 illustrates the superiority of our method.

When training 3D detectors, one of the bottlenecks lies in cross-modal data augmentation. Existing LiDAR-only detectors widely use GT-Paste [22], a data augmentation scheme to augment point clouds. GT-Paste pastes virtual objects in the forms of ground-truth boxes and LiDAR points from other scenes to the training scenes. Table 1 shows that GT-Paste improves the LiDAR-only detector by $+4.8\%$ mAP. However, directly applying GT-Paste to cross-modal detectors would destruct the consistency between LiDAR points and camera images. To address this issue, we propose a simple yet effective cross-modal augmentation method to make GT-Paste applicable to both point clouds and images. Specifically, we first follow an observer’s perspective to filter occluded LiDAR points according to the geometrical unanimity rule. We then take hold of all the objects in current scene and paste their corresponding patches onto images in a far-to-near order. With the help of the cross-modal data augmentation, our proposed 3D detector achieves competitive results over the state-of-the-art methods.

In brief, our contributions are summarized as follow.

- We explore effective CNN features from 2D object detection networks as image representation to fuse with LiDAR points for 3D object detection.
- We propose a simple yet effective cross-modal data augmentation method for training 3D object detectors, considering the modality consistency between cameras and LiDAR.
- We extensively validate the effectiveness of cross-modal fusion and data augmentation on large-scale nuScenes and Waymo datasets. The proposed 3D detector PointAugmenting achieves the new state-of-the-art results on the nuScenes leaderboard to date.

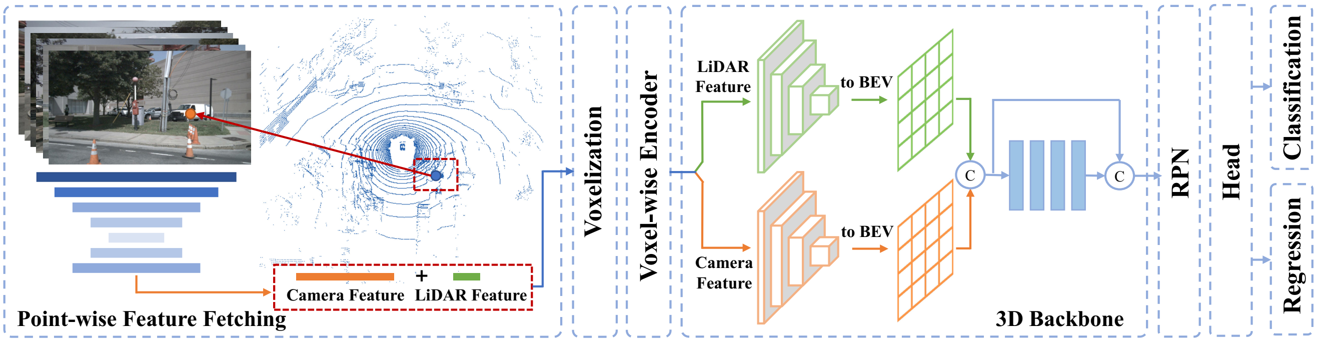

Figure 3. PointAugmenting overview. The architecture consists of two stages. (1) Point-wise feature fetching: LiDAR points are projected onto image plane and then appended by the fetched point-wise CNN features. (2) 3D detection: we extend CenterPoint with an additional 3D sparse convolution stream for camera features and fuse different modalities via a simple skip and concatenation approach in BEV maps.

2. Related Work

LiDAR-Based 3D Detection. Existing LiDAR-based methods can be broadly grouped into two categories: i.e., grid-based and point-based. The grid-based methods divide point clouds into regular 3D voxels [22, 34, 27, 6] or BEV maps [9, 24, 23]. In SECOND [22], a sparse convolution operation is proposed to parse the compact representation. CenterPoint [27] replaces the general anchor-based detector with a keypoint-based detector. For point-based approaches, PointRCNN [15] and STD [26] apply PointNet [14] to segment foreground points and generate proposals for each point. 3DSSD [25], a single-stage detector, disposes all upsampling layers and refinement modules for computational efficiency. Compared to grid-based methods, point-based approaches require high computation loads, resulting in time-consuming training on large-scale datasets such as the nuScenes [2] and Waymo [18] datasets.

Fusion-Based 3D Detection. Recently, multi-sensor fusion has shown great advantages. F-PointNet [13] generates 3D bounding box based on 2D detection results. AVOD [8] and MV3D [3] perform fusion on object proposals via ROI pooling. Researchers [11, 10, 29] have attempted to transform the front-view camera features into BEV maps. ContFuse [11] introduces a novel continuous fusion layer while 3D-CVF [29] employs auto-calibrated projection to construct a smooth BEV feature map. Despite the promising results, there exists the problem of feature blurring. Instead, other methods [7, 16, 19] explore the fusion mechanism in a point-wise manner. MVX-Net [16] and PointPainting [19] respectively fetch feature maps and segmentation scores from camera images and apply simple concatenation on both points. EPNet [7] designs a LI-Fusion module to establish a finer point-wise correspondence. In our work, we explore a better image representation and fusion mechanism to facilitate point-wise cross-modal data fusion.

Data Augmentation. Data augmentation of point clouds is crucial in 3D object detection. The original GT-Paste augmentation pastes virtual objects into current training scenes. This operation not only accelerates network convergence but also alleviates the annoying class imbalance problem. However, it is not adaptive to cross-modal data. For 2D data augmentation, patch-based methods [17, 31, 28] that dropout or paste patches in training images encourage a more robust network learning. Cutmix [31] overlays a region with a patch cut from another image and [1] adapts it for 2D detection tasks. Inspired by Cutmix, our intention for cross-modal 3D augmentation is to simultaneously paste object points and image patches into scenes while maintaining the consistency between sensors.

3. PointAugmenting

This section presents the proposed method PointAugmenting for 3D object detection. We adopt CenterPoint as our LiDAR-only baseline and extend it with a cross-modal fusion mechanism as well as an effective data augmentation scheme. Figure 3 illustrates our cross-modal network architecture, which consists of two stages: (1) Point-wise feature fetching. LiDAR points are projected onto image plane and then appended by the fetched point-wise CNN features. (2) 3D detection. We extend CenterPoint with an additional 3D sparse convolution stream for camera features and fuse features of different modalities in BEV maps. To facilitate network training, we further employ a novel data augmentation scheme for our cross-modal detector. The details of PointAugmenting are presented in the following.

3.1. Cross-Modal Fusion

For implementation efficiency, we construct our method based on CenterPoint [27], which is a one-stage and anchor-free LiDAR-only 3D detector. LiDAR points in CenterPoint are first fed to a 3D encoder cascaded by voxelization, voxel-wise feature encoder, and 3D backbone, yielding flattened compact 2D BEV feature maps. Finally, a 2D CNN broadcasts the features to multi-heads for multi-prediction:

object centers, 3D size, and orientation.

Point-wise Feature Fetching. The success of the recent detector PointPainting [19] inspires us to decorate LiDAR points with semantics from camera images. While our preliminary results in Figure 1 show that high dimensional CNN features perform better than segmentation scores. As such, we choose CNN features of images for point decoration. To extract the point-wise image features, we use an off-the-shelf network trained for 2D object detection rather than semantic segmentation. The reasons lie in three aspects. First, 2D and 3D object detection are complementary tasks that focus on different levels of granularity of objects. They benefit from each other. Second, 2D detection labels are readily available from 3D projection, whereas segmentation labels are expensive and usually unavailable. Last, the detection network is more friendly to data augmentation than the segmentation network, as suggested by [1]. To be specific, we take the output activation from DLA34 [30] of CenterNet [33, 32] as image features, where the channel number of the feature map is 64 and its scale factor is 4. To fetch the corresponding point-wise image features, we project LiDAR points onto the image plane by a homogeneous transformation to set up the correspondence. Afterwards, LiDAR points are appended by the fetched point-wise image features as network inputs to perform detection.

3D Detection. Each LiDAR point is defined by (x, y, z, r, t) and (x, y, z, r) respectively on the nuScenes and Waymo datasets, where x, y, z are location coordinates, r denotes the reflectance, and t is the relative timestamp. We set f_i as the 64D image features. The fused LiDAR points can be denoted by $(x, y, z, r, (t), f_i)$. Considering the modality gap and different data characteristics between LiDAR and cameras, unlike point-wise concatenation used by PointPainting, we employ a late fusion mechanism across modalities. As shown in Figure 3, after the voxel-wise feature encoder, we use two separate 3D sparse convolution branches to process the LiDAR and camera features. Afterwards, we flatten the two downsampled 3D feature volumes into 2D BEV maps, each with the channel number of 256. These two BEV maps are then concatenated in channel-wise and then fed into four 2D convolution blocks for feature aggregation. Each convolutional block consists of two 3×3 convolution layers followed by a batch normalization layer and a ReLU activation function. The first block shrinks the channel number from 512 to 256. Finally, we add a skip connection between the aggregated features and the previous camera and LiDAR BEV features before forwarding to the region proposal network.

3.2. Cross-Modal Data Augmentation

We propose an efficient data augmentation scheme to make GT-Paste applicable during training our cross-modal detector. Inspired by the recent image augmentation approach Cutmix [31], we attempt to simultaneously attach an image patch to images when pasting LiDAR points of a virtual object into current 3D scene. The main challenge lies in the preservation of the consistency between camera and LiDAR data. As shown in Figure 4, from the observer's perspective, the pasted bicycle is partly occluded by the car in the original 3D scene, resulting in an overlap on camera image. If we directly paste the virtual object patch onto images, the points of objects projected in the overlap region may fetch mismatching features. Furthermore, the background points projected into the virtual patch also capture incorrect information. To address this issue, we identify the occlusion relationships between foreground objects and filter those occluded LiDAR points from the observer's perspective. For camera images, we take out both virtual and original objects and attach their patches by a far-to-near order.

Augmentation for LiDAR Points. We transform the LiDAR point (x, y, z) into the LiDAR spherical coordinate system as (r, θ, ϕ) and represent the perspective of an object using the range of θ and ϕ, where the minimum and maximum of θ and ϕ are obtained from the eight corners of its ground-truth box. When selecting virtual objects, the original GT-Paste requires to avoid the collision of objects. Our method also restricts the perspective overlap between objects so as not to filter over too much foreground points. Then the selected virtual objects are pasted into current scene and we filter occluded points from the perspective of an observer. Specifically, given both original and pasted virtual objects in current scene, we process each object in a near-to-far order. If an original object is taken, we only discard those occluded points that belong to farther pasted objects. If a pasted object is processed, all occluded points farther than this object will be disposed. Moreover, we filter the background points in the perspective of this virtual object. It is because original objects occlude only the far virtual objects, whereas the pasted virtual objects occlude all the far objects as well as background points. The detailed procedure of our occlusion-aware point filtering is illustrated in Algorithm 1.

Augmentation for Camera Images. To match the consistency between LiDAR and cameras, for each virtual object that pasted into LiDAR scenes, we attach its corresponding patch within a 2D bounding box onto images. The 2D bounding box is obtained from 3D ground-truth projection. To determine the pasted position, we note that although virtual points are pasted at their original locations in LiDAR scenes, virtual patches are not exactly at the original position of camera planes due to the jitter of camera external parameters. We need to re-compute the position of 2D bounding box through the current camera external calibration and then transform the original patch by translation and scaling.

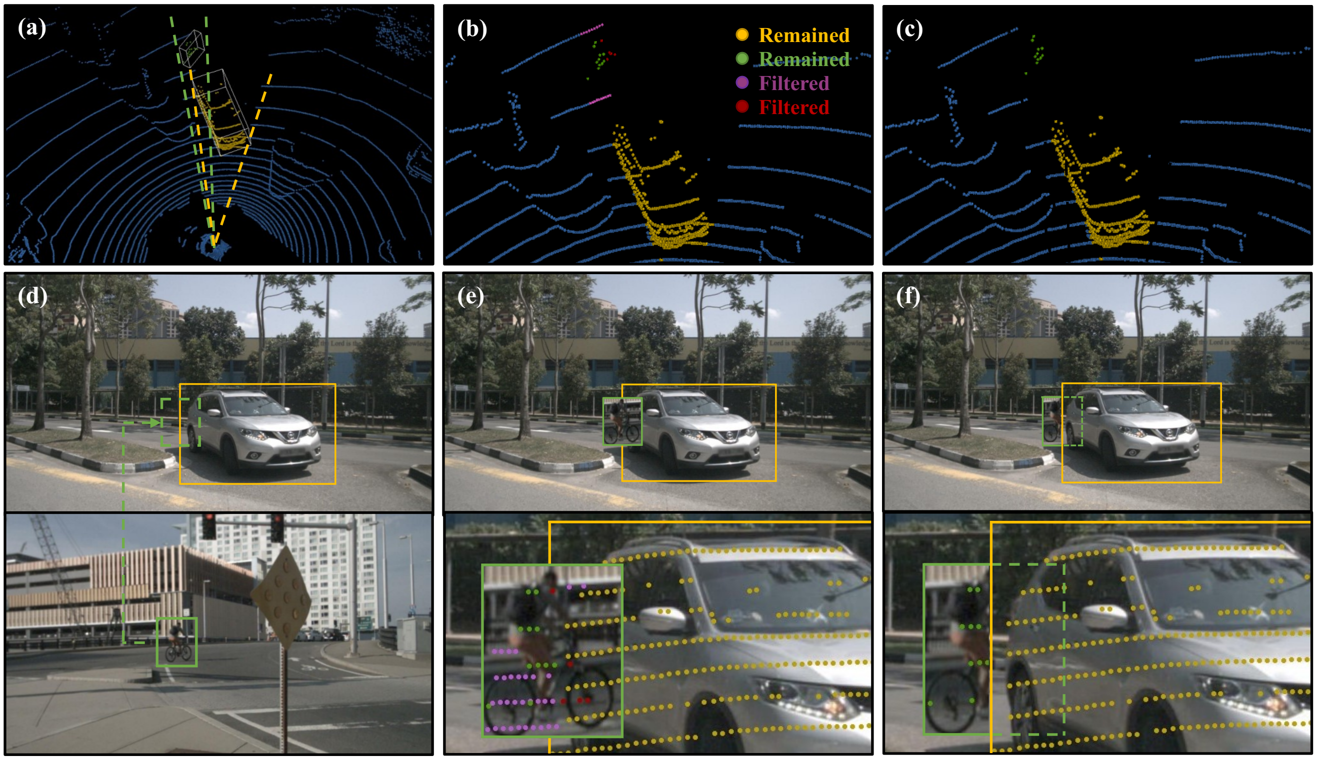

Figure 4. Example of cross-modal data augmentation. In (d), a pasted bicycle (green) is partly occluded by the car (yellow) in the original scene, whose LiDAR points are shown in (a). Directly attaching the virtual patch (green) onto the image yields mismatch between points and camera pixels (b and e): (i) the LiDAR points of both objects are partly projected into the overlap region, (ii) a few background points (purple) fetch the features in the virtual patch. To avoid such ambiguities, we filter the foreground points (red) occluded by the front objects and the background points (purple) that in the perspective of virtual objects. For images, we crop image patches of both virtual and original objects, then paste them onto images in a far-to-near order. This observes the consistency between LiDAR points (c) and images (f).

Algorithm 1: Occlusion-aware point filtering

```
Input: Objects $\mathbf{O}$, Object perspectives $\mathbf{V}$, Object
       depths $\mathbf{D}$, Points $\mathbf{P}$, Point depths $\mathbf{R}$
ObjectInds $\leftarrow$ $AscendingSort(\mathbf{D})$ ;
for $i$ in ObjectInds do
    if $O_i$ is pasted then
        FGInds $\leftarrow$ ($\mathbf{P} \in V_i$) and ($\mathbf{P} \in FG$) and
          ($\mathbf{R} \geq D_i$);
        BGInds $\leftarrow$ ($\mathbf{P} \in V_i$) and ($\mathbf{P} \in BG$);
        FilterInds $\leftarrow$ FGInds $\cup$ BGInds;
    else
        FilterInds $\leftarrow$ ($\mathbf{P}$ = pasted) and ($\mathbf{P} \in V_i$) and
          ($\mathbf{R} \geq D_i$);
    $\mathbf{P} \leftarrow \mathbf{P} - \mathbf{P}$(FilterInds) ;
Output: $\mathbf{P}$
```

Moreover, rather than directly pasting the virtual patches, we take hold of the patches of both virtual and original objects, and paste them in a far-to-near order. In this way, background objects are occluded by foreground objects in images, in accordance with the occlusion relationship between objects in LiDAR scenes.

Fade Strategy. Despite the large performance gains, data augmentation violates the real data distribution, especially for our data across LiDAR points and camera images. To this end, we disable data augmentation when the model is near convergent, leading our model to learn from real scenes. This strategy further yields an improvement of $+1.3\%$ mAP on the 1/8 nuScenes dataset (see Table 5).

4. Experiments

We evaluate the proposed PointAugmenting 3D detector on both the nuScenes and Waymo Open datasets, and conduct extensive ablation studies to verify our design choices.

4.1. Experimental Setup

Through experiments, we use the adamW [12] optimizer with the one-cycle policy [5], with max learning rate $1e-3$ and $3e-3$ for nuScenes and Waymo, weight decay 0.01, and momentum ranges from 0.85 to 0.95. During training, we

Method	mAP	NDS	Car	Truck	C.V.	Bus	Trailer	Barrier	Motor.	Bicycle	Ped.	T.C.
PointPillars [9]	30.5	45.3	68.4	23.0	4.1	28.2	23.4	38.9	27.4	1.1	59.7	30.8
3DSSD [25]	42.6	56.4	81.2	47.2	12.6	61.4	30.5	47.9	36.0	8.6	70.2	31.1
PointPainting [19]	46.4	58.1	77.9	35.8	15.8	36.2	37.3	60.2	41.5	24.1	73.3	62.4
CBGS [35]	52.8	63.3	81.1	48.5	10.5	54.9	42.9	65.7	51.5	22.3	80.1	70.9
CenterPoint [27]	60.3	67.3	85.2	53.5	20.0	63.6	56.0	71.1	59.5	30.7	84.6	78.4
Ours	**66.8**	**71.0**	**87.5**	**57.3**	**28.0**	**65.2**	**60.7**	**72.6**	**74.3**	**50.9**	**87.9**	**83.6**

Table 2. Performance comparisons on the nuScenes test set. We report the NDS, mAP, and mAP for each class.

Method	Vehicle		Pedestrian		Cyclist		All	
	L1 mAP	L2 mAP	L1 mAP	L2 mAP	L1 mAP	L2 mAP	L1 mAP/mAPH	L2 mAP/mAPH
CenterPoint [27]	66.70	62.00	73.55	68.64	72.51	70.00	70.92 / 68.26	66.88 / 64.36
Ours	67.41	62.70	75.42	70.55	76.29	74.41	73.04 / 70.39	69.22 / 66.70
Gains of fusion	+0.71	+0.70	+1.87	+1.91	+3.78	+4.41	+2.12 / +2.13	+2.34 / +2.34

Table 3. Performance comparisons on the Waymo validation set. The results of CenterPoint are reproduced by ourselves.

conduct data augmentation of random flipping along both X and Y axis, global scaling, global rotation and random global translation. We also apply our proposed cross-modal data augmentation to paste virtual objects into both LiDAR scenes and camera images. Models are trained with batch size 16 for 20 epochs on 8 V100 GPUs. At inference, we keep the top 1000 predictions in each group, then use the non-maximum-suppression (NMS) with IoU threshold 0.2 and score threshold 0.1. Following CenterPoint, we adopt the double-flip testing.

4.2. nuScenes Results

The nuScenes dataset [2] is a large-scale dataset for 3D detection, which consists of 700 scenes for training, 150 scenes for validation, and 150 scenes for test. The dataset is collected using six cameras and a 32-beam LiDAR, and 3D annotations for 10 objects in 360 degree field of view are provided. We set the detection range to within $[-54m, 54m]$ for X and Y axis, and $[-5m, 3m]$ for the Z axis, which is voxelized with $(0.075m, 0.075m, 0.2m)$ voxel size. We use 10 sweeps for LiDAR enhancement and limit the max number of non-empty voxels to 90000. We follow the official evaluation protocol [2] to report results.

We submitted our detection results to the nuScenes test server for evaluation. In Table 2, our PointAugmenting performs well over previous state-of-the-art methods on the official leaderboard by remarkable margins. Compared to CenterPoint, our approach obtains significant gains of $+6.5\%$ mAP and $+3.7\%$ NDS with consistent improvements over all the classes. In more detail, bicycle receives the largest increase with $+20.2\%$ mAP. This is because bicycles often have few LiDAR points and confusing geometry, thus the additional semantic cues serve as a valuable guidance for 3D detectors. Moreover, notable gains are achieved on both small classes ($+5.2\%$ mAP for traffic cone) and tail classes ($+8.0\%$ mAP for construction vehicle), which manifests the effectiveness of leveraging camera to assist LiDAR to deal with hard examples.

4.3. Waymo Results

Waymo Open Dataset [18] is currently the largest dataset for autonomous driving. There are totally 798 scenes for training and 202 scenes for validation, which is collected by five LiDAR sensors and five pinhole cameras and annotated with 2D and 3D labels. During training, we set detection range to $[-76.8m, 76.8m]$ for X and Y axis, and $[-5m, 3m]$ for the Z axis with a voxel size of $(0.075m, 0.075m, 0.1m)$. Max number of non-empty voxels is set as 120000. Note that cameras in Waymo only cover around 250 degree field, which is different from LiDAR points and 3D labels in full 360 degree field. Thus, nearly 1/3 LiDAR points fails to fetch their corresponding image features due to the lack of camera in back perspective. As such, we only select LiDAR points and ground-truth in the camera FOVs for training our cross-modal detector. At the inference stage, to retrieve the 3D detection predictions in the whole scene, we complete the left 1/3 scene with predictions from CenterPoint.

Table 3 compares our detection results with CenterPoint. Although our cross-modal detector utilizes less training data than CenterPoint, it still performs superbly over all the object classes and two difficulty levels. In particular, we achieve remarkable gains on pedestrian and cyclist respectively with $+1.91\%$ and $+4.41\%$ mAP on LEVEL_2, which manifests the outstanding performance of our method for the objects with fewer than 5 LiDAR points. In terms of mAPH, we also yield superior performance, indicating a more accurate heading prediction on objects. The results on the Waymo dataset further validate both the effectiveness and generalization of our proposed PointAugmenting.

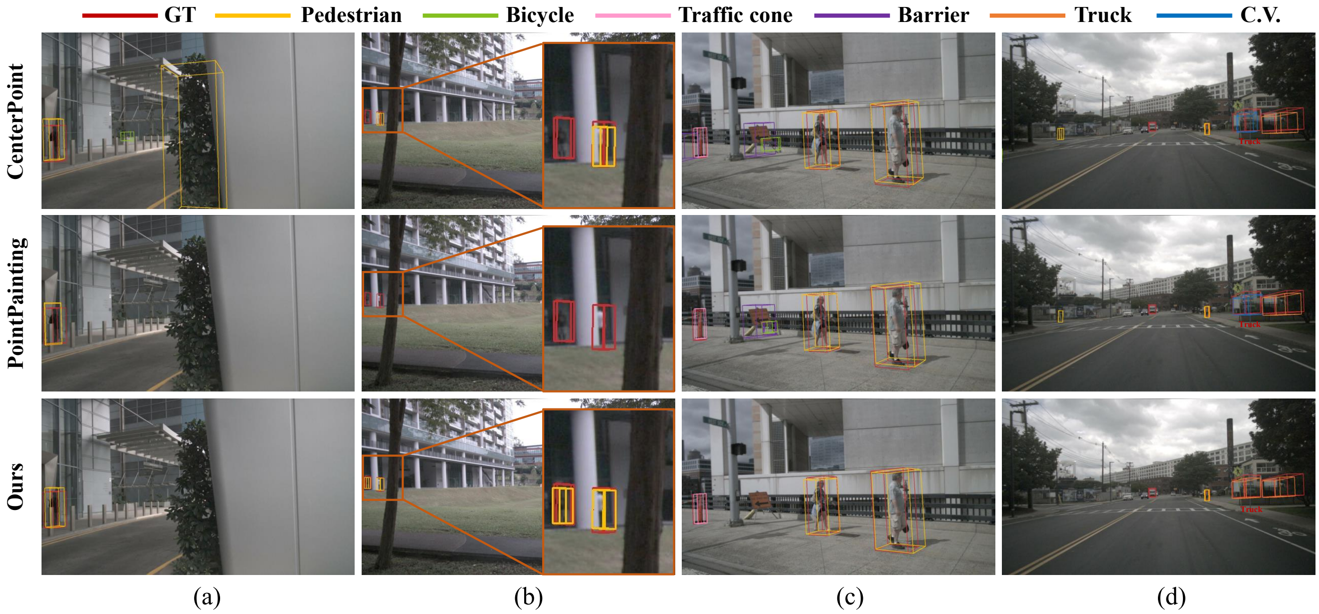

Figure 5. Qualitative comparison of detection Results. We compare our method with CenterPoint [27] and PointPainting [19]. We improve PointPainting with our fusion mechanism for fair comparison. (a) indicates the necessity of leveraging camera information, where CenterPoint falsely detects a human-like tree due to the loss of semantics. In (b), PointAugmenting successfully detects two far-away pedestrians while the other two both fail. In (c), a sign is falsely detected as a barrier or a bicycle by CenterPoint and PointPainting owing to the confusing geometry. In (d), CenterPoint and PointPainting mistake a truck as a construction vehicle while our PointAugmenting succeeds.

	Seg Score	DetFeat.	CC	LF	mAP	NDS
(a)					37.4	49.9
(b)	✓		✓		42.3	51.4
(c)		✓	✓		46.0	53.9
(d)		✓		✓	47.5	55.6

Table 4. Comparison of fusion policies. Seg Score: decorating LiDAR points with segmentation scores as suggested by PointPainting [19]. DetFeat: decorating LiDAR points with image features from the detection task. CC: fusing LiDAR and image features by point-wise concatenation. LF: our late fusion mechanism.

4.4. Ablation Studies

We conduct ablation studies on the nuScenes dataset to pinpoint the improvements. For efficiency, we use the 1/8 training data for training and test on the whole validation set. We train the models for 20 epochs with voxel size of $(0.1m, 0.1m, 0.2m)$ throughout ablation studies.

Fusion Architecture. We compare different fusion policies in Table 4. All studies here are trained under our cross-modal data augmentation but without the fade strategy. For image segmentation, we adopt HRNet-W48 [20] pretrained on Cityscapes [4]. We conclude the observation as below:

(1) Benefits of cross-modal fusion (a,d): Our fusion architecture dramatically boosts the LiDAR-only performance by $+10.1\%$ mAP, which indicates the significance of cross-modal fusion for 3D object detection.

	Naive	CM	Fade	Fusion	mAP	NDS
(e)					32.8	42.3
(f)	✓				37.6	49.5
(g)		✓			37.4	49.9
(h)				✓	42.6	50.0
(i)		✓		✓	47.5	55.6
(j)		✓	✓	✓	48.8	56.8

Table 5. Effectiveness of cross-modal data augmentation. Naive: the original GT-Paste applied to CenterPoint. CM: Our cross-modal GT-Paste data augmentation. Fade: the training strategy that discontinues our data augmentation in the last 5 epochs. Fusion: adding camera stream by our late fusion mechanism.

(2) Camera input for fusion (b,c): Replacing the segmentation scores suggested by PointPainting with our detection features yields an improvement of $+3.7\%$ mAP. Although segmentation scores offer a compact representation to complement LiDAR points, CNN features are better at providing rich appearance cues and large receptive fields. The results manifest the importance of choosing effective representation for camera modality.

(3) Different fusion mechanism (c,d): Comparing our late fusion mechanism to simple concatenation, we achieve a gain of $+1.5\%$ mAP by using our detection features as input. Early point-wise concatenation ignores the huge difference in data characteristics between LiDAR and camera

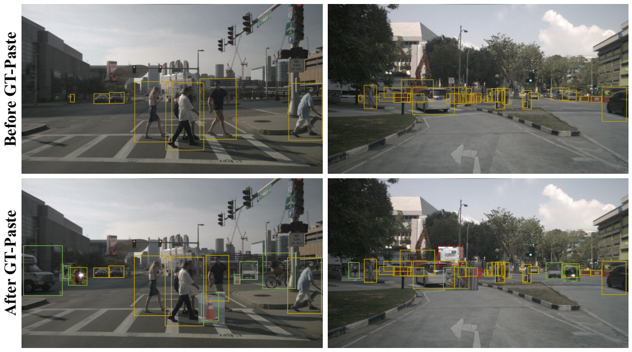

Figure 6. Qualitative results of 2D detection after our data augmentation. Top row: 2D detection results of original scenes. Bottom row: the results after our GT-Paste data augmentation. Yellow and green boxes respectively denote the detection results of original and pasted objects. Red boxes are the false negative predictions.

features whereas ours narrow down the modality gap by fusion at BEV. Although late fusion scheme performs the best, it brings additional computation cost due to the separate 3D sparse convolution stream. Therefore, an effective and efficient fusion mechanism is desirable in the future.

We present qualitative results in Figure 5 to compare three detectors, i.e, CenterPoint, PointPainting and PointAugmenting. We train three models on the whole training data of nuScenes. Figure 5 illustrates the superiority of both cross-modal fusion as well as our fusion policies. By leveraging rich camera information, our PointAugmenting significantly outperforms the other two methods in mitigating false predictions.

Cross-Modal Data Augmentation. We validate the effectiveness of our cross-modal data augmentation scheme in Table 5. Key observations are summarized as below:

(1) GT-Paste for LiDAR-only input (e,f,g): Applying GT-Paste to LiDAR points yields a boost of $+4.8\%$ mAP. This motivates us to investigate cross-modal data augmentation. Comparing (g) to (f), replacing the naive GT-Paste with ours yields a -0.2% mAP drop, which suggests that our operation on LiDAR points does not hurt LiDAR-only detection.

(2) GT-Paste for cross-modal input (h,i): When we leverage camera features to assist LiDAR detection, removing our cross-modal data augmentation for GT-Paste leads to an overall performance drop of -3.7% mAP. This discrepancy indicates the effectiveness of our strategy. Moreover, our scheme is applicable to other cross-modal detectors.

(3) Fade strategy (i,j): We apply the fade strategy during the last 5 training epochs. This further achieves an improvement of $+1.3\%$ mAP. Although our data augmentation scheme remarkably benefits the detector, it disturbs the real data distribution. The fade strategy is therefore helpful to learn from real scenes.

Methods	Image size	Fusion	mAP	2D time	Total time
CenterPoint	-	-	37.6	-	85ms
Ours	896 × 1600	LF	47.5	383ms	542ms
Ours lite 1	448 × 800	LF	47.3	95ms	238ms
Ours lite 2	448 × 800	CC	46.4	95ms	178ms

Table 6. Runtime per frame on the nuScenes dataset. CC: point-wise concatenation. LF: our late fusion mechanism. The runtime is on a NVIDIA 1080Ti GPU.

Visualization of 2D Detection. To verify the influence of GT-Paste on the 2D image backbone, we visualize 2D detection results by forwarding camera images to the 2D detection network we used for camera feature extraction, i.e., CenterNet with DLA34 backbone. Figure 6 shows the 2D detection results after data augmentation, where most of the objects can still be successfully detected. This implicitly suggests the effectiveness of image features for LiDAR point decoration under our patch-pasting operation on images. Nevertheless, false negative (red box) emerges after the augmentation on images. In Figure 6, the pasted bus is lost due to the occlusion by other objects while the traffic cone is missed caused by the similar color with the background in left virtual patch. This phenomenon is one of the reasons that drives us to adopt the fade strategy.

Runtime. We report the runtime per frame in Table 6. Compared with the lidar-only detector CenterPoint, our PointAugmenting takes extra running time due to the 2D image feature backbone (383 ms with image size of 896 × 1600) and the following 3D branch (60 ms) to generate camera features in BEV space. Running time is of great importance for autonomous driving. We find that reducing the input image size or replacing our late fusion with simple point-wise concatenation can largely accelerate our method. Table 6 shows that our two lite versions achieve much faster speed with slight drop in detection accuracy.

5. Conclusion

In this paper, we have presented a novel cross-modal 3D object detector, named PointAugmenting. With the proposed cross-modal data fusion and data augmentation scheme, PointAugmenting sets the new state-of-the-art results on the nuScenes detection leaderboard. Served as a strong baseline for cross-modal 3D detector, our PointAugmenting can be improved in two aspects in the future work. First, despite the effectiveness of our late fusion mechanism, a more efficient cross-modal fusion scheme is desirable. Moreover, considering the different fields of view between LiDAR and cameras in the Waymo dataset, a single model that adapts to different modalities, either LiDAR-only or cross-modality, is required for real applications.

Acknowledgements. This work was supported by NSFC (61906119, U19B2035), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and Shanghai Pujiang Program.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. 3, 4

[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 multi-modal dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11621–11631, 2020. 2, 3, 6

[3] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1907–1915, 2017. 1, 3

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 7

[5] Sylvain Gugger. The 1cycle policy, 2018. 5

[6] Chenhang He, Hui Zeng, Jianqiang Huang, Xian-Sheng Hua, and Lei Zhang. Structure aware single-stage 3d object detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11873–11882, 2020. 3

[7] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 35–52, 2020. 1, 3

[8] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8, 2018. 1, 3

[9] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. 3, 6

[10] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7345–7353, 2019. 1, 3

[11] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 1, 3

[12] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[13] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 918–927, 2018. 1, 3

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

[15] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 3

[16] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. Mvx-net: Multimodal voxelnet for 3d object detection. In *International Conference on Robotics and Automation (ICRA)*, pages 7276–7282, 2019. 1, 2, 3

[17] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[18] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2446–2454, 2020. 3, 6

[19] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4604–4612, 2020. 1, 3, 4, 6, 7

[20] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1, 2, 7

[21] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *arXiv preprint arXiv:1903.01864*, 2019. 1

[22] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 2, 3

[23] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155, 2018. 3

[24] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7652–7660, 2018. 3

[25] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11040–11048, 2020. 3, 6

[26] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1951–1960, 2019. 3

[27] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Center-based 3d object detection and tracking. *arXiv preprint arXiv:2006.11275*, 2020. 1, 3, 6, 7

[28] Jaejun Yoo, Namhyuk Ahn, and Kyung-Ah Sohn. Rethinking data augmentation for image super-resolution: A comprehensive analysis and a new strategy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8375–8384, 2020. 3

[29] Jin Hyeok Yoo, Yeocheol Kim, Ji Song Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. *arXiv preprint arXiv:2007.08856*, 2020. 1, 3

[30] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2018. 4

[31] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision (CVPR)*, pages 6023–6032, 2019. 3, 4

[32] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *In Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 2, 4

[33] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 2, 4

[34] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4490–4499, 2018. 3

[35] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6