MonoLSS: Learnable Sample Selection For Monocular 3D Detection

Zhenjia Li[*]
Baidu Inc.
Beijing, China
lizhenjia@baidu.com

Jinrang Jia[*]
Baidu Inc.
Beijing, China
jiajinrang@baidu.com

Yifeng Shi[†]
Baidu Inc.
Beijing, China
shiyifeng@baidu.com

Abstract

*In the field of autonomous driving, monocular 3D detection is a critical task which estimates 3D properties (depth, dimension, and orientation) of objects in a single RGB image. Previous works have used features in a heuristic way to learn 3D properties, without considering that inappropriate features could have adverse effects. In this paper, sample selection is introduced that only suitable samples should be trained to regress the 3D properties. To select samples adaptively, we propose a Learnable Sample Selection (LSS) module, which is based on Gumbel-Softmax and a relative-distance sample divider. The LSS module works under a warm-up strategy leading to an improvement in training stability. Additionally, since the LSS module dedicated to 3D property sample selection relies on object-level features, we further develop a data augmentation method named MixUp3D to enrich 3D property samples which conforms to imaging principles without introducing ambiguity. As two orthogonal methods, the LSS module and MixUp3D can be utilized independently or in conjunction. Leveraging the LSS module and the MixUp3D, without any extra data, our method named MonoLSS ranks **1st** in all three categories (Car, Cyclist, and Pedestrian) on KITTI 3D object detection benchmark, and achieves competitive results on both the Waymo dataset and KITTI-nuScenes cross-dataset evaluation. The code is available at https://github.com/Traffic-X/MonoLSS.*

1. Introduction

3D object detection has received increasing attention in the fields of autonomous driving [16, 26, 28], intelligent traffic [15, 55, 56], and robot navigation. Compared to expensive LIDAR sensor [6, 21, 49, 51], monocular camera which enables a greater range of environmental perception and object localization is economical. In recent years, many visual monocular 3D detection algorithms [14, 24, 31, 40, 50, 52] have been proposed to improve perception effect. Different from 2D object detection, accurate 3D property predictions (depth, dimension, and orientation), especially depth [29, 36], are more critical and challenging in this field.

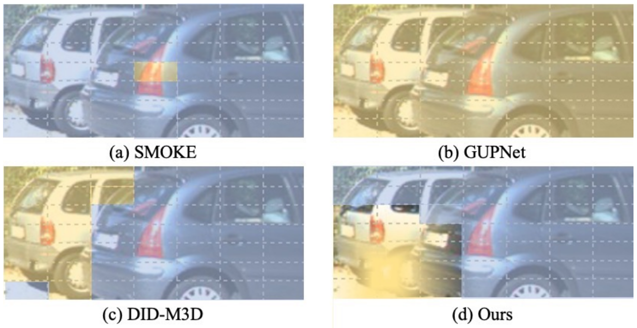

Figure 1. **Comparison among various features using by different methods for 3D property learning.** When the occluded white vehicle is the target of detection, various methods utilize distinct features for learning. Yellow color means features used and blue means not.

To achieve accurate 3D property estimations, many methods add 3D property prediction branches into 2D detectors [43, 61]. These branches leverage the features extracted by backbone networks to output 3D properties. However, it should be noted that not all of the features are appropriate for learning 3D properties. The design motivation comes from the label assignment of 2D detection. People rarely require an achor with IOU less than 0.3 as a positive sample for target detection (in anchor free method, it means far from object center). This is because the visual features do not match the learning objectives. The use of inappropriate ones can result in ambiguity and even have adverse effects. We transfer this knowledge to 3D property learning. For example, as shown in Figure 1, the white car which has a feature map with size $d * d * C$ is occluded by a gray one. SMOKE [32] only uses one fixed-position feature with size $1 * 1 * C$ located on the 3D center of object to regress 3D properties. When occlusion occurs, this feature may lie on another object. Although the receptive field is

[*] Equal contribution.
[†] Corresponding author.

not limited to the location of the feature, the network may not receive the optimal information as input. In contrast, GUPNet [34] takes advantage of all the $d * d$ features and outputs 3D properties with a global average pooling module [30]. Suffering from useless information including foreground and background interference, this approach remains problematic.

In this work, we introduce sample selection to identify the features that are beneficial for learning 3D properties and serve as positive samples, while disregarding the rest and treating them as negative samples. The challenge lies in how to divide them. An intuitive approach is to focus on the features of the target objects themselves (Figure 1 (c)), but these methods require the introduction of additional data such as depth maps [40] or segmentation labels, and still cannot select suitable samples among different internal components of objects, such as wheels, lights, or bodies. To address the 3D property sample selection problem, we propose a novel Learnable Sample Selection (LSS) module. The LSS module implements probability sampling with Gumbel-Softmax [13]. Furthermore, top-k Gumbel-Softmax [22] is employed to enable multi-sample sampling, expanding the number of samples drawn from 1 to k. Moreover, to replace the use of a same k value for all objects, we developed a hyperparameter-free sample divider based on relative distance, which achieves adaptive determination of sampling values for each object. Additionally, inspired by HTL method [34], the LSS module works with a warm-up strategy to stabilize training process.

Furthermore, the LSS module dedicated to 3D property sample selection relies on object-level features. However, the object number in training data is always limited. Meanwhile, most 3D monocular data augmentation methods, such as random crop-expand, random flip, copy and paste, etc., do not change the features of objects themselves. Some of them even introduce ambiguous features due to the violation of imaging principles. In order to improve the richness of the 3D property samples, we propose MixUp3D, which adds physical constraints on the basis of traditional 2D MixUp [57] to simulate spatial overlap in the physical world. The spatial overlap does not change 3D properties of the objects, such as a car overlapping a bicycle, but we can still judge their depths, dimensions, and orientations. As a simulation of the spatial overlap, the MixUp3D enables the objects to conform to imaging principles without introducing ambiguity. It can enrich training samples and alleviate overfitting. Moreover, the MixUp3D can be used as a fundamental data augmentation method in any monocular 3D detection approach.

Incorporating all the techniques, our monocular 3D detection method named MonoLSS outperforms prior state-of-the-art (SOTA) works with a significant margin without using any extra data. It can be trained end-to-end simply while still maintaining real-time efficiency. To summarize, the main contributions of this work are as follows:

- We emphasize that not all features are equally effective for learning 3D properties, and first reformulate it as a problem of sample selection. Correspondingly, a novel Learnable Sample Selection (LSS) module that can adaptively select samples is developed.
- To enrich 3D property samples, we devise MixUp3D data augmentation, which simulates spatial overlap and improves 3D detection performance.
- Without introducing any extra information, MonoLSS ranks 1st in all the three classes in the KITTI benchmark [9] and surpasses the current best method by more than 11.73% and 12.19% relatively on the Moderate and Hard levels of the Car class. It also achieves SOTA results on Waymo dataset [47] and KITTI-nuScenes [2] cross-dataset evaluation.

2. Related work

Monocular 3D Object Detection. The monocular 3D object detection aims to predict accurate 3D bounding boxes. According to whether use extra data, monocular 3D object detection algorithms can be mainly categorized into two groups. The first kind of method only uses a single image as input without any extra information. For example, M3D-RPN [1] adopts a standalone 3D region proposal network and proposes a depth-wise convolution to predict objects. Based on a CenterNet-style [61] network, SMOKE [32] predicts 3D bounding boxes by combining a single keypoint estimation module. Furthermore, MonoFlex [60] optimizes the truncated obstacles prediction method with an edge heatmap and edge fusion module. MonoPair [5] explores the relationships between different objects. MonoEF [62] first predicts camera extrinsic parameters by detecting vanishing point and horizon change, and then adopts a converter to rectify perturbative features in the latent space. MonoCon [50] learns auxiliary monocular contexts projected from the 3D bounding boxes in training and discards them for better inference efficiency in inference. MonoDDE [28] exploits depth clues in monocular images and develops a model which produces 20 depths for each target.

The second kind of method uses extra data, such as depth maps, LIDAR point clouds and CAD models, to obtain additional information, and enhance detection. ROI-10D [37] combines the deep feature maps and estimates dense depth maps to regress 3D bounding boxes. D4LCN [8] proposes depth-guided convolution in which the receptive field is determined adaptively by the predicted depth. DID-M3D [40] decouples the instance depths into attribute depths and visual depths by using a dense depth map. CaDDN [41] uses LIDAR points to generate depth maps and estimates depth by an additional monocular network, then converts the feature to BEV perspective for prediction. CMKD [11] de-

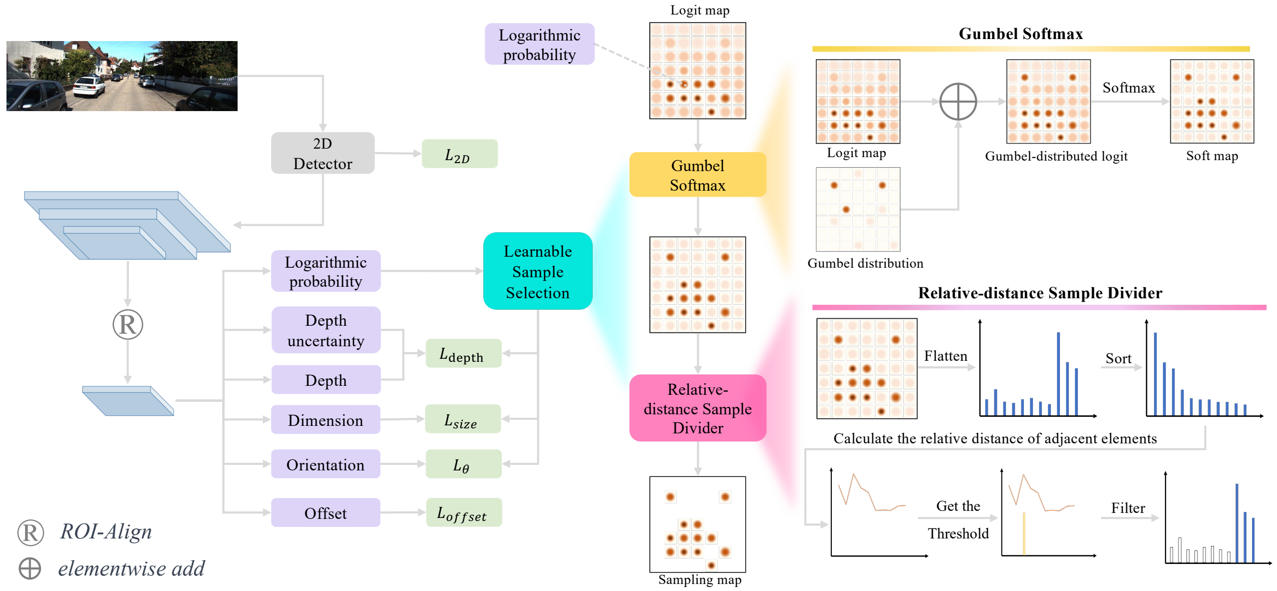

Figure 2. **An overview of the MonoLSS framework.** First, a 2D detector combined with ROI-Align is used to generate object features. Then, six heads respectively predict 3D properties (depth, dimension, orientation, and 3D center projection offset), depth uncertainty, and logarithmic probability. Finally, the Learnable Sample Selection (LSS) module adaptively selects samples and acts on the loss calculation.

velops a cross-modality method to transfer the knowledge from LiDAR modality to image. Besides depth maps and LIDAR, methods such as AutoShape [33] utilize CAD models to generate dense keypoints to alleviate the sparse constraints.

Methods that leverage additional data always exhibit superior performance due to the increased information. However, the complex sensor configuration and computational overhead limit their practical applications in industry.

Sample Selection in 2D/3D Detection. According to how to allocate samples, 2D object detection methods can be mainly categorized into two groups. The anchor-based methods [7, 42, 43] allocate positive samples based on the Intersection Over Union (IOU) between target boxes and pre-defined anchors, while the anchor-free methods [27] are based on certain rules. ATSS [59] allocates samples by setting adaptive IOU thresholds based on statistical characteristics of the object. MTL [17] finds the best sample points by gradually reducing the number of positive samples.

The sample assign strategy for 2D properties of 3D detection methods generally follows those mentioned above. Many methods [28, 32, 60] use these strategies consistent with 2D properties to learn 3D properties. Methods [34, 44] use object features extract from backbone by ROI-Align [43] to regress one 3D property, which leads the results suffer from the foreground and background interference. DID-M3D [40] uses dense depth maps to select positive samples, which requires extra annotations.

Data Augmentation in Monocular 3D Detection. Due to the violation of geometric constraints, random horizontal flipping [5, 28, 63] and photometric distortion [3, 50] are the only two data augmentation methods mostly used in monocular 3D detection. Some methods [40] use random crop and expand to simulate the proportional change of depth. However, according to the imaging principles, it is impractical for all depths on one image to have the same proportional change. Some methods [29, 53] use an additional depth map to simulate the forward and backward movement of the camera along the z-axis. While, due to parallax and depth map errors, this methods introduce a lot of noise and distorted appearance features. Instance-level copy-paste [29] is also used as a 3D data augmentation method, but limited by the complex manual processing logic, it is still not realistic enough.

3. Methodology

Monocular 3D object detection extracts features from a single RGB image, estimates the category and 3D bounding box for each object in the image. The 3D bounding box can be further divided into 3D center location (x, y, z), dimension (h, w, l) and orientation (yaw angle) θ. The roll and pitch angles of objects are set to 0.

In this work, we propose a novel Learnable Sample Selection (LSS) module to optimize the monocular 3D object detection process. The overall architecture of the MonoLSS is illustrated in Figure 2, which mainly includes 2D detector, ROI-Align, 3D detection heads, and LSS module.

Our 2D detector is built on CenterNet [61]. It takes an image $I \in \mathbb{R}^{H \times W \times 3}$ as input and adopts DLA34 [54] to compute the deep feature $F \in \mathbb{R}^{\frac{H}{4}\times\frac{W}{4}\times C}$, where C is the channel number. Then deep feature F is fed into three 2D detection heads to regress 2D offset, 2D size and 2D heatmap. We achieve 2D boxes by combining these 2D predictions and further use ROI-Align to extract object features $F_{obj} \in \mathbb{R}^{n\times d\times d\times C}$ from deep feature F, where $d \times d$ is the ROI-Align size and n refers to the number of ROIs.

Unlike other methods that only predict a single 3D bounding box based on object features, our method uses each sample point in object features to predict a 3D bounding box and a logarithmic probability. In addition, we follow the multi-bin design for predicting the orientation and predict the uncertainty for depth, which is the same as GUPNet [34]. Therefore, we have 3D box dimension $S_{3d} \in \mathbb{R}^{n\times d\times d\times 3}$, 3D center projection offset $O_{3d} \in \mathbb{R}^{n\times 2}$, orientation $\Theta \in \mathbb{R}^{n\times d\times d\times 12\times 2}$, depth $D \in \mathbb{R}^{n\times d\times d}$, depth uncertainty $U \in \mathbb{R}^{n\times d\times d}$, and logit map $\Phi \in \mathbb{R}^{n\times d\times d}$. Based on the logit map predicted by the network, the LSS module can adaptively select positive samples for 3D properties when training. During inference, the LSS module selects the best 3D properties according to the highest logarithmic probability in logit map.

3.1. Learnable Sample Selection

Assert $U \sim Uniform(0, 1)$, then we can use inverse transform sampling to generate the Gumbel distribution G by computing $G = -log(-log(U))$. By independently perturbing the log-probabilities with Gumbel distribution and using the $argmax$ function to find the largest element, the Gumbel-Max trick [10] achieves probability sampling without random choices. Based on this work, Gumbel-Softmax [13] uses the softmax function as the continuous, differentiable approximation to $argmax$, and achieves overall differentiability with the help of reparameterization. Gumbel-Top-k [22] extends the number of sample points from Top-1 to Top-k by drawing an ordered sampling of size k without replacement, where the k is a hyperparameter. However, a same k is not suitable for all objects, for instance, occluded objects should have fewer positive samples than normal ones. To this end, we design a hyperparameter-free relative-distance based module to divide samples adaptively. In summary, we propose a Learnable Sample Selection (LSS) module to address sample selection problem in 3D property learning, which is formed by Gumbel-Softmax and relative-distance sample divider. The diagram of the LSS module is shown on the right side of Figure 2.

Let $\Phi = \{\phi_1, \phi_2, ..., \phi_N\}$ be the logit map output by the model, where $N = d \times d$ denotes the number of sample points. Each element ϕ_i for $i \in [1, N]$ represents a logarithmic probability. Gumbel-Softmax is performed to the logit map Φ to achieve probability sampling. Concretely, we first generate the Gumbel distribution G with size $d \times d$ based on the previous description and add it to the logit map Φ to obtain the Gumbel-distributed logit $\hat{\Phi} = \{\hat{\phi}_1, \hat{\phi}_2, ..., \hat{\phi}_N\}$. In this way, the originally low logarithmic probability values have the opportunity to surpass other high logarithmic probability values. Then a temperature-scaled softmax is adopted to process $\hat{\Phi}$ to get soft map $S = \{S_1, S_2, ..., S_N\}$. The overall process can be formulated as follows:

$$\hat{\Phi} = (G + \Phi) \tag{1}$$

$$S_i = \frac{exp(\hat{\phi}_i/\tau))}{\sum_{j=1}^{N} exp(\hat{\phi}_j/\tau)} \; for \; i = 1, ..., N \tag{2}$$

where the temperature coefficient τ is set to 1 in this work.

After that, the relative-distance sample divider is adopted to replace the fixed k in Gumbel-Top-k to implement adaptive sample allocation. We use the maximum interval between the elements of the soft map to distinguish positive and negative samples. Generally, absolute distance $(Abs\_dis = |a - b|)$ is used to indicate the interval. However, due to the amplification effect of the softmax function, using the absolute distance to divide samples may lead to an insufficient number of positive samples. We employ the relative distance $(Rel\_dis = \frac{a}{b})$ to increase the number of positive samples[1]. For example, if the softmax function has an input vector [20, 18, 17, 7], the output would be [0.84, 0.12, 0.04, 0], using the absolute distance will assign only one positive sample while the relative distance assigns three.

First, we flatten the soft map S to a one-dimensional vector $Soft\_S$ and sort it to get a sorted vector $Sort\_S$. Second, we calculate the relative distance between adjacent elements of the vector $Sort\_S$ by the following formula:

$$\begin{aligned} Dis\_S_i &= \frac{Sort\_S_i}{Sort\_S_{i+1}} = \frac{exp(\hat{\phi}_{f(i)}/\tau)}{exp(\hat{\phi}_{f(i+1)}/\tau)} \\ &= exp(\frac{\hat{\phi}_{f(i)} - \hat{\phi}_{f(i+1)}}{\tau}) \; for \; i = 1, ..., N-1 \end{aligned} \tag{3}$$

where $f()$ denotes the mapping relation from $Sort\_S$ to $Soft\_S$. Assume the Dis_S_i is the maximum value in Dis_S, since the exponential function is a monotone increasing function, the $\hat{\phi}_{f(i)} - \hat{\phi}_{f(i+1)}$ is the maximum interval in $\hat{\Phi}$. In essence, we are finding the most discriminative value in Gumbel-distributed logit $\hat{\Phi}$ to distinguish between positive and negative samples. We choose the $Sort\_S_i$ corresponding to the Dis_S_i as the threshold to filter negative samples. Specifically, the values in the soft map S that smaller than $Sort\_S_i$ are set to 0 to obtain the final sampling map $Sample\_S$. Consequently, reparameterization [13] trick is performed to $Sample\_S$ to achieve differentiability. Based on the above designs, the LSS module realizes the derivable dynamic sample allocation without hyperparameters.

[1] See the Supplementary Material for more proof details.

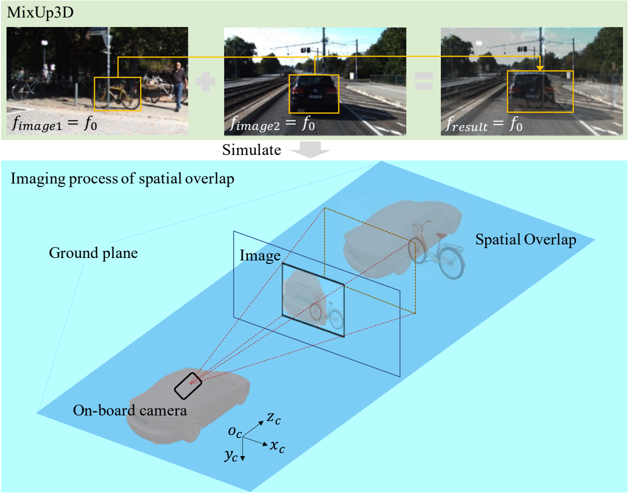

Figure 3. **Visualization of the MixUp3D which simulates spatial overlap.** A car overlaps a bicycle in the physical world and their appearance features in resulting image do not introduce ambiguity for 3D property learning.

3.2. Loss Function and Training Strategy

The overall loss L consists of 2D loss L_{2d} and 3D lossL_{3d}, where the 2D loss L_{2d} follows the design in CenterNet [61] and the 3D loss L_{3d} employs a multi-task loss function based on the LSS module to supervise the learning of the 3D properties:

$$L = L_{2d} + L_{O_{3d}} + (L_{S_{3d}} + L_{depth} + L_{\theta}) \cdot Sample\_S \tag{4}$$

where the $L_{S_{3d}}$ is L1 loss for dimensions predict, and the $L_{O_{3d}}$ denotes Smooth-L1 loss for 3D center projection offset regression. The L_{depth} denotes the depth loss where we employ the Laplacian aleatoric uncertainty loss [5, 18] to supervise depth estimation. The L_{θ} denotes orientation loss, which use the multi-bin loss that follow the [38]. After calculating the loss of each property, we multiply loss map of the depth, orientation and dimension properties by final sampling map $Sample\_S$ of LSS to prevent the loss backpropagation of the negative samples.

The HTL [34] training strategy is employed to reduce instability. In addition, since the instability of the 3D properties loss will interfere with the selection of positive sample points, we use a warm-up strategy when training the LSS module. Specifically, all samples will be used to learn 3D properties in the early stage of training until the depth loss stabilizes. We consider the depth loss stabilization as a necessary condition to start the LSS module, since depth is the most important property that affects the accuracy of 3D bounding boxes [29, 36].

3.3. MixUp3D for Spatial Overlap Simulation

Due to strict imaging constraints, data augmentation methods are limited in monocular 3D detection. Besides photometric distortion and horizontal flipping, most data augmentation methods introduce ambiguous features due to breaking the imaging principles. Additionally, since the LSS module focuses on object-level features, methods that do not modify the features of the objects themselves are not expected to be effective enough for the LSS module.

Thanks to the advantages of the MixUp [57, 58], pixel-level features of objects can be enhanced. We propose MixUp3D, which adds physical constraints to the 2D MixUp, enabling the newly generated image is essentially plausible imaging of spatial overlap. Specifically, the MixUp3D violates only the collision constraints of objects in the physical world, while ensuring that the resulting image adheres to imaging principles, thus avoiding any ambiguity.

Traditional MixUp method blends different images proportionally in a 2D pixel coordinate system, without considering whether the resulting image is compatible with the imaging principles in the 3D physical world. For example, two images with different focal lengths or resolutions are always directly mixed together, which introduces depth ambiguity. Another example is that the mixing of images taken from different views can result in confusion with regard to perspective. In this paper, we impose strict constraints on the MixUp images to ensure they have the same focal length, principal points, resolution and views of camera (pitch and roll angle). This enables the simulation of an image captured by one camera at a single time with spatial overlap by leveraging images taken by two cameras at different times and locations. Generally, images with the same focal length always mean that their principal points and resolutions are also the same. Concurrently, the images are all captured by on-board pinhole cameras, with their x_c-axis and z_c-axis being parallel to the ground plane, leading to similar views. Therefore, the MixUp3D only needs to consider ensuring that the focal lengths of the images are the same. The schematic diagram of the MixUp3D is shown in Figure 3.

Considering a training dataset $I = \{I_{f_1}, I_{f_2}, ..., I_{f_K}\}$, where I_{f_k} for $k \in [1, K]$ means images in I_{f_k} have the same focus f_k. We denote images and corresponding labels as $I_{f_k} = \{(x_1^{f_k}, y_1^{f_k}), (x_2^{f_k}, y_2^{f_k}), ..., (x_N^{f_k}, y_N^{f_k})\}$, which has totally N samples. The MixUp3D process can be defined as the following form:

$$\begin{cases} x_n^{f_k} = \lambda \cdot x_n^{f_k} + (1 - \lambda) \cdot x_m^{f_k} \\ y_n^{f_k} = y_n^{f_k} + y_m^{f_k} \end{cases} \tag{5}$$

where $n, m \in [1, N]$ and $n \neq m$. λ denotes mix proportion.

The proposed MixUp3D can enrich training samples without introducing ambiguity, and effectively alleviate

Method	Reference	Extra Data	$AP_{3D}(IOU=0.7\|R_{40})$			$AP_{BEV}(IOU=0.7\|R_{40})$			Runtime
			Easy	**Mod.**	Hard	Easy	Mod.	Hard	(ms)
MonoPSR [23]	CVPR 2019	LIDAR	10.76	7.25	5.85	18.33	12.58	9.91	200
PatchNet [35]	ECCV 2020	Depth	15.68	11.12	10.17	22.97	16.86	14.97	400
MonoRUn [3]	CVPR 2021	LIDAR	19.65	12.30	10.58	27.94	17.34	15.24	70
CaDDN [41]	CVPR 2021	LIDAR	19.17	13.41	11.46	27.94	18.91	17.19	630
DFR-Net [63]	ICCV 2021	Depth	19.40	13.63	10.35	28.17	19.17	14.84	180
AutoShape [33]	ICCV 2021	CAD	22.47	14.17	11.36	30.66	20.08	15.59	50
DID-M3D [40]	ECCV 2022	Depth	24.40	16.29	13.75	32.95	22.76	19.83	40
DD3D [39]	ICCV 2021	Depth	23.22	16.34	14.20	30.98	22.56	20.03	-
CMKD [11]	ECCV 2022	LIDAR	25.09	16.99	15.30	33.69	23.10	20.67	-
M3D-RPN [1]	ICCV 2019	None	14.76	9.71	7.42	21.02	13.67	10.23	160
SMOKE [32]	CVPR 2020	None	14.03	9.76	7.84	20.83	14.49	12.75	30
MonoPair [5]	CVPR2020	None	13.04	9.99	8.65	19.28	14.83	12.89	60
MonoDLE [36]	CVPR2021	None	17.23	12.26	10.29	24.79	18.89	16.00	40
MonoFlex [60]	CVPR 2021	None	19.94	13.89	12.07	28.23	19.75	16.89	30
GUPNet [34]	ICCV 2021	None	20.11	14.20	11.77	-	-	-	30
DEVIANT [25]	ECCV 2022	None	21.88	14.46	11.89	29.65	20.44	17.43	-
MonoCon [50]	AAAI 2022	None	22.50	16.46	13.95	31.12	22.10	19.00	25
MonoDDE [28]	CVPR 2022	None	24.93	17.14	15.10	33.58	23.46	20.37	40
MonoLSS (Ours)	-	None	**26.11**	**19.15**	**16.94**	**34.89**	**25.95**	**22.59**	35

Table 1. **Monocular 3D detection performance of Car category on KITTI *test* set.** All results are evaluated on KITTI testing server. Same as KITTI leaderboard, methods are ranked under the moderate difficulty level. We highlight the best results in bold and the second ones in underlined. For the extra data: 1) **LIDAR** denotes methods use extra LIDAR cloud points in training process. 2) **Depth** means utilizing depth maps or models pre-trained under another depth estimation dataset. 3) **CAD** denotes using dense shape annotations provided by CAD models. 4) **None** means no extra data is used.

Method	Extra	$AP_{3D}(IOU=0.5\|R_{40})$					
		Pedestrian			Cyclist		
		Easy	**Mod.**	Hard	Easy	Mod.	Hard
DFR-Net [63]	Depth	6.09	3.62	3.39	5.69	3.58	3.10
CaDDN [41]	LIDAR	12.87	8.14	6.76	7.00	3.41	3.30
DD3D [39]	Depth	13.91	9.30	8.05	2.39	1.52	1.31
CMKD [11]	LIDAR	17.79	11.69	10.09	9.60	5.24	4.50
MonoDDE [28]	None	11.13	7.32	6.67	5.94	3.78	3.33
MonoCon [50]	None	13.10	8.41	6.94	2.80	1.92	1.55
GUPNet [34]	None	14.95	9.76	8.41	5.58	3.21	2.66
MonoDTR [12]	None	15.33	10.18	8.61	5.05	3.27	3.19
MonoLSS	None	**17.09**	**11.27**	**10.00**	**7.23**	**4.34**	**3.92**

Table 2. **Monocular 3D detection performance of Pedestrian and Cyclist category on KITTI *test* set.**

overfitting problems. It can be conveniently applied in any monocular 3D detection task as an essential data augmentation method.

4. Experiments

4.1. Setup

Dataset and Evaluation metrics. We evaluate our proposed method on the widely used KITTI [9], Waymo [47] and nuScenes [2] benchmarks.

- ***KITTI*** consists of 7481 training images and 7518 testing images. It has three classes (Car, Pedestrian, and Cyclist), each with three difficulty levels (Easy, Moderate, and Hard). We follow the prior work [4] to divide the 7481 training images into a training set (3712) and validation set (3769) for ablation study. Following the official protocol [46], we use $AP_{3D|R_{40}}$ and $AP_{BEV|R_{40}}$ on Moderate category as main metrics.
- ***Waymo*** evaluates objects at two levels: Level 1 and Level 2, based on the number of LiDAR points present in their 3D box. The evaluation is conducted at three distances: [0, 30), [30, 50), and [50, ∞) meters. Waymo utilizes the APH_{3D} percentage metric, which incorporates heading information in AP_{3D}, as a benchmark for evaluation.
- ***nuScenes*** comprises 28,130 training and 6,019 validation images captured from the front camera. We use validation split for cross-dataset evaluation.

Implementation details. Our proposed MonoLSS is trained on 4 Tesla V100 GPUs with a batch size of 16. Without the MixUp3D, we train the model for 150 epochs, after which overfitting occurrs. While using MixUp3D, the model can be trained for 600 epochs without overfitting. We use Adam [19] as our optimizer with an initial learning rate $1e-3$. The learning scheduler has a linear warm-up strategy in the first 5 epochs. Following [40], the ROI-Align size $d \times d$ is set to 7×7. The LSS starts after 0.3 × total epochs (experimental parameters) for warmup.

4.2. Main Results

Results of Car category on KITTI test set. As shown in Table 1, our proposed MonoLSS achieves superior performance than previous methods, even those with extra data. Specifically, compared with MonoDDE [28] which is the recent top1-ranked image-only method, MonoLSS gains significant improvement of **4.73%/11.73%/12.19%** in AP_{3D} and **3.90%/10.61%/10.90%** in AP_{BEV} relatively on the easy, moderate, and hard levels while $IOU = 0.7$.

Results of Pedestrian/Cyclist on KITTI test set. We present the results of pedestrians and cyclists on the test set

IOU_{3D}	Difficulty	Method	Extra	AP_{3D}				APH_{3D}			
				All	0-30	30-50	50-∞	All	0-30	30-50	50-∞
0.7	Level_1	CaDDN [41]	LIDAR	5.03	15.54	1.47	0.10	4.99	14.43	1.45	0.10
		PatchNet [35] in [48]	Depth	0.39	1.67	0.13	0.03	0.39	1.63	0.12	0.03
		PCT [48]	Depth	0.89	3.18	0.27	0.07	0.88	3.15	0.27	0.07
		M3D-RPN [1] in [41]	None	0.35	1.12	0.18	0.02	0.34	1.10	0.18	0.02
		GUPNet [34] in [25]	None	2.28	6.15	0.81	0.03	2.27	6.11	0.80	0.03
		DEVIANT [25]	None	2.69	6.95	0.99	0.02	2.67	6.90	0.98	0.02
		MonoLSS (Ours)	None	**3.71**	**9.82**	**1.14**	**0.16**	**3.69**	**9.75**	**1.13**	**0.16**
0.7	Level_2	CaDDN [41]	LIDAR	4.49	14.50	1.42	0.09	4.45	14.38	1.41	0.09
		PatchNet [35] in [48]	Depth	0.38	1.67	0.13	0.03	0.36	1.63	0.11	0.03
		PCT [48]	Depth	0.66	3.18	0.27	0.07	0.66	3.15	0.26	0.07
		M3D-RPN [1] in [41]	None	0.35	1.12	0.18	0.02	0.33	1.10	0.17	0.02
		GUPNet [34] in [25]	None	2.14	6.13	0.78	0.02	2.12	6.08	0.77	0.02
		DEVIANT [25]	None	2.52	6.93	0.95	0.02	2.50	6.87	0.94	0.02
		MonoLSS (Ours)	None	**3.27**	**9.79**	**1.11**	**0.15**	**3.25**	**9.73**	**1.10**	**0.15**
0.5	Level_1	CaDDN [41]	LIDAR	17.54	45.00	9.24	0.64	17.31	44.46	9.11	0.62
		PatchNet [35] in [48]	Depth	2.92	10.03	1.09	0.23	2.74	9.75	0.96	0.18
		PCT [48]	Depth	4.20	14.70	1.78	0.39	4.15	14.54	1.75	0.39
		M3D-RPN [1] in [41]	None	3.79	11.14	2.16	0.26	3.63	10.70	2.09	0.21
		GUPNet [34] in [25]	None	10.02	24.78	4.84	0.22	9.94	24.59	4.78	0.22
		DEVIANT [25]	None	10.98	26.85	5.13	0.18	10.89	26.64	5.08	0.18
		MonoLSS (Ours)	None	**13.49**	**33.64**	**6.45**	**1.29**	**13.38**	**33.39**	**6.40**	**1.26**
0.5	Level_2	CaDDN [41]	LIDAR	16.51	44.87	8.99	0.58	16.28	44.33	8.86	0.55
		PatchNet [35] in [48]	Depth	2.42	10.01	1.07	0.22	2.28	9.73	0.97	0.16
		PCT [48]	Depth	4.03	14.67	1.74	0.36	4.15	14.51	1.71	0.35
		M3D-RPN [1] in [41]	None	3.61	11.12	2.12	0.24	3.46	10.67	2.04	0.20
		GUPNet [34] in [25]	None	9.39	24.69	4.67	0.19	9.31	24.50	4.62	0.19
		DEVIANT [25]	None	10.29	26.75	4.95	0.16	10.20	26.54	4.90	0.16
		MonoLSS (Ours)	None	**13.12**	**33.56**	**6.28**	**1.15**	**13.02**	**33.32**	**6.22**	**1.13**

Table 3. **Monocular 3D detection performance of Vehicle category on Waymo *val* set.**

Method	KITTI Val				nuScenes frontal Val			
	0-20	20-40	40-∞	All	0-20	20-40	40-∞	All
M3D-RPN [1]	0.56	1.33	2.73	1.26	0.94	3.06	10.36	2.67
MonoRCNN [45]	0.46	1.27	2.59	1.14	0.94	2.84	8.65	2.39
GUPNet [34]	0.45	1.10	1.85	0.89	0.82	1.70	6.20	1.45
DEVIANT [25]	0.40	1.09	1.80	0.87	0.76	**1.60**	**4.50**	**1.26**
MonoLSS	**0.35**	**0.89**	**1.77**	**0.82**	**0.59**	2.01	5.40	1.42

Table 4. **Cross-dataset evaluation of the KITTI *val* model on KITTI *val* and nuScenes frontal *val* cars with depth MAE.**

of KITTI in Table 2. MonoLSS outperforms all image-only methods by a large margin. When compared with methods using extra data, MonoLSS performs better than most of them while only slightly weaker than CMKD [11].

Results on Waymo val set. We evaluated our MonoLSS method on the Waymo dataset, which is more diverse than KITTI. The experimental results presented in Table 3 demonstrate that MonoLSS achieves superior performance compared to the state-of-the-art DEVIANT method [25] across multiple evaluation metrics and thresholds, particularly for nearby objects. Notably, MonoLSS also outperforms PatchNet [35] and PCT [48] without utilizing depth information. Although MonoLSS's performance is slightly lower than that of CaDDN [41], it is worth noting that CaDDN relies on LiDAR data during training, whereas MonoLSS is an image-only approach.

Cross-Dataset Evaluation. Tabel 4 shows the result of our KITTI val model on the KITTI val and nuScenes [2] frontal val images, using mean absolute error (MAE) of the depth [45]. MonoLSS is better than GUPNet [34] and achieves similar competitive performance to DEVIANT [25]. This is because DEVIANT is equivariant to the depth translations and is more robust to data distribution changes.

D	S	θ	W	M	$AP_{3D}/AP_{BEV}(IOU=0.7 \vert R_{40})$		
					Easy	Mod.	Hard
					21.72/30.74	15.63/22.74	12.80/19.14
✓					17.03/24.40	12.77/18.73	11.08/15.84
✓			✓		24.78/33.32	17.65/23.92	14.53/20.21
✓	✓	✓	✓		24.63/33.63	17.55/25.03	14.62/21.48
				✓	24.65/34.15	17.33/24.56	14.34/20.97
✓	✓	✓	✓	✓	25.69/33.54	17.84/24.62	15.62/21.25
✓			✓	✓	**25.91/34.70**	**18.29/25.36**	**15.94/21.84**

Table 5. **Ablation Study on different components of our overall framework on KITTI *val* set for Car category**. D, S, and θ denote that the LSS module acts on depth, dimension, and orientation angle, respectively. W denotes warm-up strategy and M denotes the MixUp3D.

4.3. Ablation Study

In this subsection, we investigate the impact of each component in our method[2]. All ablation results are reported on the Car class of KITTI validation set and trained 150 epochs. To ensure result reliability, we report the median performance across five different seeds for each ablation experiment.

Effectiveness of the LSS Module. As reported in the first

[2] See the Supplementary Material for more ablation study, such as different Sampling Strategies on the LSS and influence of the MixUp3D.

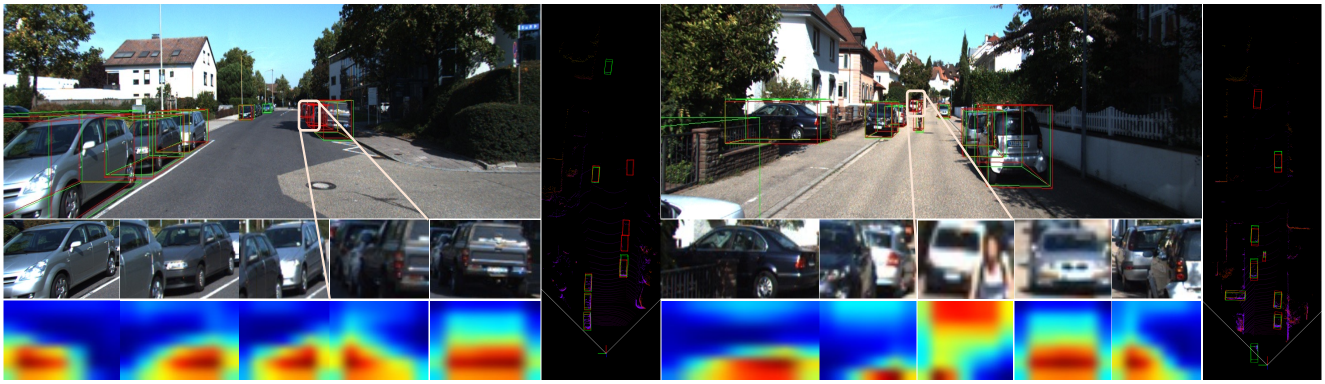

Figure 4. **Qualitative visualization of some samples on KITTI *val* set.** The 3D red boxes are produced by MonoLSS and the green boxes are the ground truth. Some unlabeled objects detected by MonoLSS are highlighted on images. The last line represents the LSS sampling map of the corresponding object. Best viewed in color with zoom in.

Method	$AP_{3D}/AP_{BEV}(IOU = 0.7\|R_{40})$		
	Easy	Mod.	Hard
1×1	18.14/25.54	12.68/18.68	10.36/15.66
3×3	20.06/28.30	15.28/21.95	12.92/19.00
5×5	20.75/30.23	15.14/22.39	13.02/18.93
7×7	21.72/30.74	15.63/22.74	12.80/19.14
Depth	23.85/31.36	15.92/21.17	11.64/16.76
Seg	24.11/31.82	15.82/21.51	13.09/16.90
LSS	**24.78/33.32**	**17.65/23.92**	**14.53/20.21**

Table 6. **Comparison of the LSS module with other sample selection methods on KITTI *val* set for Car category.** The $\mathbf{1 \times 1}$, $\mathbf{3 \times 3}$, $\mathbf{5 \times 5}$ respectively represent that 1, 9, and 25 sample points in the center are regarded as positive samples, and $\mathbf{7 \times 7}$ indicates that the method takes all sample points as positive samples. **Depth** denotes using extra depth map to select positive samples. **Seg** means using segmentation pseudo-labels generated by SAM [20]. **LSS** means our proposed LSS method.

4 rows of Table 5, the LSS module can significantly improve the AP. While the effect is slightly off when it acts on the orientation and dimension property. This is because the depth estimation error is the most critical limiting factor in monocular 3D detection, which has been identified in GeoAug [29] and MonoDLE [36]. Thus, for the convenience of comparison, the LSS module only acts on the depth property in the following ablation experiments.

Necessity of the warm-up. Results in the first 3 rows of Table 5 show the importance of the warm-up strategy. Without warm-up (2nd row), LSS will start random sampling at the beginning of training, leading to the fact that the true negative samples may be forced to learn attributes while the true positive samples are discarded instead, which significantly reduces the performance (21.72 to 17.03 in Easy level).

Effect between LSS and MixUp3D. As presented in Table 5, while the LSS module and MixUp3D each exhibit a positive impact when applied independently (+3.06 and +2.93 in Easy level), their combined usage results in a higher improvement (+4.19).

Comparison of Sample Selection Strategies. We contrast the LSS module with other sample allocation strategies, and the results are shown in Table 6. The LSS module adaptively selects positive samples based on object features, leading to a significant improvement in AP over methods that treat fixed-position sample points as positive ones. Furthermore, our method also outperforms the approach of using an additional depth or segmentation map to select positive samples.

4.4. Qualitative Results

In this subsection, we show 3D detection results of the MonoLSS in images and BEV maps. As shown in Figure 4, the MonoLSS can accurately estimate the 3D position of objects, even those not labeled by the annotators. In order to explore what features the LSS module is concerned with, we visualize the sampling map. As shown in the last row of Figure 4, the LSS module can adaptively determine the positive samples that are more suitable for learning 3D properties. Generally, it prefers to choose the bottom part of an object as positive samples. When occlusion occurs, the LSS module focuses on the regions without occlusion.

5. Conclusion

In this paper, we point out that the 3D property learning of monocular 3D detection faces a sample selection problem. We adopt a LSS module to adaptively determine the positive samples for each object. Moreover, we propose an unambiguous data augmentation method MixUp3D to improve the diversity of object-level samples. Extensive experiments on the KITTI, Waymo and nuScenes benchmarks verify the effectiveness and efficiency of our MonoLSS.

References

[1] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings*

of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019. 2, 6, 7

[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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 6, 7

[3] Hansheng Chen, Yuyao Huang, Wei Tian, Zhong Gao, and Lu Xiong. Monorun: Monocular 3d object detection by reconstruction and uncertainty propagation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10379–10388, 2021. 3, 6

[4] 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

[5] 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 (CVPR)*, 2020. 2, 3, 5, 6

[6] Ziming Chen, Yifeng Shi, and Jinrang Jia. Transiff: An instance-level feature fusion framework for vehicle-infrastructure cooperative 3d detection with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 18205–18214, 2023. 1

[7] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[8] Mingyu Ding, Yuqi Huo, Hongwei Yi, Zhe Wang, Jianping Shi, Zhiwu Lu, and Ping Luo. Learning depth-guided convolutions for monocular 3d object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[9] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 2, 6

[10] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office, 1954. 4

[11] Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3d object detection. In *ECCV*. Springer, 2022. 2, 6, 7

[12] Kuan-Chih Huang, Tsung-Han Wu, Hung-Ting Su, and Winston H. Hsu. Monodtr: Monocular 3d object detection with depth-aware transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4012–4021, 2022. 6

[13] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017. 2, 4

[14] Jinrang Jia, Zhenjia Li, and Yifeng Shi. MonoUNI: A unified vehicle and infrastructure-side monocular 3d object detection network with sufficient depth clues. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1

[15] Jinrang Jia, Yifeng Shi, Yuli Qu, Rui Wang, Xing Xu, and Hai Zhang. Competition for roadside camera monocular 3d object detection. *National Science Review*, 10(6):nwad121, 2023. 1

[16] Bo Ju, Wei Yang, Jinrang Jia, Xiaoqing Ye, Qu Chen, Xiao Tan, Hao Sun, Yifeng Shi, and Errui Ding. Danet: Dimension apart network for radar object detection. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, page 533–539, New York, NY, USA, 2021. Association for Computing Machinery. 1

[17] Wei Ke, Tianliang Zhang, Zeyi Huang, Qixiang Ye, Jianzhuang Liu, and Dong Huang. Multiple anchor learning for visual object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[18] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017. 5

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023. 8

[21] Xianghao Kong, Wentao Jiang, Jinrang Jia, Yifeng Shi, Runsheng Xu, and Si Liu. Dusa: Decoupled unsupervised sim2real adaptation for vehicle-to-everything collaborative perception. In *Proceedings of the 31st ACM International Conference on Multimedia*, page 1943–1954, New York, NY, USA, 2023. Association for Computing Machinery. 1

[22] Wouter Kool, Herke Van Hoof, and Max Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019. 2, 4

[23] Jason Ku, Alex D. Pon, and Steven L. Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6

[24] Abhinav Kumar, Garrick Brazil, and Xiaoming Liu. GrooMeD-NMS: Grouped mathematically differentiable nms for monocular 3D object detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[25] Abhinav Kumar, Garrick Brazil, Enrique Corona, Armin Parchami, and Xiaoming Liu. Deviant: Depth equivariant network for monocular 3d object detection. In *European Conference on Computer Vision*, pages 664–683. Springer, 2022. 6, 7

[26] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019. 1

[27] ZhenJia Li, SongLu Chen, Qi Liu, Feng Chen, and XuCheng Yin. Anchor-free location refinement network for small license plate detection. In *Pattern Recognition and Computer Vision*, pages 506–519, Cham, 2022. Springer Nature Switzerland. 3

[28] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2791–2800, 2022. 1, 2, 3, 6

[29] Qing Lian, Botao Ye, Ruijia Xu, Weilong Yao, and Tong Zhang. Exploring geometric consistency for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2022. 1, 3, 5, 8

[30] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 2

[31] Yuxuan Liu, Yuan Yixuan, and Ming Liu. Ground-aware monocular 3d object detection for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2):919–926, 2021. 1

[32] Zechen Liu, Zizhang Wu, and Roland Toth. Smoke: Single-stage monocular 3d object detection via keypoint estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 1, 2, 3, 6

[33] Zongdai Liu, Dingfu Zhou, Feixiang Lu, Jin Fang, and Liangjun Zhang. Autoshape: Real-time shape-aware monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15641–15650, 2021. 3, 6

[34] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3111–3121, 2021. 2, 3, 4, 5, 6, 7

[35] Xinzhu Ma, Shinan Liu, Zhiyi Xia, Hongwen Zhang, Xingyu Zeng, and Wanli Ouyang. Rethinking pseudo-lidar representation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 6, 7

[36] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4721–4730, 2021. 1, 5, 6, 8

[37] Fabian Manhardt, Wadim Kehl, and Adrien Gaidon. Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[38] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Kosecka. 3d bounding box estimation using deep learning and geometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[39] Dennis Park, Rares Ambrus, Vitor Guizilini, Jie Li, and Adrien Gaidon. Is pseudo-lidar needed for monocular 3d object detection? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3142–3152, 2021. 6

[40] Liang Peng, Xiaopei Wu, Zheng Yang, Haifeng Liu, and Deng Cai. Did-m3d: Decoupling instance depth for monocular 3d object detection. In *European Conference on Computer Vision*, 2022. 1, 2, 3, 6

[41] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distribution network for monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8555–8564, 2021. 2, 6, 7

[42] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[43] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 1, 3

[44] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15172–15181, 2021. 3

[45] Xuepeng Shi, Qi Ye, Xiaozhi Chen, Chuangrong Chen, Zhixiang Chen, and Tae-Kyun Kim. Geometry-based distance decomposition for monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15172–15181, 2021. 7

[46] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 6

[47] 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/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 6

[48] 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. 7

[49] Xiaopei Wu, Liang Peng, Honghui Yang, Liang Xie, Chenxi Huang, Chengqi Deng, Haifeng Liu, and Deng Cai. Sparse fuse dense: Towards high quality 3d detection with depth completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5418–5427, 2022. 1

[50] Tianfu Wu Xianpeng Liu, Nan Xue. Learning auxiliary monocular contexts helps monocular 3d object detection. In *36th AAAI Conference on Artifical Intelligence (AAAI)*, 2022. 1, 2, 3, 6

[51] 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/CVF International Conference on Computer Vision (ICCV)*, 2019. 1

[52] Xiaoqing Ye, Liang Du, Yifeng Shi, Yingying Li, Xiao Tan, Jianfeng Feng, Errui Ding, and Shilei Wen. Monocular 3d object detection via feature domain adaptation. In *European Conference on Computer Vision*, pages 17–34. Springer, 2020. 1

[53] Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21341–21350, 2022. 3

[54] 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)*, 2018. 3

[55] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, and Zaiqing Nie. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21361–21370, 2022. 1

[56] Haibao Yu, Wenxian Yang, Hongzhi Ruan, Zhenwei Yang, Yingjuan Tang, Xu Gao, Xin Hao, Yifeng Shi, Yifeng Pan, Ning Sun, Juan Song, Jirui Yuan, Ping Luo, and Zaiqing Nie. V2x-seq: A large-scale sequential dataset for vehicle-infrastructure cooperative perception and forecasting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5486–5495, 2023. 1

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2, 5

[58] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? In *International Conference on Learning Representations*, 2020. 5

[59] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z. Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[60] Yunpeng Zhang, Jiwen Lu, and Jie Zhou. Objects are different: Flexible monocular 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3289–3298, 2021. 2, 3, 6

[61] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. In *arXiv preprint arXiv:1904.07850*, 2019. 1, 2, 3, 5

[62] Yunsong Zhou, Yuan He, Hongzi Zhu, Cheng Wang, Hongyang Li, and Qinhong Jiang. Monocular 3d object detection: An extrinsic parameter free approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7556–7566, 2021. 2

[63] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2713–2722, 2021. 3, 6