Supplementary Materials of *LIGA-Stereo: Learning LiDAR Geometry Aware Representations for Stereo-based 3D Detector*

Xiaoyang Guo Shaoshuai Shi Xiaogang Wang Hongsheng Li
CUHK-SenseTime Joint Laboratory, The Chinese University of Hong Kong
{xyguo, ssshi, xgwang, hsli}@ee.cuhk.edu.hk

1. More Implementation Details

1.1. The Structure of the Stereo Detector

The main structure of our stereo 3D detector follows that of DSGN [2] but with much less memory consumption and lower computation cost. The network structure is described in Table 1 in details.

2D Feature Extraction. For the 2D backbone network, we employ a modified version of ResNet-34 [3] with spatial pyramid pooling (SPP) module and feature upsampling. Compared with [2], the numbers of blocks in *conv2-5* are reduced from {3, 6, 12, 4} to {3, 4, 6, 3}. The channels of *conv2-5* are set to {64, 128, 128, 128}. The SPP module is the same as previous implementations [1, 2]. In addition, we append a small U-Net [5] on the top of the 2D backbone to upsample the SPP feature back into full resolution to provide high-resolution features for stereo matching.

Stereo Aggregation Network. Although the 2D stereo features $\mathcal{F}_l, \mathcal{F}_r$ are at full image resolution, we still construct the plane sweep volume (the stereo cost volume) at $1/4$ resolution to save memory. The number of base channels of the stereo aggregation network is set to 32, which is half of the number of channels in [2].

Space Conversion. The volumetric feature in stereo space $\mathcal{V}_{st}$ is converted into 3D space using Eq. (2) in the paper, which is then filtered by a 3D convolution layer and average pooling layer (along Y dimension) to output the volume in 3D space $\mathcal{V}_{3d}$. The BEV feature is constructed by merging the y dimension and the channel dimension of $\mathcal{V}_{3d}$ and then compressed into 64 channels.

BEV Aggregation Network. The structure for the BEV aggregation network follows [2], which is a shallow hourglass-like network.

3D Detection Head. For the 3D detection head, we follow the open source implementation of SECOND [7] in OpenPCDet [6]. For each class and each (x, z) location in the BEV space, we create two anchors with fixed average size and rotations of 0 and 90 degrees. The default anchor sizes are l_a=3.9, w_a=1.6, h_a=1.56 for cars, l_a=0.8, w_a=0.6, h_a=1.73 for pedestrians, and l_a=1.76, w_a=0.6, h_a=1.73 for cyclists, and their y coordinates are set to y_a=$\{1.78m, 0.6m, 0.6m\}$, respectively. The training targets are assigned with IoU-based criteria. The matched and unmatched IoU thresholds for the three classes are set to 0.6, 0.5, 0.5 and 0.45, 0.35, 0.35. Anchors with IoU between matched and unmatched thresholds are ignored for training. For each anchor, we output 3-dimensional classification and 7-dimensional regression predictions. The 3D bounding box regression targets are given by the following box encoding functions,

$$\begin{aligned} &x_t = \frac{x_g - x_a}{d_a}, y_t = \frac{y_g - y_a}{h_a}, z_t = \frac{z_g - z_a}{d_a}, \\ &w_t = \log \frac{w_g}{w_a}, l_t = \log \frac{l_g}{l_a}, h_t = \log \frac{h_g}{h_a}, \\ &\theta_t = \theta_g - \theta_a, \end{aligned} \tag{1}$$

where the subscripts t, a, g denote the encoded regression targets, the anchors, and the ground truth. θ is the yaw direction around the y-axis.

The classification loss $\mathcal{L}_{cls}$, the direction classification loss $\mathcal{L}_{dir}$, and the L1 regression loss $\mathcal{L}_{reg}^{L1}$ are the same as SECOND [7], and the auxiliary IoU-based regression loss is defined by,

$$\mathcal{L}_{reg}^{IoU} = 1 - \text{IoU}_{3d}\left(\text{decode}(\delta_p, \xi_a), \xi_g\right) \tag{2}$$

where δ_p is the regression predictions, ξ_a is the anchor $\{x_a, y_a, z_a, w_a, l_a, h_a, \theta_a\}$, and ξ_g is the assigned ground-truth bounding box $\{x_g, y_g, z_g, w_g, l_g, h_g, \theta_g\}$. Similar to the L1 regression loss, the IoU regression loss is only applied to positive samples. Non-maximum suppression (NMS) is applied to the 3D box predictions for each class separately, with the IoU threshold set to 0.25.

2D Detection Head. From the 2D feature extraction part, we have obtained 5-level FPN features *spp1-5* from $\mathcal{F}_{sem}$ with a sequence of stride-2 convolution layers. The strides of these features are $\{4, 8, 16, 32, 64\}$. For each level of the features, we apply a 2D detection head with two branches, classification branch and regression branch, to predict 2D bounding boxes. Following ATSS [8], each position is only

Output	Input	Module Config	#Channel	Size
2D Feature Extraction				
conv1	$\mathcal{I}_{l/r}$	Conv (7×7), s=2	64	$^{H}/_{2}\times^{W}/_{2}$
conv2		BasicBlock×3	64	$^{H}/_{2}\times^{W}/_{2}$
conv3		BasicBlock×4, s=2	128	$^{H}/_{4}\times^{W}/_{4}$
conv4		BasicBlock×6, d=2	128	$^{H}/_{4}\times^{W}/_{4}$
conv5		BasicBlock×3, d=4	128	$^{H}/_{4}\times^{W}/_{4}$
spp1		AvgPool (64×64); Conv (1×1); Upsample 64×	32	$^{H}/_{4}\times^{W}/_{4}$
spp2		AvgPool (32×32); Conv (1×1); Upsample 32×	32	$^{H}/_{4}\times^{W}/_{4}$
spp3		AvgPool (16×16); Conv (1×1); Upsample 16×	32	$^{H}/_{4}\times^{W}/_{4}$
spp4		AvgPool (8×8); Conv (1×1); Upsample 8×	32	$^{H}/_{4}\times^{W}/_{4}$
spp	*spp1-4, conv3-5*	Concat	512	$^{H}/_{4}\times^{W}/_{4}$
hres1	*conv2*	Conv (1×1)	64	$^{H}/_{2}\times^{W}/_{2}$
hres2	$\mathcal{I}_{l/r}$	Conv (1×1)	32	$H\times W$
up1	*spp*	Conv (3×3); Upsample 2×; Add *hres1*; ReLU	64	$^{H}/_{2}\times^{W}/_{2}$
up2		Conv (3×3); Upsample 2×; Add *hres2*; ReLU	32	$H\times W$
$\mathcal{F}_{l/r}$		Conv (3×3)×2	32, 32	$H\times W$
$\mathcal{F}_{sem}$	*spp* of $\mathcal{I}_{l}$	Conv (3×3)×2	128, 32	$^{H}/_{4}\times^{W}/_{4}$
fpn_pre		Conv (1×1)	64	$^{H}/_{4}\times^{W}/_{4}$
fpn0		Conv (3×3)	64	$^{H}/_{4}\times^{W}/_{4}$
fpn1		Conv (3×3), s=2	64	$^{H}/_{8}\times^{W}/_{8}$
fpn2		Conv (3×3), s=2	64	$^{H}/_{16}\times^{W}/_{16}$
fpn3		Conv (3×3), s=2	64	$^{H}/_{32}\times^{W}/_{32}$
fpn4		Conv (3×3), s=2	64	$^{H}/_{64}\times^{W}/_{64}$
Stereo Aggregation Network				
$\mathcal{V}_{st}$	$\mathcal{F}_{l}$, $\mathcal{F}_{r}$	Construct Plane Sweep Volume (Eq.(1))	64	$^{D}/_{4}\times^{H}/_{4}\times^{W}/_{4}$
st_conv1		Conv (3×3×3)	32	$^{D}/_{4}\times^{H}/_{4}\times^{W}/_{4}$
st_conv2		Conv (3×3×3); Add *st_conv1*	32	$^{D}/_{4}\times^{H}/_{4}\times^{W}/_{4}$
st_hg1		Conv (3×3×3)×2, s=2	64	$^{D}/_{8}\times^{H}/_{8}\times^{W}/_{8}$
st_hg2		Conv (3×3×3)×2, s=2	64	$^{D}/_{16}\times^{H}/_{16}\times^{W}/_{16}$
st_hg3		Deconv (3×3×3); Add *st_hg1*; ReLU	64	$^{D}/_{8}\times^{H}/_{8}\times^{W}/_{8}$
$\tilde{\mathcal{V}}_{st}$		Deconv (3×3×3); Add *st_conv2*	32	$^{D}/_{4}\times^{H}/_{4}\times^{W}/_{4}$
st_prob		(3×3×3)×2	32, 1	$^{D}/_{4}\times^{H}/_{4}\times^{W}/_{4}$
$\mathcal{P}_{st}$		Upsample 4×; Softmax	1	$D\times H\times W$
Stereo Space →3D Space →BEV Space				
$\mathcal{V}_{3d}^{raw}$	$\mathcal{F}_{sem}$, $\mathcal{V}_{st}$	Construct 3D Volume (Eq.(2))	64	$N_x\times N_y\times N_z$
$\mathcal{V}_{3d}$		Conv (3×3×3); AvgPool (1×4×1)	32	$N_x\times^{N}y/_{4}\times N_z$
$\mathcal{F}_{BEV}$		Reshape; Conv (3×3)	32×$^{N}y/_{4}$, 64	$N_x\times N_z$
BEV Aggregation Network				
bev_hg1		Conv (3×3)×2, s=2	128	$^{N_x}/_{2}\times^{N_z}/_{2}$
bev_hg2		Conv (3×3)×2, s=2	128	$^{N_x}/_{4}\times^{N_z}/_{4}$
bev_hg3		Deconv (3×3); Add *bev_hg1*; ReLU	128	$^{N_x}/_{2}\times^{N_z}/_{2}$
$\tilde{\mathcal{F}}_{BEV}$		Deconv (3×3)	64	$N_x\times N_z$
3D Detection Head				
conv_cls	$\tilde{\mathcal{F}}_{BEV}$	Conv (3×3)×2	64	$N_x\times N_z$
bbox_cls	*conv_cls*	Conv (3×3)	6×3	$N_x\times N_z$
bbox_dir	*conv_cls*	Conv (1×1)	6×2	$N_x\times N_z$
conv_reg	$\tilde{\mathcal{F}}_{BEV}$	Conv (3×3)×2	64	$N_x\times N_z$
bbox_reg	*conv_reg*	Conv (3×3)	6×7	$N_x\times N_z$
2D Detection Head				
conv_cls	*fpn i* (i=0, 1, 2, 3, 4)	Conv (3×3)×4	64	$^{H}/_{4\cdot 2^i}\times^{W}/_{4\cdot 2^i}$
bbox_cls	*conv_cls*	Conv (3×3)	3	$^{H}/_{4\cdot 2^i}\times^{W}/_{4\cdot 2^i}$
conv_reg	*fpn i* (i=0, 1, 2, 3, 4)	Conv (3×3)×4	64	$^{H}/_{4\cdot 2^i}\times^{W}/_{4\cdot 2^i}$
bbox_reg	*conv_reg*	Conv (3×3)	3×4	$^{H}/_{4\cdot 2^i}\times^{W}/_{4\cdot 2^i}$
bbox_centerness	*conv_reg*	Conv (3×3)	1	$^{H}/_{4\cdot 2^i}\times^{W}/_{4\cdot 2^i}$

Table 1. Detailed network structure of our stereo-based 3D detection network. By default, the convolution layers in the 2D feature extraction module and the 2D head module are followed by batch normalization layers, and the other convolution layers are attached with group normalization layers (the number of groups is set to 32).

Output	Input	Module Config	#Channel	Size
Sparse 3D Convolution Backbone				
conv1	input	SpConv (3×3×3)	16	$4N_x\times 2N_y\times 4N_z$
conv2		SpConv (3×3×3)×3, s=2	32	$2N_x\times N_y\times 2N_z$
conv3		SpConv (3×3×3)×3, s=2	64	$N_x\times^{N}y/_{2}\times N_z$
conv4		SpConv (3×3×3)×3, s=1×2×1	64	$N_x\times^{N}y/_{4}\times N_z$
$\mathcal{V}_{3d}$		SpConv (1×1×1)	32	$N_x\times^{N}y/_{4}\times N_z$
$\mathcal{F}_{BEV}$		Reshape; Conv (3×3)	32×$^{N}y/_{4}$, 64	$N_x\times N_z$
BEV Aggregation Network & 3D Detection Head				
Same as the stereo detector, see Table 1				

Table 2. Detailed network structure of the LiDAR detector.

IoU	U-Net	Uni-modal	Car AP_{3D}		
			Easy	Mod	Hard
			76.44	56.73	49.52
✓			78.31	59.17	52.07
✓	✓		78.95	59.72	54.03
✓	✓	✓	81.34	61.35	54.56

Table 3. Ablation studies for the *tricks*.

Supervision	Err. Med. Fg/All (mm)	< 0.2m Fg/All (%)	< 0.4m Fg/All (%)	AP_{3D} (%)
Smooth L1	0.64 / 0.13	61.7 / 37.4	40.0 / 22.6	78.9 / 59.7 / 54.0
L1	**0.61** / 0.10	55.2 / 32.9	35.7 / 20.5	78.3 / 61.3 / 54.2
Hard-assigned	0.63 / 0.094	50.9 / 29.9	32.8 / 18.7	80.1 / 61.2 / 54.5
Gaussian σ=0.2	0.63 / 0.092	**50.5 / 29.5**	**32.3 / 18.2**	78.5 / 61.2 / **55.9**
Gaussian σ=0.4	0.66 / 0.11	55.1 / 32.6	35.4 / 19.7	78.5 / 61.8 / 54.9
Gaussian σ=0.8	0.70 / 0.13	59.5 / 37.1	38.4 / 22.0	75.7 / 58.8 / 51.9
Laplacian λ=0.2	0.64 / 0.10	52.9 / 31.3	33.8 / 19.3	80.7 / **62.3** / 55.3
Laplacian λ=0.4	0.66 / 0.11	55.5 / 33.4	35.6 / 20.3	79.0 / 61.4 / 54.6
Laplacian λ=0.8	0.68 / 0.13	58.9 / 36.9	38.2 / 22.2	77.5 / 59.0 / 52.2
Bilinear (Eq.(6))	0.63 / **0.091**	51.1 / 29.9	33.1 / 18.7	**81.2** / 61.5 / 54.6

Table 4. Comparison between different depth losses. *Depth Err. Med.* denotes the average median of depth errors. Foreground (*Fg* in the Table) metrics are evaluated by averaging object-level results, where boxes with less than 5 ground-truth LiDAR points are ignored.

attached with one anchor box. The anchor box sizes for each level are set to $\{32, 64, 128, 256, 512\}$. Please refer to the original paper of ATSS [8] for details.

1.2. The Structure of the LiDAR Detector

The main structure of the LiDAR teacher detector follows that of SECOND [7] with minor modifications. We utilize the same BEV aggregation network and 3D detection head as the stereo detector. Detailed network structure is described in Table 2.

2. More Ablation Studies

2.1. Influence of the *trick* modifications

To improve the performance of the baseline model DSGN [2], we made several important modifications for both training stability and detection performance. The modifications include adding extra IoU-based regression loss, constructing stereo volume using full-resolution 2D feature, and replacing soft-argmin [1] with the uni-modal depth supervision loss. The ablation results are shown in Table 3.

The effectiveness of the IoU regression Loss. In previous stereo-based 3D detector implementations, the loss coefficients for regressing locations, orientations, and sizes are usually the same. Although tuning these coefficients has the potential to improve performance, it is trivial and not adaptable. The IoU regression loss [10, 4] instead directly maximizes the 3D IoU between predictions and targets, which can implicitly adapt the regression coefficients. Our results in Table 3 show that the IoU loss can improve about 2.5% mAP for cars.

2D Detection Network	*pt*	AP_{2D}		
		Car	Ped	Cyc
ResNet-34 [3]	✓	88.3	66.4	49.4
ResNet-34 [3] w/o MaxPool	✓	93.2	71.9	58.3
Ours (2D only)		88.6	51.7	36.3
Ours (2D only)	$*$	91.6	54.3	41.6
Ours (2D only)	✓	93.2	69.2	59.2
Ours (Full model)		90.8	51.9	34.3
Ours (Full model)	$*$	91.3	60.2	47.2

Table 5. Ablation studies for 2D detection head. $*$ only loads pre-trained weights in *conv1*, *conv2*, and *conv3*.

Construct stereo volume with high-resolution features. In PSMNet [1] and DSGN [2], the stereo volume is constructed using left-right image features of $1/4$ size. However, for stereo detection, high-resolution features are essential to improve the depth estimation precision, especially for distant objects. Inspired by the observation, we append an extra small upsampling network (*U-Net*) to construct full-resolution features from SPP features. From Table 3, the *U-Net* improves the 3D detection performance of *moderate* samples by 0.6% mAP and *hard* samples by 2.0% mAP.

Depth Supervision Loss. According to [9], indirectly learning cost volume by soft-argmin and smooth-L1 loss is prone to overfitting since the cost volume is under constrained. In comparison, directly minimizing Kullback–Leibler divergence between the predicted distribution and the unimodal distribution centered at true disparities provides stronger constraints to the cost volume, which can learn more robust implicit depth features $\tilde{\mathcal{V}}_{st}$. Since the ground-truth distribution is constant, the KL divergence can be simplified as cross entropy loss with soft targets,

$$\mathcal{L}_{depth} = \frac{1}{N_{gt}} \sum_{u,v} \sum_{w} \left[-p^*_w \log \mathcal{P}_{st}(u, v, w) \right], \quad (3)$$

where p^*_w is the ground-truth distribution centered at true disparity d^*. Here we investigate several variants of ground-truth distributions, including the bilinearly interpolated distribution (Eq. (6) in the paper), hard-assigned distribution (p_w=1 if $d(w)$ is closest to d^*), gaussian distribution ($p_w \propto \exp\left(-\frac{1}{2}(\frac{d(w)-d^*}{\sigma})^2\right)$), and laplacian distribution ($p_w \propto \exp\left(-\frac{|d(w)-d^*|}{\lambda}\right)$). The results are shown in Table 4. To evaluate local depth embeddings, instead of using global soft-argmin [1] to parse depth values from depth distributions, we employ local soft-argmin to predict the final depth,

$$\tilde{d}_{u,v} = \sum_{w=k-2}^{k+2} d(w) \cdot \frac{\mathcal{P}_{st}(u, v, w)}{\sum_{w'=k-2}^{k+2} \mathcal{P}_{st}(u, v, w')}, \quad (4)$$

Figure 1. Visualization results of the KITTI *validation* set. The green boxes are ground-truth 3D / BEV bounding boxes. The blue boxes are our predictions. The numbers around the ground-truth BEV boxes are the IoU values of their best predictions. The IoU values will be zero if the corresponding 3D boxes are not detected.

where k=argmax$(\mathcal{P}_{st}(u, v, :))$ is the depth index with the maximum probability. Local soft-argmin can avoid the influence of the probability values that are far from the peak probability, which can be utilized to evaluate the local geometric accuracy of the implicit stereo embeddings $\tilde{\mathcal{V}}_{st}$. Results in Table 4 show that ground-truth distributions p^*_w

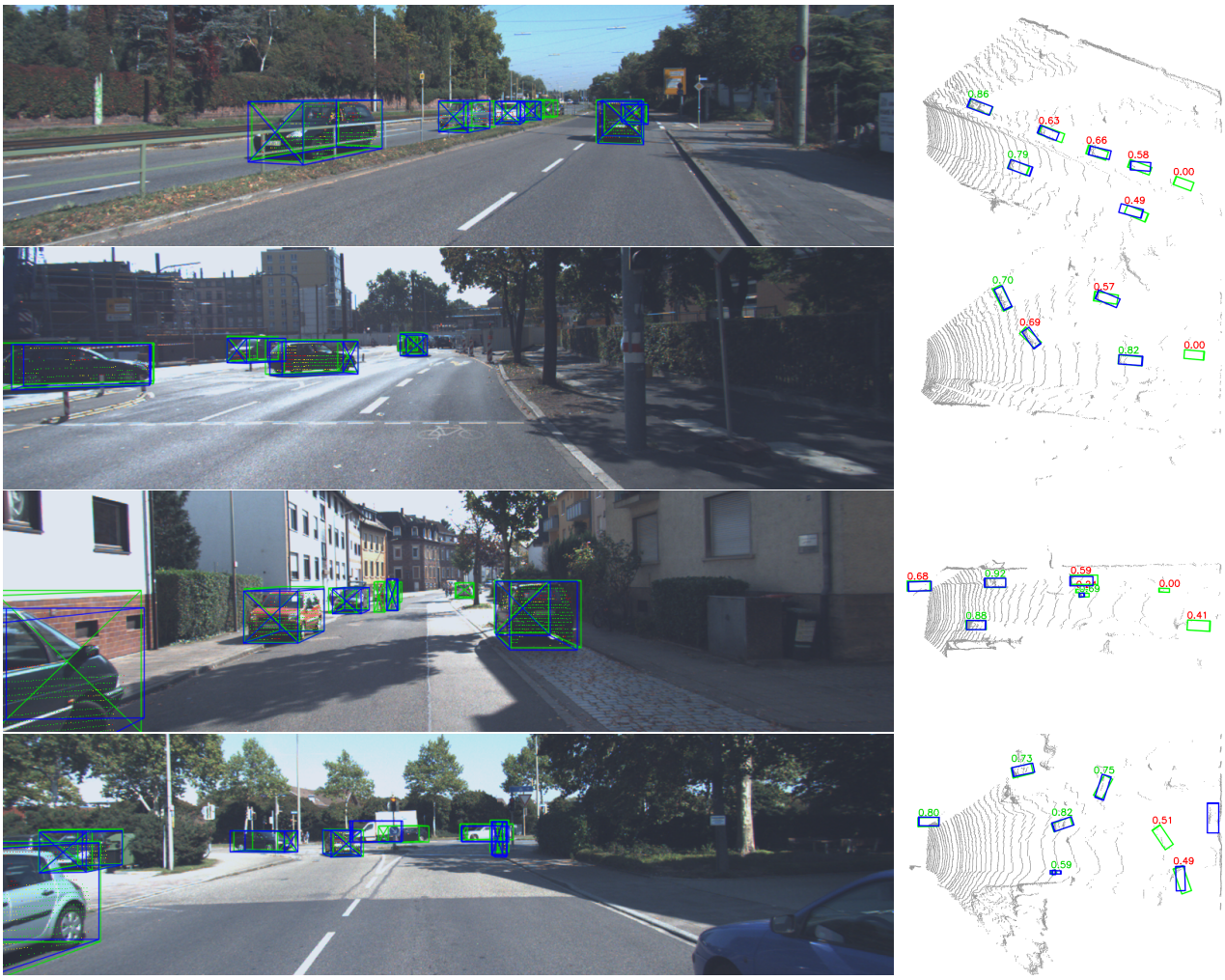

Figure 2. Failure Cases.

that are sharper and more concentrated around d^* can give better results. The choice of distribution encoding methods is not essential, and hard-assigned distribution can even give better performance than L1 loss. The good choices include hard-assigned distribution, gaussian distribution with σ=0.2, laplacian distribution with λ=0.2, and bilinearly interpolated distribution.

2.2. 2D Detection Performance

We compare our 2D detection branch with ResNet-34 [3] to confirm that our semantic bottleneck $\mathcal{F}_{sem}$ does not constrain the performance of 2D detection. Since our model does not employ max pooling after *conv1*, we give the results of ResNet-34 without max pooling in the second row of Table 5 for fair comparison. By comparing the results of *ResNet-34 w/o Maxpool* and *Ours (2D only)*, both models achieve similar performance given ImageNet pretrained weights, which proves that the semantic bottleneck does not constrain the 2D detection performance and has the capability of learning good semantic features. By comparing the models without and with ImageNet pretrained weights, we can see pretrained weights are essential for 2D detection due to the limit of training data in the KITTI dataset.

3. Visualization Results

Please see the visualization results in Fig. 1. Most of the objects can be detected with high IoU successfully, even for distant objects. We also visualize several failure cases in Fig. 2. Most of the failure cases are caused by occlusions and depth estimation errors. Several predictions give large orientation errors, which we believe can be fixed by incorporating predictions of 2D key-points of bounding boxes in the future.

References

[1] Jia-Ren Chang and Yong-Sheng Chen. Pyramid stereo matching network. In *CVPR*, pages 5410–5418, 2018. 1, 3

[2] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Dsgn: Deep stereo geometry network for 3d object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 3, 5

[4] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 3

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[6] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. https://github.com/open-mmlab/OpenPCDet, 2020. 1

[7] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 3

[8] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 1, 3

[9] Youmin Zhang, Yimin Chen, Xiao Bai, Suihanjin Yu, Kun Yu, Zhiwei Li, and Kuiyuan Yang. Adaptive unimodal cost volume filtering for deep stereo matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12926–12934, 2020. 3

[10] Dingfu Zhou, Jin Fang, Xibin Song, Chenye Guan, Junbo Yin, Yuchao Dai, and Ruigang Yang. Iou loss for 2d/3d object detection. In *2019 International Conference on 3D Vision (3DV)*, pages 85–94. IEEE, 2019. 3