Unleash the Potential of Image Branch for Cross-modal 3D Object Detection (*Supplementary Materials*)

A Appendix

In this appendix, we provide the details omitted from the manuscript due to space limitation. We organize the appendix as follows.

- Section A.1: Implementation details.
- Section A.2: More quantitative results.
- Section A.3: Experimental results on Waymo dataset.
- Section A.4: More ablation studies.
- Section A.5: Efficiency analysis.
- Section A.6: Visual results of 3D object detection.
- Section A.7: Visual results of 2D semantic segmentation.
- Section A.8: Details on the official KITTI test leaderboard.

A.1 Implementation Details

Network Architecture. Fig. 1 illustrates the architectures of the point cloud and image backbone networks. For the encoder of the point cloud branch, we further show the details of multi-scale grouping (MSG) network in Table 1. Following 3DSSD [23], we take key points sampled by the 3^{rd} SA layer to generate vote points and estimate the 3D box. Then we feed these 3D boxes and features output by the last FP layer to the refinement stage. Besides, we adopt density-aware RoI grid pooling [6] to encode point density as an additional feature. Note that 3D center estimation [5] aims to learn the relative position of each foreground point to the object center, while the 3D box is estimated based on sub-sampled points. Thus, the auxiliary task of 3D center estimation differs from 3D box estimation and can facilitate learning structure-aware features of objects.

Table 1: Details of set abstraction layers in the point cloud branch. We report the sampling strategy used in the sampling operation, ball radius of group operation, "nquery" that denotes the number of group points, and dimensions of the unit PointNet layer for multi-scale grouping. The features at different scales are concatenated and dimensionally reduced to the specific output channels.

Layer	Sampling Strategy	Radius	nquery	Feature Dimension	Output Channels
1^{st} SA	D-FPS	[0.2, 0.4, 0.8]	[32, 32, 64]	[[16, 16, 32], [16, 16, 32], [32, 32, 64]]	64
2^{nd} SA	D-FPS & S-FPS	[0.4, 0.8, 1.6]	[32, 32, 64]	[[64, 64, 128], [64, 64, 128], [64, 96, 128]]	128
3^{rd} SA	D-FPS & S-FPS	[1.6, 3.2, 4.8]	[64, 64, 128]	[[128, 128, 256], [128, 196, 256], [128, 256, 256]]	256

Training Details. Through the experiments on KITTI dataset, we adopted Adam [8] (β_1=0.9, β_2=0.99) to optimize our BiProDet. We initialized the learning rate as 0.003 and updated it with the one-cycle policy [18]. And we trained the model for a total of 80 epochs in an end-to-end manner. In our experiments, the batch size was set to 8, equally distributed on 4 NVIDIA 3090 GPUs. We kept the input image with the original resolution and padded it to the size of 1248×376, and down-sampled the input point cloud to 16384 points during training and inference. Following the common practice, we set the detection range of the x, y, and z axis to [0m, 70.4m], [-40m, 40m] and [-3m, 1m], respectively.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

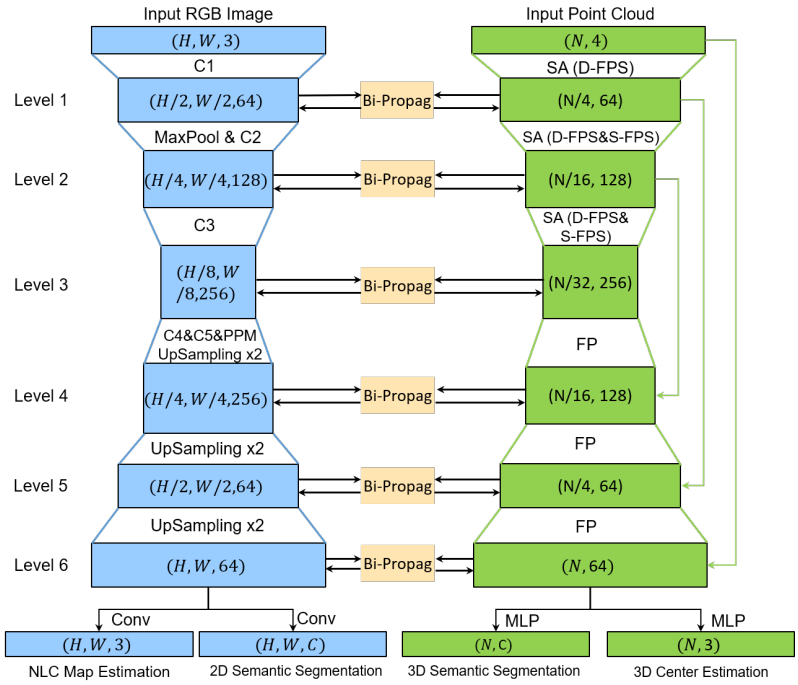

Figure 1: The detailed architecture of 2D and 3D backbones. We adopt ResNet18 as the encoder of the image branch, followed by a decoder with Pyramid Pooling Module (PPM) and several up-sampling blocks. C1, C2, C3, C4, and C5 denote convolutional layers of different stages in ResNet. Extra convolutional layers are deployed after each up-sampling layer. For the point cloud branch, we adopt the PointNet++ structure. SA: set abstraction layer, D-FPS: 3D Euclidean distance-based farthest point sampling, S-FPS: semantic-guided farthest point sampling, FP: feature propagation layer, MLP: shared multi-layer perceptron. Besides, "Bi-Propag" denotes the proposed bidirectional feature propagation between the 2D and 3D backbones.

Data Augmentation. We applied common data augmentation strategies at global and object levels. The global-level augmentation includes random global flipping, global scaling with a random scaling factor between 0.95 and 1.05, and global rotation around the z-axis with a random angle in the range of $[-\pi/4, \pi/4]$. Each of the three augmentations was performed with a 50% probability for each sample. The object-level augmentation refers to copying objects from other scenes and pasting them to current scene [22]. In order to perform sampling synchronously on point clouds and images, we utilized the instance masks provided in [14]. Specifically, we pasted both the point clouds and pixels of sampled objects to the point cloud and images of new scenes, respectively.

A.2 More quantitative Results

Performance on KITTI Val Set. We also reported the performance of our BiProDet on all three classes of the KITTI validation set in Table 2, where it can be seen that our BiProDet also achieves the highest mAP of 77.73%, which is obviously higher than the second best method CAT-Det.

Performance of Single-Class Detector. Quite a few methods [4, 29] train models only for car detection. Empirically, the single-class detector performs better in the car class compared with multi-class detectors. Therefore, we also provided performance of BiProDet trained only for the car class, and compared it with several state-of-the-art methods in Table 3.

Generalization to Asymmetric Backbones. As shown in Fig. 1, we originally adopted an encoder-decoder network in the LiDAR branch that is architecturally similar to the image backbone. Nevertheless, it is worth clarifying that our approach is not limited to symmetrical structures and can be generalized to different point-based backbones. Here, we replaced the 3D branch of the original framework with an efficient single-stage detector—SASA [2], using a backbone only with the encoder in the LiDAR branch, which is asymmetric with the encoder-decoder structure of the image backbone.

Table 2: Quantitative comparisons on the KITTI validation set under the evaluation metric of 3D Average Precision (AP) calculated with 11 sampling recall positions. We highlight the best and the second best results in bold and underlined, respectively.

Method	Modality	3D Car (IoU=0.7)			3D Ped. (IoU=0.5)			3D Cyc. (IoU=0.5)			**mAP**
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
PointPillars [10]	LiDAR	86.46	77.28	74.65	57.75	52.29	47.90	80.05	62.68	59.70	66.53
SECOND [22]	LiDAR	88.61	78.62	77.22	56.55	52.98	47.73	80.58	67.15	63.10	68.06
3DSSD [23]	LiDAR	88.55	78.45	77.30	58.18	54.31	49.56	86.25	70.48	65.32	69.82
PointRCNN [15]	LiDAR	88.72	78.61	77.82	62.72	53.85	50.24	86.84	71.62	65.59	70.67
PV-RCNN [16]	LiDAR	89.03	<u>83.24</u>	78.59	63.71	57.37	52.84	86.06	69.48	64.50	71.65
TANet [11]	LiDAR	88.21	77.85	75.62	70.80	63.45	58.22	85.98	64.95	60.40	71.72
Part-A^2 [17]	LiDAR	89.55	79.40	78.84	65.68	60.05	55.44	85.50	69.90	65.48	72.20
AVOD-FPN [9]	LiDAR+RGB	84.41	74.44	68.65	-	58.80	-	-	49.70	-	-
PointFusion [21]	LiDAR+RGB	77.92	63.00	53.27	33.36	28.04	23.38	49.34	29.42	26.98	42.75
F-PointNet [13]	LiDAR+RGB	83.76	70.92	63.65	70.00	61.32	53.59	77.15	56.49	53.37	65.58
CLOCs [12]	LiDAR+RGB	89.49	79.31	77.36	62.88	56.20	50.10	87.57	67.92	63.67	70.50
EPNet [7]	LiDAR+RGB	88.76	78.65	78.32	66.74	59.29	54.82	83.88	65.50	62.70	70.96
CAT-Det [26]	LiDAR+RGB	**90.12**	81.46	<u>79.15</u>	**74.08**	<u>66.35</u>	<u>58.92</u>	<u>87.64</u>	<u>72.82</u>	<u>68.20</u>	<u>75.42</u>
BiProDet (Ours)	LiDAR+RGB	<u>89.73</u>	**86.40**	**79.31**	<u>71.77</u>	**68.49**	**62.52**	**89.24**	**76.91**	**75.18**	**77.73**

Table 3: Comparison with state-of-the-art methods on the KITTI val set for car 3D detection. All results are reported by the average precision with 0.7 IoU threshold. $\mathrm{R11}$ and $\mathrm{R40}$ denotes AP calculated with 11 and 40 recall sampling recall points, respectively.

Method	Modal	$\mathrm{AP}_{\mathrm{3D\|R11}}$ (%)			$\mathrm{AP}_{\mathrm{3D\|R40}}$ (%)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
Voxel R-CNN [4]	LiDAR	89.41	84.52	78.93	92.38	85.29	82.86
PV-RCNN [16]	LiDAR	89.35	83.69	78.7	92.57	84.83	82.69
SA-SSD [5]	LiDAR	90.15	79.91	78.78	93.14	84.65	81.86
SE-SSD [29]	LiDAR	**90.21**	85.71	79.22	93.19	86.12	83.31
GLENet-VR [27]	LiDAR	89.93	86.46	79.19	**93.51**	86.10	83.60
MV3D [3]	LiDAR+RGB	-	-	-	71.29	62.68	56.56
3D-CVF [24]	LiDAR+RGB	-	-	-	89.67	79.88	78.47
BiProDet (Ours)	LiDAR+RGB	89.72	**86.52**	**79.34**	93.03	**86.46**	**83.73**

Accordingly, the proposed bidirectional propagation is only performed between the 3D backbone and the encoder of the 2D image backbone. The experimental results are shown in Table 4. We can observe that the proposed method works well even when the two backbones are asymmetric, which demonstrates the satisfactory generalization ability of our method for different LiDAR backbones.

Table 4: Bidirectional propagation is also effective when the 2D and 3D backbones are asymmetric. Here we adopt a single-stage detector [2] whose backbone includes only an encoder, which is asymmetric with the encoder-decoder network in the image branch.

Method	3D Car (IoU=0.7)			3D Ped. (IoU=0.5)			3D Cyc. (IoU=0.5)			**mAP**
	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
SASA	92.17	84.90	82.57	66.75	61.40	56.00	89.91	74.05	69.41	75.24
Ours (SASA)	92.11	85.67	82.99	70.52	63.38	58.16	91.58	74.81	70.22	76.61

A.3 Results on Waymo Open Dataset

The Waymo Open Dataset [19] is a large-scale dataset for 3D object detection. It contains 798 sequences (15836 frames) for training, and 202 sequences (40077 frames) for validation. According to the number of points inside the object and the difficulty of annotation, the objects are further divided into two difficulty levels: LEVEL_1 and LEVEL_2. Following common practice, we adopted the metrics of mean Average Precision (mAP) and mean Average Precision weighted by heading accuracy (mAPH), and reported the performance on both LEVEL_1 and LEVEL_2. We set the detection range to [-75.2m, 75.2m] for x and y axis, and [-2m, 4m] for z axis. Following [20] and [1], the training on Waymo dataset consists of two stages to allow flexible augmentations. First, we only trained the LiDAR branch without image inputs and bidirectional propagation for 30 epochs. We enabled the copy-and-paste augmentation in this stage. Then, we trained the whole pipeline

Table 5: 3D detection results on the Waymo Open Dataset validation set. "-" denotes that the results are not reported in their papers.

Method	Vehicle L1		Vehicle L2		Pedestrian L1		Pedestrian L2		Cyclist L1		Cyclist L2	
	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH	mAP	mAPH
PointAugmenting	67.41	-	62.7	-	75.42	-	70.55	-	76.29	-	74.41	-
TransFusion	-	-	-	65.14	-	-	-	64.00	-	-	-	67.40
Ours	78.36	77.91	69.45	69.04	76.32	71.67	65.93	61.81	79.64	78.55	76.36	75.26

for another 6 epochs, during which the copy-and-paste is disabled. Note that the image semantic segmentation head is disabled, since ground-truth segmentation maps are not provided [19].

As shown in Table 5, our method achieves substantial improvement compared with previous state-of-the-arts. Particularly, unlike existing approaches including PointAugmenting [20] and TransFusion [1] where the camera backbone is pre-trained on other datasets and then frozen, we trained the entire pipeline in an end-to-end manner. It can be seen that even without the 2D segmentation auxiliary task, our method still achieves higher accuracy under all scenarios except "Ped L2", demonstrating its advantage.

A.4 More Ablation Studies

Table 6: Effect of the semantic-guided SA layer. Compared with the single-modal baseline, BiProDet can better exploit image semantics and preserve more foreground points during downsampling.

Method	Single-Modal		BiProDet (Ours)		*Improvement*	
SA Layer	FG rate	Instance recall	FG rate	Instance recall	FG rate	Instance recall
Level-2	15.87	97.92	20.70	98.23	*+4.83*	*+0.31*
Level-3	29.73	97.35	38.03	97.82	*+8.29*	*+0.47*

Table 7: Comparison between our multi-task training methods and the single-modal 2D semantic segmentation baseline (PSPNet). The results show that point features effectively improve the segmentation performance on pedestrian and cyclist classes.

Method	Car	Pes.	Cyc.	mIoU
PSPNet [28]	77.49	30.45	23.83	43.92
BiProDet (Ours)	78.45	36.15	30.42	48.34

Effect of Semantic-guided Point Sampling. When performing downsampling in the SA layers of the point cloud branch, we adopted S-FPS [2] to explicitly preserve as many foreground points as possible. We report the percentage of sampled foreground points and instance recall (i.e., the ratio of instances that have at least one point) in Table 6, where it can be seen that exploiting supplementary semantic features from images leads to substantial improvement of the ratio of sampled foreground points and better instance recall during S-FPS.

Influence on 2D Semantic Segmentation. We also aimed to demonstrate that the 2D-3D joint learning paradigm benefits not only the 3D object detection task but also the 2D semantic segmentation task. As shown in Table 7, the deep interaction between different modalities yields an improvement of 4.42% mIoU. The point features can naturally complement RGB image features by providing 3D geometry and semantics, which are robust to illumination changes and help distinguish different classes of objects, for 2D visual information. The results suggest the potential of joint training between 3D object detection and more 2D scene understanding tasks in autonomous driving.

Conditional Analysis. To better figure out where the improvement comes from when using additional image features, we compared BiProDet with the single-modal detector on different occlusion levels and distant ranges. The results shown in Table 8 and Table 9 include separate APs for objects belonging to different occlusion levels and APs for moderate class in different distance ranges. For car detection, our BiProDet achieves more accuracy gains for long-distance and highly occluded objects, which suffer from the sparsity of observed LiDAR points. The cyclist and pedestrian are much more difficult categories on account of small sizes, non-rigid structures, and fewer training samples. For these two categories, BiProDet still brings consistent and significant improvements on different levels even in extremely difficult cases.

Table 8: Performance breakdown over different occlusion levels. As defined by the official website of KITTI, occlusion levels 0, 1, and 2 correspond to fully-visible samples, partly-occluded samples, and samples that are difficult to see, respectively.

Class	Car			Pedestrian			Cyclist		
Occlusion	Level-0	Level-1	Level-2	Level-0	Level-1	Level-2	Level-0	Level-1	Level-2
Single-Modal	91.98	77.18	55.41	67.44	26.76	6.10	91.23	24.66	1.74
BiProDet (Ours)	92.26	77.44	58.39	74.00	35.13	7.99	92.89	30.02	2.53
Improvement	+0.28	+0.26	+2.97	+6.56	+8.38	+1.89	+1.66	+5.36	+0.79

Table 9: Performance breakdown over different distances.

Class	Car			Pedestrian			Cyclist		
Distance	0-20m	20-40m	40m-Inf	0-20m	20-40m	40m-Inf	0-20m	20-40m	40m-Inf
Single-Modal	96.28	85.21	43.91	71.28	38.42	1.63	93.61	61.56	34.48
BiProDet (Ours)	96.36	86.48	49.88	76.56	45.72	2.46	94.12	67.27	39.10
Improvement	+0.07	+1.27	+5.97	+5.28	+7.30	+0.83	+0.51	+5.71	+4.62

Generalization to Sparse LiDAR Signals. We also compared our BiProDet with the single-modal baseline on LiDAR point clouds with various sparsity. In practice, following Pseudo-LiDAR++ [25], we simulated the 32-beam, 16-beam, and 8-beam LiDAR signals by selecting LiDAR points whose elevation angles fall within specific intervals. As shown in Table 10, the proposed BiProDet outperforms the single-modal baseline under all settings. The consistent improvements suggest our method can generalize to sparser signals. Besides, the proposed BiProDet significantly performs better than the baseline in the setting of LiDAR signals with fewer beams, demonstrating the effectiveness of our method in exploiting the supplementary information in the image domain.

Table 10: Comparison with single-modal baselines under LiDAR signals with different beams, where we report $AP_{3D|R40}$ on the KITTI validation set.

LiDAR Beams	Modal	Car	Ped.	Cyc.	**mAP**
64	LiDAR	86.71	62.23	78.68	75.87
	LiDAR + RGB	87.18	67.52	81.21	78.64
	Improvement	+0.47	+5.29	+2.53	+2.76
32	LiDAR	83.49	57.83	70.82	70.71
	LiDAR + RGB	84.47	62.56	73.93	73.65
	Improvement	+0.98	+4.73	+3.11	+2.94
16	LiDAR	79.80	53.84	60.51	64.71
	LiDAR + RGB	80.78	59.44	67.35	69.19
	Improvement	+0.99	+5.60	+6.84	+4.48
8	LiDAR	64.42	22.57	43.19	43.39
	LiDAR + RGB	67.09	31.03	47.97	48.70
	Improvement	+2.66	+8.46	+4.78	+5.30

Robustness against Input Corruption. We also conducted extensive experiments to verify the robustness of our BiProDet to sensor perturbation. Specifically, we added Gaussian noises to the reflectance value of points or RGB images. Fig. 2 shows that the mAP value of our cross-modal BiProDet is consistently higher than that of the single-modal baseline and decreases slower with the LiDAR noise level increasing. Particularly, as listed in Table 11, when the variance of the LiDAR noise is set to 0.15, the perturbation affects our cross-modal BiProDet much less than the single-modal detector. Besides, even applying the corruption to both LiDAR input and RGB images, the mAP value of our BiProDet only drops by 2.49%.

Table 11: Performances (mAPs) of the single-modal baseline and our BiProDet on the KITTI val set under input corruptions of simulated LiDAR and image noise sampled from the Gaussian distribution. Note that the image noise is only applicable to multi-modal detectors.

Corruptions Type	Modal	Car	Ped.	Cyc.	**mAP**
No Corruption	LiDAR	86.71	62.23	78.68	75.87
LiDAR Noise	LiDAR	84.37	49.19	77.27	70.28
No Corruption	LiDAR + RGB	87.18	67.52	81.21	78.64
LiDAR Noise	LiDAR + RGB	86.82	65.61	77.63	76.69
Image Noise	LiDAR + RGB	87.14	66.26	77.97	77.12
LiDAR + Image Noise	LiDAR + RGB	86.60	65.42	76.44	76.15

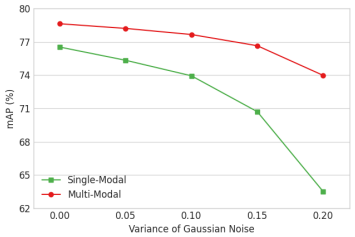

Figure 2: Comparisons of noise robustness between the single-modal baseline and our BiProDet.

Effectiveness of Multi-stage Interaction. As mentioned before, both 2D and 3D backbones adopt an encoder-decoder structure, and we perform bidirectional feature propagation at both downsampling and upsampling stages. Here, we conducted experiments to verify the superiority of the multi-stage interaction over single-stage interaction. As shown in Table 12, only performing the bidirectional feature propagation in the encoder (i.e., Table 12 (b)) or the decoder (i.e., Table 12 (c)) leads to worse performance than that of performing the module in both stages (i.e., Table 12 (d)).

Table 12: Ablative experiments on the multi-stage manner of bidirectional propagation, where SA and FP denote applying bidirectional propagation at downsampling (in the encoder) and upsampling (in the decoder) stages of the point cloud branch, respectively.

	Stage		3D Car (IoU=0.7)			3D Ped. (IoU=0.5)			3D Cyc. (IoU=0.5)			**mAP**
	SA	FP	Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
(a)	-	-	91.92	85.22	82.98	68.82	61.47	56.39	91.93	74.56	69.58	75.88
(b)	✓	-	92.67	85.86	83.40	70.94	65.29	60.35	93.60	75.60	71.04	77.64
(c)	-	✓	92.19	85.44	83.27	69.02	63.41	58.47	92.85	76.39	71.91	76.99
(d)	✓	✓	92.63	85.77	83.13	72.68	67.64	62.25	94.39	77.77	71.47	78.64

A.5 Efficiency Analysis

We also compared the inference speed and number of parameters of the proposed BiProDet with state-of-the-art cross-modal approaches in Table 13. Our BiProDet has about the same number of parameters as CAT-Det [26], but a much higher inference speed at 9.52 frames per second on a single GeForce RTX 2080 Ti GPU. In general, our BiProDet is inevitably slower than some single-modal detectors, but it achieves a good trade-off between speed and accuracy among cross-modal approaches.

Table 13: Comparison of the number of network parameters, inference speed, and detection accuracy of different multi-modal methods on the KITTI test set.

Method	Params (M)	Frames per second	mAP (%)
AVOD-FPN [9]	38.07	10.00	56.84
F-PointNet [13]	12.45	6.25	57.86
EPNet [7]	16.23	5.88	-
CAT-Det [26]	23.21	3.33	67.05
BiProDet (Ours)	24.98	9.52	70.13

A.6 Visual Results of 3D Object Detection

In Figure 3, we present the qualitative comparison of detection results between the single-modal baseline and our BiProDet. We can observe that the proposed BiProDet shows better localization capability than the single-modal baseline in challenging cases. Besides, we also show qualitative results of BiProDet on the KITTI test split in Figure 4. We can clearly observe that our BiProDet performs well in challenging cases, such as pedestrians and cyclists (with small sizes) and highly-occluded cars.

A.7 Visual Results of 2D Semantic Segmentation

Several examples are shown in Figure 5. For distant cars in the first and the second row as well as the pedestrian in the sixth row, the size of objects is small and PSPNet tends to treat them as background, while our BiProDet is able to correct such errors. In the third row, our BiProDet finds the dim cyclist missed by PSPNet. Our BiProDet also performs better for the highly occluded objects as shown in the fourth and the fifth lines. This observation shows the 3D feature representations extracted from point clouds can boost 2D semantic segmentation, since the image-based method is sensitive to illumination and can hardly handle corner cases with only single-modal inputs.

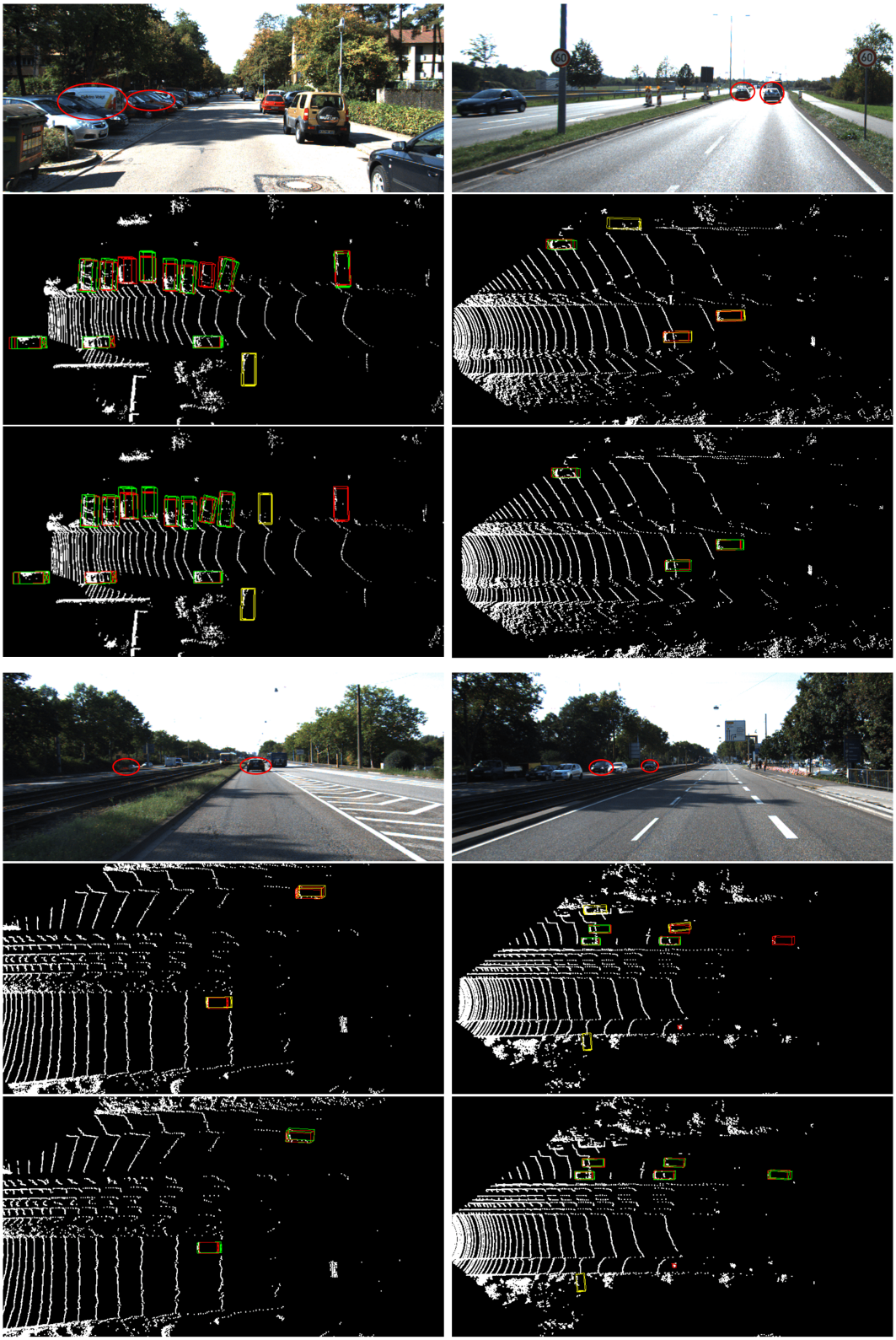

Figure 3: Qualitative comparison between single-modal baseline and our multi-modal BiProDet. For each comparison, from top to bottom, we have the image, detection results of single-modal baseline, and detection results of BiProDet. We use red, green, and yellow to denote the ground-truth, true positive and false positive bounding boxes, respectively. We highlight some objects in images with red circles, which are detected by BiProDet but missed by the single-modal method.

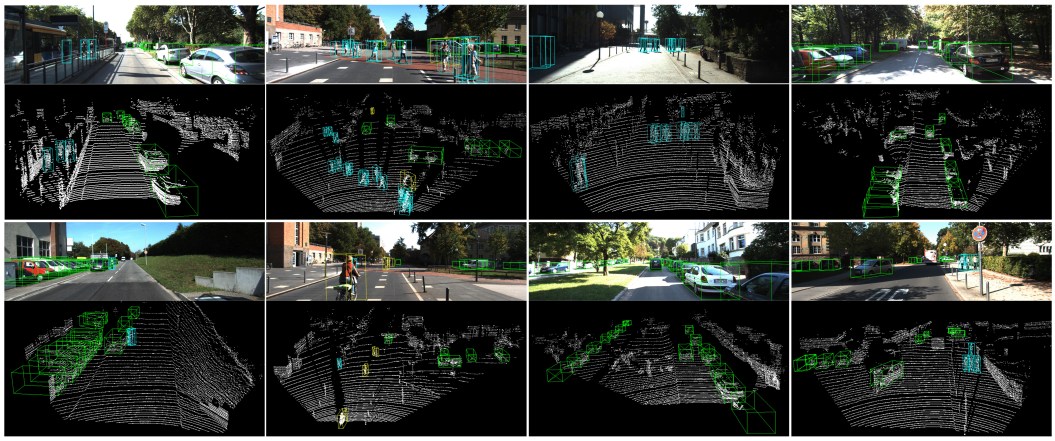

Figure 4: Extra qualitative results of BiProDet on the KITTI test set. The predicted bounding boxes of car, pedestrian, and cyclist are visualized in green, cyan, and yellow, respectively. We also show the corresponding projection of boxes on images. Best viewed in color and zoom in for more details.

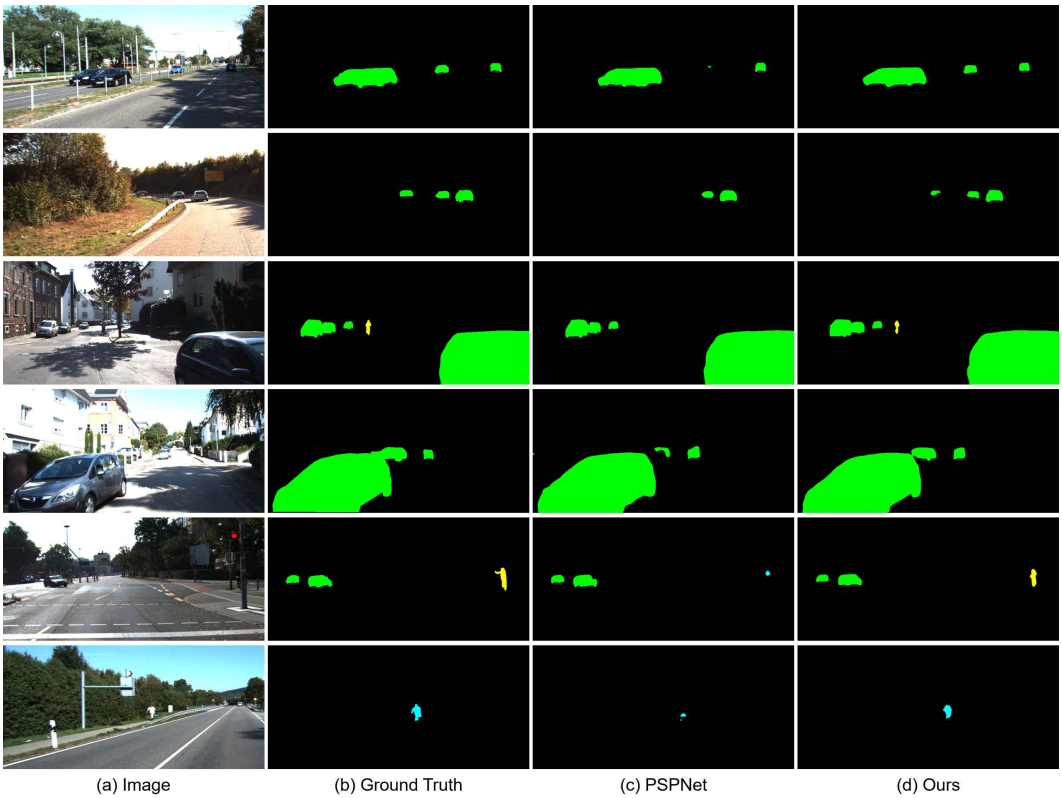

Figure 5: Visual results of 2D semantic segmentation on the KITTI val set. The prediction boxes are shown in green for car, cyan for pedestrian, and yellow for cyclist. Best viewed in color. Compared with PSPNet, our cross-modal BiProDet produces more accurate and detailed results.

A.8 Details on Official KITTI Test Leaderboard

We submitted the results of our BiProDet to the official KITTI website, and it ranks $\mathbf{1}^{st}$ on the 3D object detection benchmark for the cyclist class. Figure 6 shows the screenshot of the leaderboard. Figure 7 illustrates the precision-recall curves along with AP scores on different categories of the KITTI test set. The samples of the KITTI test set are quite different from those of training/validation set in terms of scenes and camera parameters, so the impressive performance of our BiProDet on the test set demonstrates it also achieves good generalization.

	Method	Setting	Code	Moderate	Easy	Hard	Runtime	Environment
1	BiProDet			**74.32 %**	86.74 %	**67.45 %**	0.1 s	GPU @ 2.5 Ghz (Python + C/C++)
2	TED			74.12 %	**88.82 %**	66.84 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
3	CasA++		code	73.79 %	87.76 %	66.84 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
4	CasA		code	73.47 %	87.91 %	66.17 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
5	SGNet			70.40 %	86.75 %	62.73 %	0.09 s	GPU @ 2.5 Ghz (Python)
6	HMFI		code	70.37 %	84.02 %	62.57 %	0.1 s	1 core @ 2.5 Ghz (C/C++)
7	CAD			69.94 %	84.68 %	62.21 %	0.1 s	GPU @ 2.5 Ghz (Python + C/C++)
8	SARFE			69.67 %	84.88 %	62.26 %	0.03 s	1 core @ 2.5 Ghz (C/C++)
9	EQ-PVRCNN		code	69.10 %	85.41 %	62.30 %	0.2 s	GPU @ 2.5 Ghz (Python + C/C++)
10	VoCo			69.00 %	82.74 %	62.46 %	0.1 s	1 core @ 2.5 Ghz (Python + C/C++)

Figure 6: Screenshot of the KITTI 3D object detection benchmark for cyclist class on August 15th, 2022.

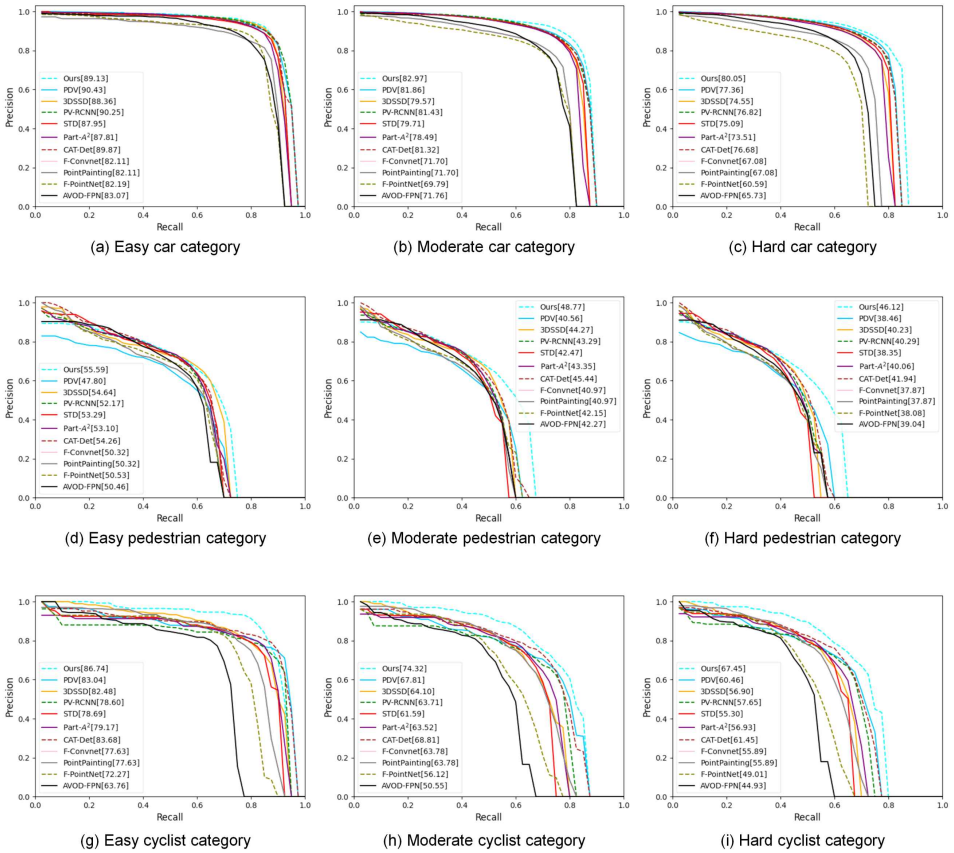

Figure 7: Precision-recall curves of different methods on the KITTI 3D object detection test set on Aug. 15^{th}, 2022. We also report APs in different categories for each method.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1090–1099, 2022.

[2] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 1, pp. 221–229, 2022.

[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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[4] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1201–1209, 2021.

[5] 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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11873–11882, 2020.

[6] Jordan SK Hu, Tianshu Kuai, and Steven L Waslander. Point density-aware voxels for lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8469–8478, 2022.

[7] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *European Conference on Computer Vision*, pp. 35–52. Springer, 2020.

[8] D.P. Kingma and J.L. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, pp. 1–15, 2015.

[9] 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*, pp. 1–8. IEEE, 2018.

[10] A.H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12689–12697, 2019.

[11] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11677–11684, 2020.

[12] Su Pang, Daniel Morris, and Hayder Radha. Clocs: Camera-lidar object candidates fusion for 3d object detection. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 10386–10393. IEEE, 2020.

[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/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018.

[14] Lu Qi, Li Jiang, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Amodal instance segmentation with kins dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3014–3023, 2019.

[15] S. Shi, X. Wang, and H. Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–779, 2019.

[16] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10529–10538, 2020.

[17] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8):2647–2664, 2020.

[18] Leslie N Smith. Cyclical learning rates for training neural networks. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 464–472. IEEE, 2017.

[19] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2443–2451, 2020.

[20] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11794–11803, 2021.

[21] Danfei Xu, Dragomir Anguelov, and Ashesh Jain. Pointfusion: Deep sensor fusion for 3d bounding box estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2018.

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

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

[24] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision*, pp. 720–736. Springer, 2020.

[25] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving. In *International Conference on Learning Representations*, pp. 1–13, 2020.

[26] Yanan Zhang, Jiaxin Chen, and Di Huang. Cat-det: Contrastively augmented transformer for multi-modal 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 908–917, 2022.

[27] Yifan Zhang, Qijian Zhang, Zhiyu Zhu, Junhui Hou, and Yixuan Yuan. Glenet: Boosting 3d object detectors with generative label uncertainty estimation. *arXiv preprint arXiv:2207.02466*, 2022.

[28] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2881–2890, 2017.

[29] Wu Zheng, Weiliang Tang, Li Jiang, and Chi-Wing Fu. Se-ssd: Self-ensembling single-stage object detector from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14494–14503, 2021.