Image-Guided Semantic Pseudo-LiDAR Point Generation for 3D Object Detection

Supplementary Material

A. Additional Experimental Details

A.1. Dense Dataset Construction

To train our network, a dense supervision dataset is required in which ground-truth (GT) objects are represented in a densified or completed form. Following the methodology proposed in BtcDet [16], we construct such a dataset using a heuristic, rule-based strategy that combines objects of similar size to compensate for occluded regions. Furthermore, for the car and cyclist classes, we mirror point clouds under the assumption of bilateral symmetry to enhance data completeness. For the Waymo Open Dataset (WOD) [13], which provides temporal annotations enabling object tracking across frames, we aggregate points from multiple frames corresponding to the same object instance to generate dense representations. Similarly, we assume symmetry for the vehicle and cyclist classes and apply mirroring to further enrich the training data. However, such aggregation methods based on temporal frames are not sufficient to construct complete object shapes, as some parts remain invisible even when multiple frames are aggregated. Therefore, we adopt the approach proposed by BtcDet [16], which is used in the KITTI [3], to construct fully densified object shapes for the Waymo Open Dataset (WOD) [13]. Figure A1 and A2 show the dense dataset construction results for the KITTI [3] and WOD [13] benchmarks.

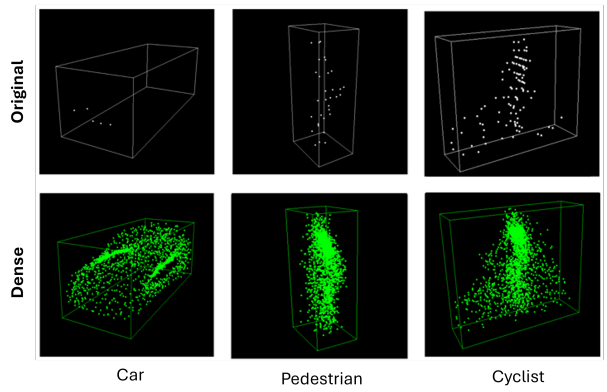

Figure A1. Visualization of the dense point cloud dataset for the KITTI [3] benchmark.

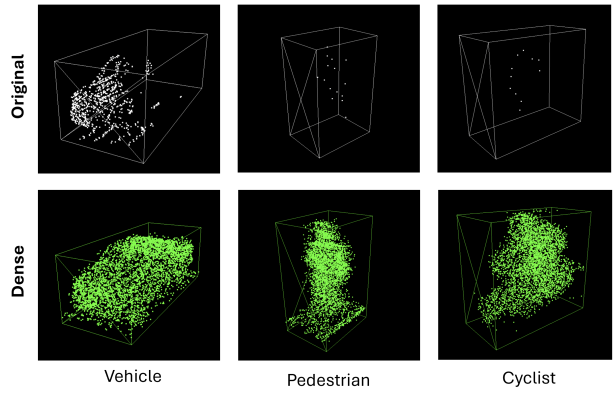

Figure A2. Visualization of the dense point cloud dataset for the WOD [13] benchmark.

A.2. Implementation Details

We apply multiple geometric transformations to the input point clouds to enhance transformation-equivariant feature learning. The number of transformation actions N_t is set to 4 during training and 6 during evaluation for the KITTI [3] dataset. Due to computational cost and VRAM usage for training the whole network, N_t is set to 3 for both training and evaluation on the Waymo Open Dataset (WOD) [13]. Each transformation action is composed of `Flip-X`, `Rotation`, and `Scaling`. `Flip-Y` is additionally used for the WOD [13]. Note that the final transformation action preserves the original point cloud distribution, i.e., no flipping, rotation, or scaling is applied for the last transformation action.

The 2D image backbone employed in ImagePG is Swin Transformer Tiny [10], implemented via OpenPCDet [11, 14]. It is pretrained with Mask R-CNN [4] on the nuImages [1] dataset for 2D object detection and segmentation, enabling the model to leverage prior knowledge of driving scenes [11]. To capture fine-grained features across multiple spatial resolutions, we additionally incorporate a Feature Pyramid Network (FPN) [9], which aggregates multi-scale features from low-resolution to high-resolution representations. Specifically, we utilize the final high-resolution feature map from the FPN as the image feature input for the subsequent modules. To preserve the pretrained semantic knowledge, we freeze the bottom three layers of the four-layer image backbone and enable gradient backpropagation only through the top layer.

We employ Anchor Head [14] as the region proposal network (RPN) for the KITTI [3] dataset and CenterPoint [17] for the Waymo Open Dataset (WOD) [13]. For the IG-RPG module, we utilize a single Transformer Encoder [15] with a hidden dimension of 512. N_s is set to 4096. RoI grid size

Class	Min	Q1	Q2	Q3	Max
Car	0	34	111	377	3573
Pedestrian	0	54	118	289	1171
Cyclist	0	25	47	131	1137

Table A1. Number of points summarization of the KITTI [3] *val* set.

G is set to 6 for convention [2, 12]. Adam optimizer [5] with a one-cycle policy is used. For the KITTI [3], we trained for 80 epochs with an initial learning rate of 0.01. We used 8 NVIDIA RTX A6000 GPUs for a batch size of 16, and the training time was less than 12 hours. For the WOD [13], we trained for 30 epochs with an initial learning rate of 0.01.

Deformable Attention is widely used for 2D and 3D object detection [7, 8, 18] with the model to focus on semantically meaningful regions, even under geometric transformations or sparsity. The deformable attention process can be formulated as follows:

$$\text{DeformAttn}(Q_i, p_i, x) = \sum_{m=1}^{M} W_m \left[\sum_{k=1}^{K} A_{mik} \cdot W'_m \langle x \cdot (p_i + \Delta p_{mik}) \rangle \right], \quad (1)$$

where p_i, Δp_{mik}, A_{mik}, and x denote the projected point coordinates, sampling offsets, attention weights of the kth sampling point in the m-th attention head and image feature map extracted from the 2D image backbone, respectively. W_m and W'_m are learnable weights, and $\langle a \cdot b \rangle$ is a bilinear interpolation of feature map a from the reference point b. We set the number of attention heads to 4 and sample 4 offset locations for both the I-OPN and IG-RPG modules.

B. Additional Experiments Analysis

Effectiveness in Detecting Sparse Object. We summarize the number of LiDAR points contained within ground truth (GT) bounding boxes in Table A1. Similar to object size statistics, a criterion is required to define whether an object is considered sparse. In this study, we define sparse objects as those whose point counts fall within the Q_1 or Q_2 quantiles. Table A2 presents detection performance stratified by point counts within GT bounding boxes. As the results indicate, performance improvements are observed across all categories, with particularly notable gains for the pedestrian and cyclist classes. Specifically, ImagePG improves performance for objects below Q_1 by +2.65%p and +1.92%p, and below Q_2 by +15.68%p and +5.39%p, respectively. These object types typically exhibit low point densities, underscoring the benefit of our image-guided point generation framework. The results suggest that ImagePG effectively enhances 3D object detection for sparse objects by generating dense and semantically meaningful points informed by image features.

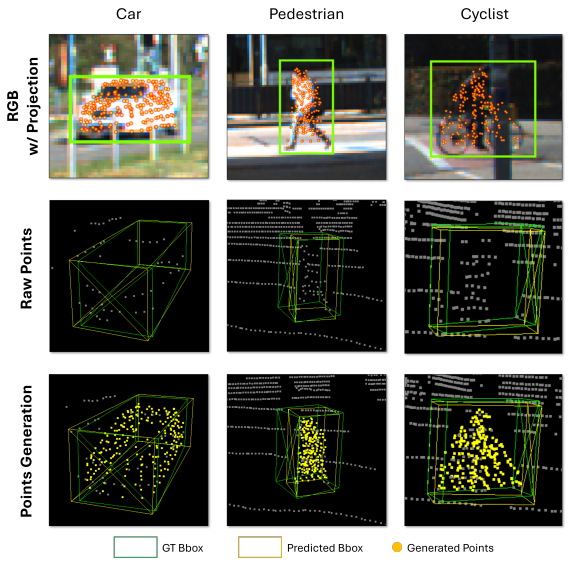

Figure A3. Qualitative results of semantic point generation. The generated points align well with image objects.

Effectiveness in Detecting Occluded Object. Occlusion from surrounding objects or background elements often leads to sparsely sampled point clouds, as occluded regions cannot be captured by LiDAR sensors. To assess the robustness of our proposed ImagePG framework in handling such occlusions, we analyze its performance in object detection under varying occlusion levels. The KITTI [3] benchmark provides occlusion annotations categorized into three levels: 1 (fully visible), 2 (partially occluded), and 3 (heavily occluded). As presented in Table A3, ImagePG consistently outperforms the LiDAR-only baseline [6] across all occlusion levels. Notably, our method achieves substantial improvements even for heavily occluded objects (level 3), demonstrating that image-guided point generation enriches semantic context and effectively compensates for LiDAR sparsity caused by occlusion.

Generalization to LiDAR Resolution. LiDAR sensors yield varying point cloud densities depending on hardware specifications. To evaluate the generalizability of ImagePG under different LiDAR configurations—particularly in low-resolution scenarios—we analyze its performance under varying input sparsity levels. Specifically, we simulate reduced beam counts by converting LiDAR scans into range maps and downsampling their vertical resolution to emulate 32-beam and 16-beam sensors. The resulting range maps are projected back into 3D space to generate sparse point clouds. Both ImagePG and the baseline [6] are trained from scratch on these sparsified inputs using the same configuration as the full 64-beam setting of the KITTI [3].

Table A4 presents the AP results across different LiDAR beam densities on the KITTI [3] *val* set. As the number of LiDAR beams decreases, ImagePG exhibits even larger per-

Method	Car 3D (IoU=0.7)				Pedestrian 3D (IoU=0.5)				Cyclist 3D (IoU=0.5)			
	[Min, Q1)	[Q1, Q2)	[Q2, Q3)	[Q3, Max]	[Min, Q1)	[Q1, Q2)	[Q2, Q3)	[Q3, Max]	[Min, Q1)	[Q1, Q2)	[Q2, Q3)	[Q3, Max]
Baseline [6]	15.59	66.69	87.14	95.74	6.34	28.05	37.82	59.96	7.34	35.36	77.35	86.13
+ ImagePG (Ours)	19.34 (3.75%↑)	69.38 (2.69%↑)	88.56 (1.42%↑)	96.07 (0.34%↑)	8.99 (2.65%↑)	43.73 (15.68%↑)	57.62 (19.80%↑)	78.40 (18.44%↑)	9.26 (1.92%↑)	40.75 (5.39%↑)	80.19 (2.84%↑)	90.70 (4.57%↑)

Table A2. Detection performance comparison across number of points quantiles on the KITTI [3] *val* set. We use 40 recall positions to compute AP. Objects in lower quantiles (e.g., Q_1, Q_2) are sparse and more challenging to detect.

Method	Car 3D (IoU=0.7)			Pedestrian 3D (IoU=0.5)			Cyclist 3D (IoU=0.5)		
	LVL_1	LVL_2	LVL_3	LVL_1	LVL_2	LVL_3	LVL_1	LVL_2	LVL_3
Baseline [6]	84.62	77.01	54.41	60.48	19.56	4.49	83.50	25.60	0.93
+ ImagePG (Ours)	84.77 (0.15%↑)	79.58 (2.57%↑)	60.47 (6.06%↑)	73.96 (13.48%↑)	33.93 (14.37%↑)	9.02 (4.53%↑)	85.53 (2.03%↑)	28.30 (2.70%↑)	2.79 (1.86%↑)

Table A3. Detection performance comparison across different occlusion levels on the KITTI [3] *val* set. We use 40 recall positions to compute AP. Higher occlusion levels correspond to more severely occluded objects.

LiDAR Beams	Method	Car 3D (IoU=0.7)			Pedestrian 3D (IoU=0.5)			Cyclist 3D (IoU=0.5)		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
64	Baseline† [6]	92.49	84.87	82.40	65.86	58.85	53.47	91.29	72.13	67.44
	+ ImagePG (Ours)	92.41 (0.08%↓)	85.67 (0.80%↑)	83.45 (1.05%↑)	77.64 (11.78%↑)	71.01 (12.16%↑)	65.56 (12.09%↑)	96.53 (5.24%↑)	78.94 (6.81%↑)	74.79 (7.35%↑)
32	Baseline† [6]	70.59	48.97	44.26	62.19	54.16	48.15	66.72	40.79	38.55
	+ ImagePG (Ours)	78.82 (8.23%↑)	58.40 (9.43%↑)	53.27 (9.01%↑)	72.80 (10.61%↑)	65.23 (11.07%↑)	58.26 (10.11%↑)	70.81 (4.09%↑)	44.91 (4.12%↑)	42.51 (3.96%↑)
16	Baseline† [6]	52.17	33.27	28.44	40.91	35.16	30.88	35.70	20.39	18.72
	+ ImagePG (Ours)	68.10 (15.93%↑)	45.38 (12.11%↑)	39.15 (10.71%↑)	60.72 (19.81%↑)	52.52 (17.36%↑)	46.26 (15.38%↑)	54.20 (18.50%↑)	31.55 (11.16%↑)	29.66 (10.94%↑)

Table A4. Detection performance comparison across different LiDAR specification settings on the KITTI [3] *val* set. We use 40 recall positions to compute AP. †: Reproduced.

Setting	N_t	FPS	VRAM	Car 3D	Ped. 3D	Cyc. 3D
(a)	2	7.1	4.0	86.90	70.62	82.15
(b)	3	5.5	4.4	87.11	71.02	81.66
(c)	4	4.6	4.4	**87.20**	70.83	82.36
(d)	5	3.8	4.5	87.12	70.76	**84.09**
(e)	6	3.2	4.7	87.18	**71.40**	83.42

Table A5. Computational comparison for the KITTI [3] *val* set. Note that evaluation is conducted with a single NVIDIA A6000 GPU.

Module	2D	3D	I-OPN	RPN	RoI Head	Total
Params (M)	29.13	1.00	2.88	7.31	12.22	52.54

Table A6. Number of parameters of each module in ImagePG.

formance improvements over the baseline [6]. These results highlight that ImagePG not only enhances detection accuracy with high-resolution LiDAR but also demonstrates robustness under low-resolution configurations. This makes ImagePG a versatile and practical solution across diverse LiDAR hardware platforms, particularly enabling effective deployment with low-cost, low-resolution LiDAR sensors.

B.1. Additional Qualitative Analysis

In Figure A3, we visualize the input images, original point clouds, and our generated semantic point clouds. The generated points are well aligned with visual semantics and effectively densify the original LiDAR data. Figure A4 and A5 show the additional detection and points generation results for the KITTI [3] validation and test set, respectively. The results indicate that most objects are successfully detected, with high quality points generation, maintaining low false positives. Note that confidence scores larger than 0.3 are only visualized.

B.2. Additional Efficiency Analysis

Table A5 presents detection performance under varying numbers of transformation actions. For accuracy-critical

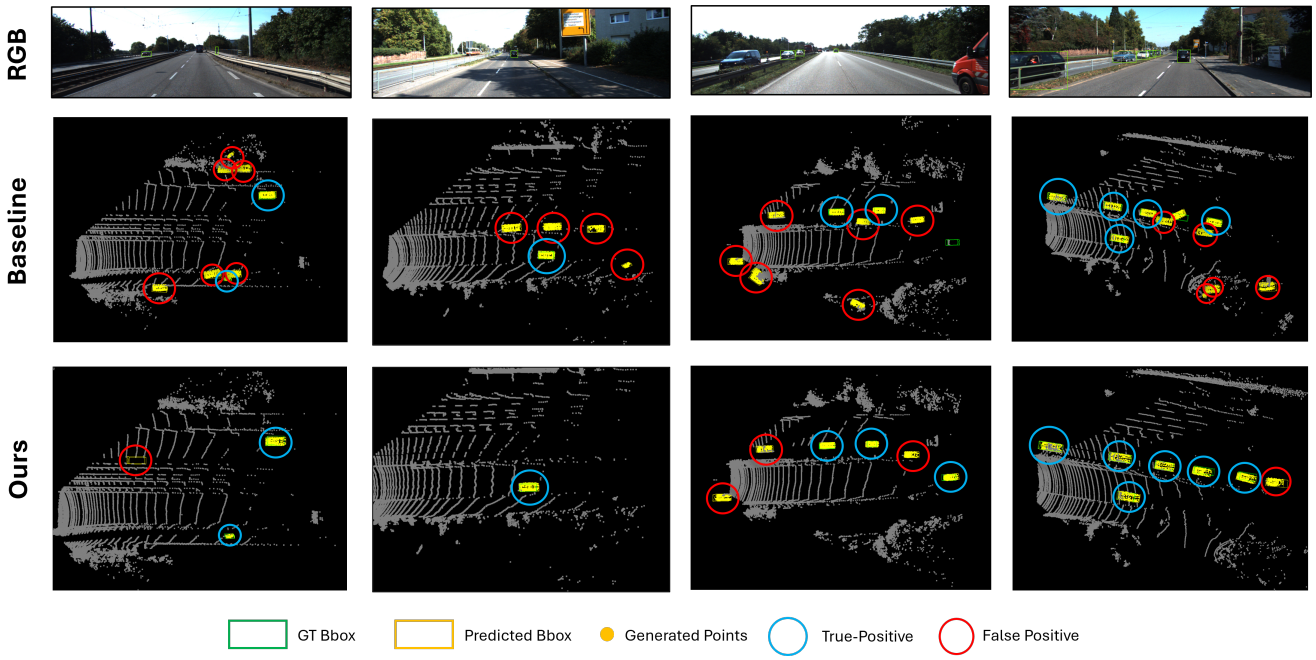

Figure A4. Additional qualitative comparison between the baseline [6] and our proposed ImagePG on the KITTI [3] *val* set. ImagePG demonstrates well-aligned and semantically meaningful point generation while maintaining a low false positive rate.

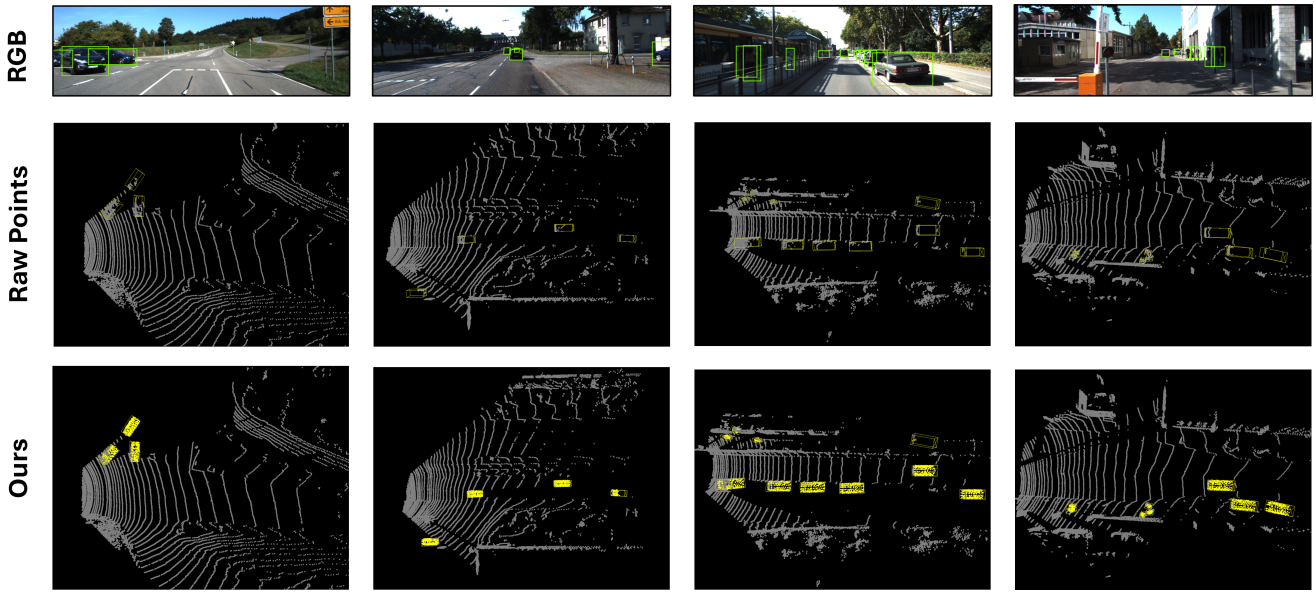

Figure A5. Detection and point generation results on the KITTI [3] *test* set.

applications, configuration (e) yields competitive performance, while configuration (a) offers significantly improved inference speed, making it preferable for latency-sensitive deployments. Table A6 details the parameters, with a total of 52.54M parameters, over half of which (29.13M) are attributed to the image backbone. The overall GPU memory footprint remains moderate at approximately 4.7 GiB, demonstrating the practicality of ImagePG.

References

[1] 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. 1

[2] 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*, pages 1201–1209, 2021. 2

[3] 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. 1, 2, 3, 4

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[5] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 2

[6] Inyong Koo, Inyoung Lee, Se-Ho Kim, Hee-Seon Kim, Woo-Jin Jeon, and Changick Kim. Pg-rcnn: Semantic surface point generation for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18142–18151, 2023. 2, 3, 4

[7] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, et al. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17524–17534, 2023. 2

[8] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 1

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[11] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023. 1

[12] 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*, pages 10529–10538, 2020. 2

[13] Pei Sun, Henrik Kretzschmar, Xavier Dotiwalla, Christophe Chouard, Vivek Patnaik, Paul Tsui, Jiquan 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. 1, 2

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

[15] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 1

[16] Qiangeng Xu, Yiqi Zhong, and Ulrich Neumann. Behind the curtain: Learning occluded shapes for 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2893–2901, 2022. 1

[17] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 1

[18] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2