Supplementary Materials of MonoNeRD: NeRF-like Representations for Monocular 3D Object Detection

Junkai Xu[1,2*] Liang Peng[1,2,*] Haoran Chen[1,2,*] Hao Li[2]
Wei Qian[2] Ke Li[4] Wenxiao Wang[3†] Deng Cai[1,2]
[1]State Key Lab of CAD & CG, Zhejiang University [2]FABU Inc.
[3] School of Software Technology, Zhejiang University [4]Fullong Inc
{xujunkai, pengliang, haorancheng}@zju.edu.cn

Considering the space limitation of the main text, we provide additional results and discussion in this supplementary material, which is organized as follows:

- Section A: Explanations and Discussions
  - Depth-transformed *vs.* NeRF-like. A.1
  - About generalization. A.2
  - Limitations. A.3
- Section B: Additional Quantitative Results
  - KITTI *test* results for other categories. B.1
  - Ablation of 2D and 3D supervisions. B.2
  - Ablation of depth plane samples. B.3
  - Latency analysis. B.4
  - Reproducibility. B.5
- Section C: Additional Qualitative Results
  - What if no object detection? C.1
  - Qualitative results on KITTI *val*. C.2
  - Rendered results. C.3
  - Occupancy demo video. C.4

A. Explanations and Discussions

A.1. Depth-transformed *vs.* NeRF-like

Questions may arise about the superiority of the proposed method when compared to previous depth-based works such as CaDDN [17]. Directly comparing their experimental results is unfair, as there are many differences between CaDDN and MonoNeRD, besides the intermediate representation. For example, CaDDN utlizes a heavy 2D image backbone (ResNet-101 [8]) and depth estimate head (DeepLabV3 [2]) while MonoNeRD inherits auxiliary tasks such as 2D detection and LiDAR feature imitation from the design in LIGA-Stereo [7]. To demonstrate the advantage of the proposed method, we create a fair baseline by replacing our proposed module with a two-layer 2D convolutional neural network (consisting of a 3x3 conv2D layer, ReLU activation function, and a 3x3 conv2D layer) to produce categorical depth distributions [17]. All other modules were left unchanged. The baseline method can be represented by the following equation:

Baseline = MonoNeRD - NeRF-like + CaDDN-trans.

We have already presented the 3D visualization comparison in the main text. In Table 1, we list the differences and experimental results on KITTI 3D [6]. The superiority of the proposed NeRF-like representation can be observed by comparing the results of baseline method and MonoNeRD on KITTI *val* set.

	CaDDN [17]	Baseline	MonoNeRD
Model Size	774M	83M	83M
Backbone	ResNet-101	ResNet-34	ResNet-34
Depth Head	DeepLabV3 [2]	2-Layer-CNN	None
Depth Label	LiDAR/depth completion [9]	LiDAR	LiDAR
3D Rep.	Depth-transformed	Depth-transformed	NeRF-like
3D Det. Head	PointPillars [10]	SECOND [23]	SECOND [23]
LIGA-Auxilary	None	L_{2d}, L_{im}	L_{2d}, L_{im}
KITTI *val* AP_{3D}	23.57/16.31/13.84	18.75/14.49/12.55	20.64/15.44/13.99
KITTI *test* AP_{3D}	19.17/13.41/11.46	-	22.75/17.13/15.63

Table 1: Comparison between CaDDN and MonoNeRD. We create a baseline for convenience and fair comparison. "3D Rep." and "3D Det. Head" refer to 3D Representation and 3D Detection Head. Baseline = MonoNeRD - NeRF-like + CaDDN-transformation.

A.2. About generalization

The generalization of a neural network model refers to its ability to accurately perform on new, unseen data that was not used during training. It is usually used to measure

*Work performed during an internship at FABU Inc.
†Corresponding author

how well the model has learned the underlying patterns and relationships in the training data and its ability to apply such knowledge to new data. In this section, we will discuss the generalization of the proposed MonoNeRD from several aspects that readers may be concerned about.

Generalization across different scenes. Vanilla NeRF [12] encodes the scene into a multi-layer perceptron (MLP). Given any input 3D coordinate and viewing direction, it outputs the corresponding volume density and radiance. The generalization of NeRF is revealed by synthesizing unseen novel views, in other words, the ability to interpolate between 3D coordinates or view directions that were not present in the training data. So vanilla NeRF has to be optimized per scene because new scenes are entirely different domains for a trained NeRF model. MonoNeRD takes monocular images as input and predicts the signed distance scalar for pre-defined 3D locations. By incorporating inductive biases, such as translational equivariance, through the convolutional architecture design, the purposed method learns underlying patterns and representations that are shared by the training data allowing it to generalize to previously unseen scenes. Such generalization ability has also been demonstrated by researches in sing-view reconstruction [18], 3D-aware view synthesis [11, 27] and generative models [4, 19, 13].

Generalization gap between training and inference. In general, neural network models can have better generalization ability when trained on larger datasets compared to smaller ones. It is because larger datasets typically provide more diverse samples for the model to learn from, so the underlying patterns and relationships in the training data can be better captured. In Table 2, we present the performance of MonoNeRD and other two depth-based methods (CaDDN [17], DID-M3D [15]) when dealing with datasets of varying sizes. It can be observed that MonoNeRD benefits more from larger datasets. We attribute this success to the introduced dense NeRF-like 3D representation.

	KITTI *val*	KITTI *test*	Waymo *val*
Train size	3k	7k	50k
Test size	3k	7k	40k
Class	Car	Car	Vehicle
Metric	AP_{3D}	AP_{3D}	3D mAP
IoU	0.7	0.7	0.5
Setting	Moderate	Moderate	Level 1/ Overall
CaDDN [17]	16.31	13.41	17.54
DID-M3D [15]	16.12	16.29	20.66
MonoNeRD	15.44	17.13	31.18
vs. CaDDN	-5.33%	+27.74%	+77.77%
vs. DID-M3D	-4.22%	+5.16%	+50.92%

Table 2: Performance gap with different data sizes.

Generalization for cross-dataset. The generalization for cross-dataset is a popular research topic in the field of domain adaption. In 3D object detection, the task of adapting detectors from one dataset to another is first introduced by [22]. Current researches [24, 25] mainly focus on LiDAR-based detection because LiDAR points exhibit relatively consistent properties across different datasets, such as translational invariance and rotational equivariance. Image samples in different datasets are collected using cameras with different intrinsics, making it challenging for image-based detectors to mitigate the domain gap. Our MonoNeRD is a typical monocular 3D object detector that heavily relies on the camera intrinsics to achieve 2D-3D constraints. We do not design any special module to handle the cross-dataset setting because the domain adaption task is not in the scope of this paper.

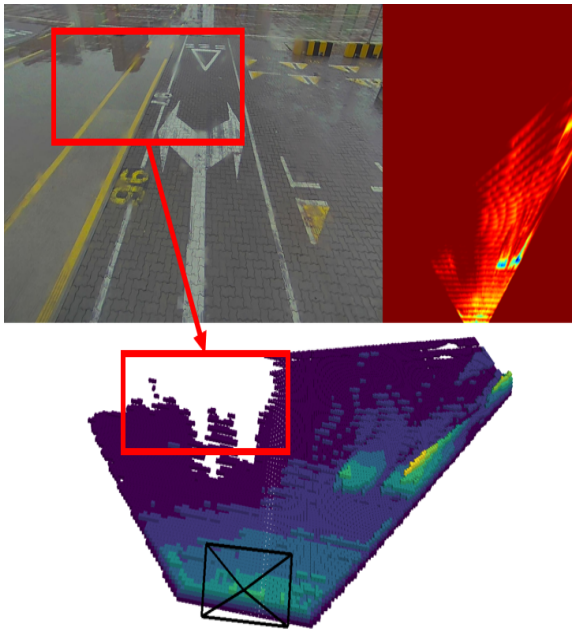


Figure 1: MonoNeRD fails to predict density at glossy surfaces (red box area). We find such cases by running MonoNeRD on some outdoor monocular images of our in-house data.

A.3. Limitations

We briefly discussed the limitations of our method in the main text, and here we provide more detailed explanations. First, working with bounds modeling in volume rendering is generally believed to help improve the learning of implicit scenes [26, 28, 12]. However, for monocular 3D detection task, the frustum covers an infinite range of distances in the camera view, and incorporating image features from out-of-bound areas, *e.g.*, the sky, could harm the final detec-

Methods	*Pedestrian* AP_{BEV}/AP_{3D}			*Cyclist* AP_{BEV}/AP_{3D}		
	Easy	Moderate	Hard	Easy	Moderate	Hard
M3D-RPN[1]	5.65 / 4.92	4.05 / 3.48	3.29 / 2.94	1.25 / 0.94	0.81 / 0.65	0.78 / 0.47
D4LCN[5]	5.06 / 4.55	3.86 / 3.42	3.59 / 2.83	2.72 / 2.45	1.82 / 1.67	1.79 / 1.36
MonoPair[3]	10.99 / 10.02	7.04 / 6.68	6.29 / 5.53	4.76 / 3.79	2.87 / 2.12	2.42 / 1.83
MonoFlex[29]	10.36 / 9.43	7.36 / 6.31	6.29 / 5.26	4.41 / 4.17	2.67 / 2.35	2.50 / 2.04
CaDDN[17]	14.72 / 12.87	9.41 / 8.14	8.17 / 6.76	**9.67 / 7.00**	**5.38 / 3.41**	**4.75 / 3.30**
LPCG[14]	12.11 / 10.82	7.92 / 7.33	6.61 / 6.18	8.14 / 6.98	4.90 / 4.38	3.86 / 3.56
MonoNeRD(ours)	**15.27/ 13.20**	**9.66 / 8.26**	**8.28 / 7.02**	5.24 / 4.73	2.80 / 2.48	2.55 / 2.16

Table 3: Performance for *Pedestrian* and *Cyclist* on KITTI *test*. The best results are **bold**.

tion performance. Second, signed distance functions (SDF) based modeling cannot represent non-watertight manifolds or manifolds with boundaries, such as zero thickness surfaces. Third, as described in Section A.2, our MonoNeRD takes only one image as input so it is impossible to reconstruct a 360 degree scene. Fourth, our proposed method fails to deal with glossy surfaces (as shown in Figure 1) because we model the radiance at each point as a function of the spatial location. This limitation has also been discussed by [21].

B. Additional Quantitative Results

B.1. KITTI *test* results for other categories

Table 3 shows the performance comparison for *Pedestrian* and *Cyclist* categories on the KITTI *test* server. Our method achieves state-of-the-art performance for *Pedestrian* category and obtains comparable results for *Cyclist* category. The performance gap on cyclist can be attributed to the limited number of training samples available for this class. Additionally, we provide 3D volume densit visualization results (Section C.4) for cyclists.

B.2. Ablation of 2D and 3D supervisions

Exp.	Setting			AP_{BEV}/AP_{3D}		
	$L_{rgb}^{left}+L_{rgb}^{right}$	L_{depth}	L_{sdf}	Easy	Moderate	Hard
1	✓			25.49 / 18.20	19.56 / 14.11	17.18 / 12.22
2		✓		26.91 / 18.72	20.87 / 14.54	18.57 / 12.60
3	✓	✓		27.60 / 20.07	20.61 / 14.66	17.95 / 12.45
4		✓	✓	27.94 / 20.28	21.44 / **15.32**	18.82 / 13.48
5	✓	✓	✓	**29.65 / 20.82**	**22.10** / 15.29	**20.02 / 13.55**

Table 4: Ablation of 2D and 3D supervisions. "$L_{rgb}^{left}+L_{rgb}^{right}$": using stereo images (left and right RGB images) as implicit depth supervision (See Figure 5 in the main text), depth supervision exists implicitly under this setting; "L_{depth}": using LiDAR depth labels as depth supervision; "L_{sdf}": using SDF loss for explicit 3D supervision. We can see that even explicit depth supervision is unavailable, our method still can learn depth information from stereo images.

In this section, our aim is to investigate the impact of different depth supervisions and explicit 3D supervision. we use two types of depth supervisions: (1) LiDAR depth labels, which provide explicit supervision, and (2) stereo RGB images, which force the network to implicitly learn depth through the reconstruction loss between rendered RGB images and the original RGB images. Table 4 reveals that both depth supervisions, namely LiDAR depth labels stereo RGB images (referenced in the main text Section 4.1.2) are able to provide 3D information for training (Exp. 1 and 2). Interestingly, when enforcing both types of depth supervision during training, the model does not exhibit significant improvements in performance (Exp. 3). It is possibly because the depth loss and multi-view (stereo) loss essentially provide different types of depth information via volume rendering, where the former uses a explicit manner and the latter uses a implicit manner. Such heterogeneous depth supervisions may bring heavy learning burdens for the network. Furthermore, implementing the SDF loss leads to additional improvements in the model (Exp. 2 $\rightarrow$ 4 and 3 $\rightarrow$ 5). The SDF loss helps the model to concentrate on geometry surfaces by enforcing 3D constraints, which even facilitates the learning of different types of depth supervisions (Exp. 4 $\rightarrow$ 5). It is worthy noting that the stereo hardware is not a general setting for most robot/self-driving systems. Thus we focus on the monocular setting, namely, other experiments including the main text and this supplementary material do not employ the right RGB image loss (L_{rgb}^{right}).

B.3. Ablation of depth plane samples

Exp.	D	AP_{BEV}/AP_{3D}		
		Easy	Moderate	Hard
a	36	26.54 / 17.50	20.16 / 13.84	18.23 / 12.02
b	54	28.07 / 18.79	21.31 / 14.57	18.47 / 12.39
c	72	29.03 / 20.64	22.03 / 15.44	19.41 / 13.99
d	108	28.41 / 20.31	21.33 / 15.43	18.78 / 13.52

Table 5: Ablation of the number of sampled depth planes ("D").

We also conducted an ablation study on the depth plane sampling number D mentioned in Section 4.1.1 of main

Method	Run	3D mAP / mAPH (IoU = 0.7)				3D mAP / mAPH (IoU = 0.5)			
		Overall	0 - 30m	30 - 50m	50m - ∞	Overall	0 - 30m	30 - 50m	50m - ∞
MonoNeRD	LEVEL 1								
	1	10.56 / 10.46	27.36 / 27.13	5.38 / 5.34	0.65 / 0.65	31.33 / 30.84	61.33 / 60.51	26.21 / 25.84	6.60 / 6.47
	2	10.69 / 10.59	27.84 / 27.59	5.31 / 5.27	0.75 / 0.74	31.15 / 30.69	60.70 / 59.91	26.05 / 25.71	6.54 / 6.41
	3	10.74 / 10.62	28.31 / 27.99	5.51 / 5.46	0.76 / 0.75	31.07 / 30.57	61.31 / 60.43	25.97 / 25.59	6.65 / 6.53
	Avg	**10.66 / 10.56**	**27.84 / 27.57**	**5.40 / 5.36**	**0.72 / 0.71**	**31.18 / 30.70**	**61.11 / 60.28**	**26.08 / 25.71**	**6.60 / 6.47**
	LEVEL 2								
	1	9.93 / 9.84	27.28 / 27.04	5.23 / 5.19	0.54 / 0.53	29.43 / 28.97	61.14 / 60.32	25.49 / 25.13	5.77 / 5.66
	2	10.05 / 9.96	27.75 / 27.50	5.16 / 5.13	0.62 / 0.61	29.26 / 28.83	60.49 / 59.70	25.33 / 25.00	5.72 / 5.61
	3	10.10 / 9.99	28.22 / 27.90	5.36 / 5.31	0.63 / 0.62	29.18 / 28.71	61.10 / 60.22	25.25 / 24.88	5.82 / 5.71
	Avg	**10.03 / 9.93**	**27.75 / 27.48**	**5.25 / 5.21**	**0.60 / 0.59**	**29.29 / 28.84**	**60.91 / 60.08**	**25.36 / 25.00**	**5.77 / 5.66**

Table 6: Three different runs on on Waymo *val* set.

text. As presented in Table 5, we can see consistent improvements as D is increased up to 72 sampling planes, and the detector performance stays relatively stable from 72 to 108. We think it could be a constraint from high-resolution sample strategy, which will oversample along the depth axis, resulting in too many frustum planes with similar features.

B.4. Latency analysis

	2D backbone	Our module	Detection module
Latency	25.7ms	91.5ms	94.6ms
Ratio	12.13%	43.20%	44.66%

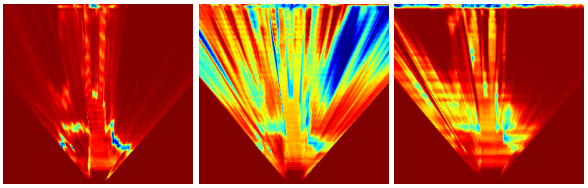
Table 7: Latency for different modules.

We analyze the inference latency of our method. The average runtime of each module is shown in Table 7. We do not include the time taken for calculating coordinate projection in the analysis as it can be pre-calculated before training or inference.

B.5. Reproducibility

To see the reproducibility of our proposed method, we conducted three runs of MonoNeRD on Waymo Open Dataset (WOD) [20] *val* split, and the results are presented in Table 6.

C. Additional Qualitative Results



(a) Exp.(g) (b) w/o det. (c) right + w/o det.

Figure 2: Visualizations of bev-density. From left to right: with detection loss, without detection loss, without detection loss but employing right camera view supervision.

C.1. What if no object detection?

We find the information regarding occluded regions primarily relies on the object detection loss. To further investigate the impact, we conduct two experiments based on experiment (g) in the main text. Figure 2 presents the visualization of accumulated Bird's Eye View (BEV) density. When removing the object detection loss, the prediction of occluded regions is uncontrollable. However, when incorporating supervisions from other views, we find that the problem of uncontrollable predictions is mitigated.

C.2. Qualitative results on KITTI *val*

We present several qualitative examples on KITTI *val* set in Figure 3.

C.3. Rendered results

We show the rendered results in Figure 4 and 5. All the rendered visualization results are corresponding to the experiments in Table 4. The depth maps with both LiDAR labels and stereo RGB images supervision are better than only with LiDAR depth labels, especially in the areas where LiDAR depth labels are not available, *e.g.*, the sky.

C.4. Occupancy demo video

We select two sequence videos from KITTI Raw Data [6], which are not included in the Object Detection subdataset, to conduct the 3D occupancy (volume density) visualization. The details of our selections are shown in Table 8. Our 3D visualization is implemented with Mayavi [16]. The generated video clips (*sequence_0.mp4*, *sequence_1.mp4*) are provided in the supplementary zip file. We choose several frames in two video clips and show them in Figure 6 and 7.

Video tag	Raw data sequence	Index
0	2011_09_26_drive_0005_sync	0000000020-0000000090
1	2011_10_03_drive_0042_sync	0000000180-0000000210

Table 8: Our selections for 3D occupancy visualization.

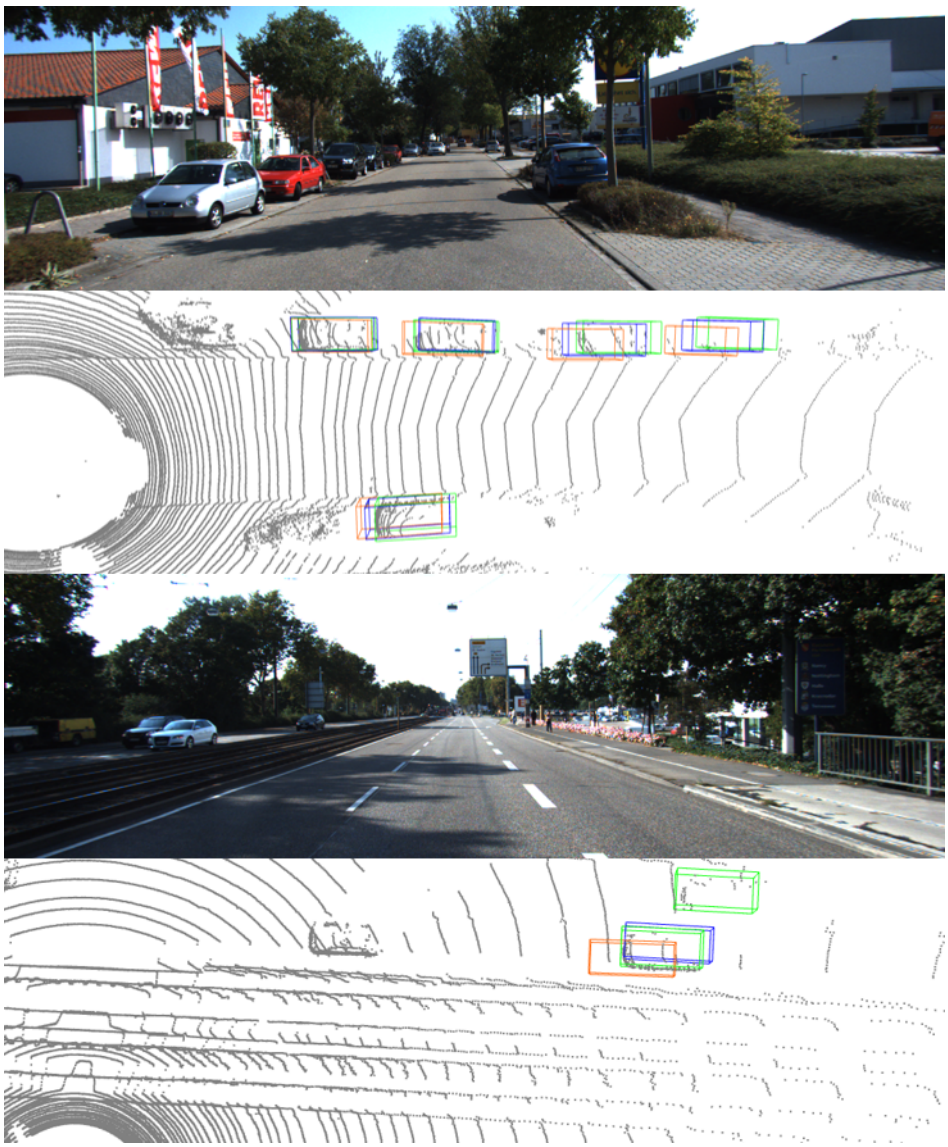
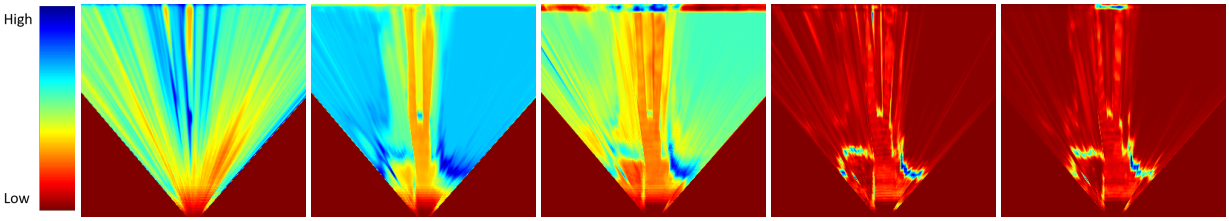
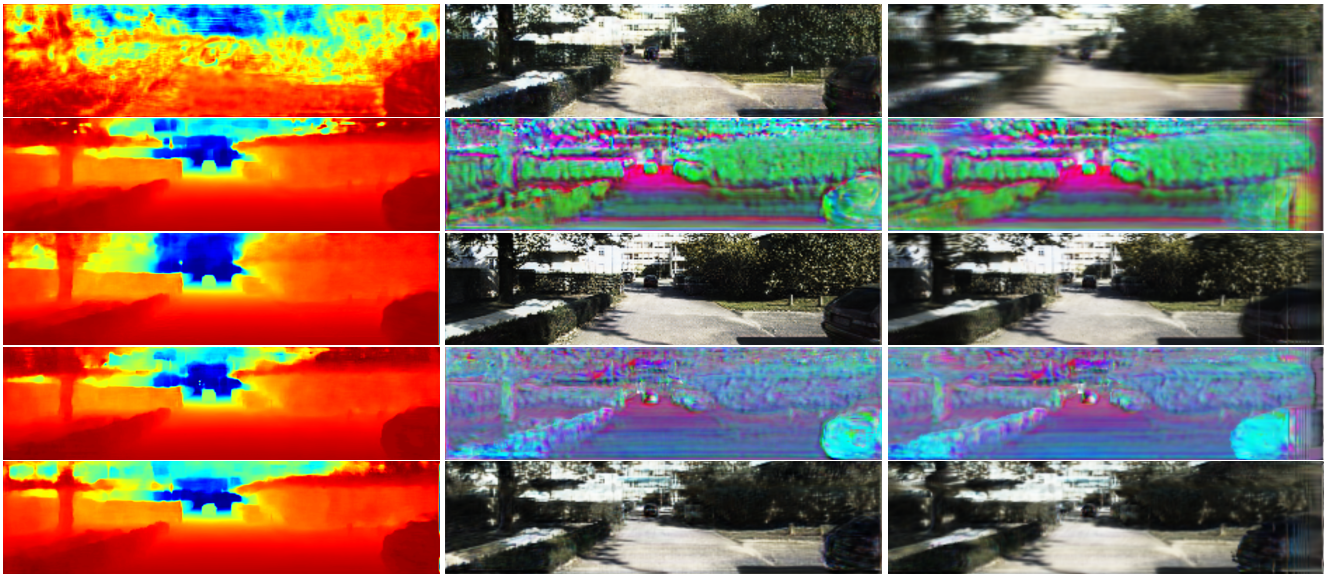


Figure 3: Qualitative visualization of MonoNeRD detections from KITTI-3D *val* set. Green: ground-truth 3D boxes; Orange: baseline predictions; Blue: our predictions. Best viewed in color with zoom in.

(a) Original stereo images. Note that only the left image is the model input.



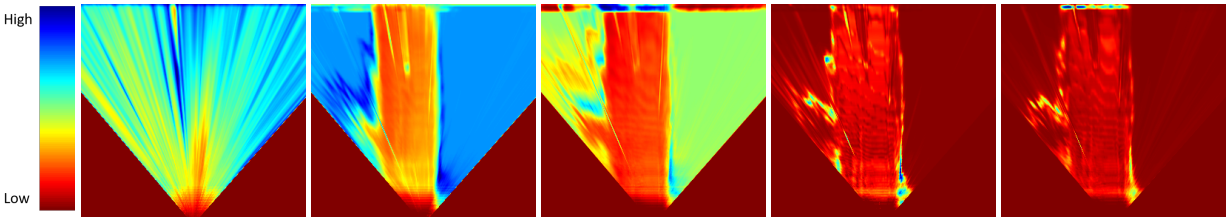
(b) Visualizations of bev-density, density is accumulated along with the height axis. From left to right: Experiment setting (1, 2, 3, 4, 5), respectively. Low density means the 3D space is empty, while high density means it is occupied. Zoom in for better details.



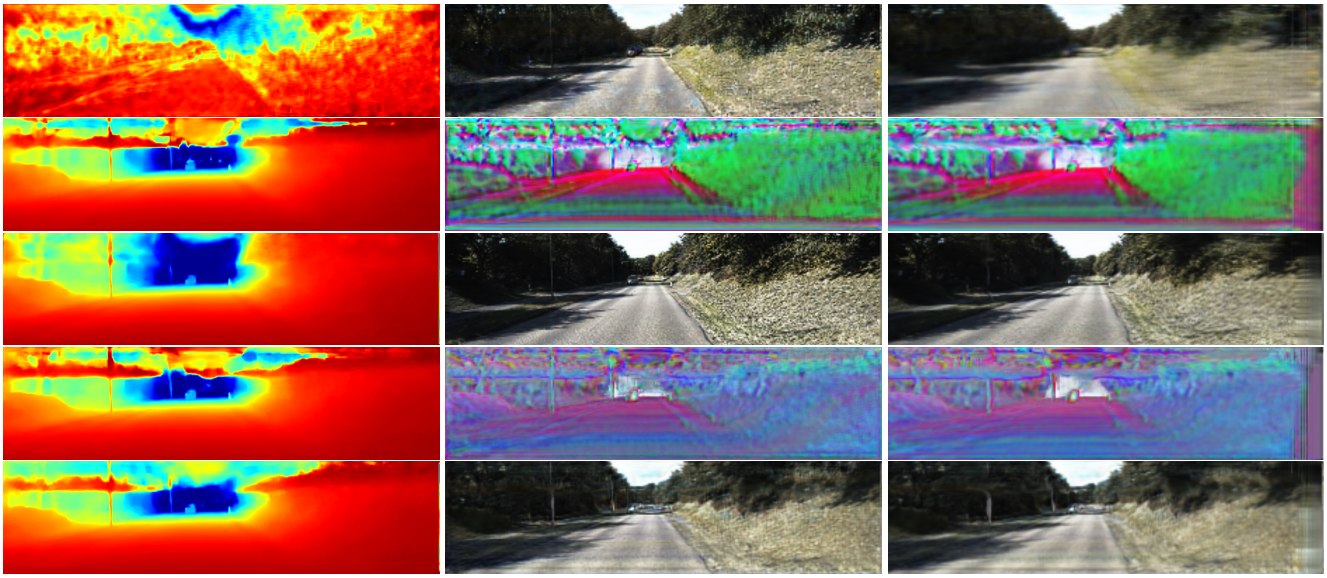
(c) Rendered depth maps. (d) Rendered left RGB images. (e) Rendered right RGB images.

Figure 4: The first case rendered results of Experiments in Table 4 . The rendered images from top to bottom: Experiments setting (1, 2, 3, 4, 5), respectively. Exp. (1) uses stereo images as depth supervision. Exp. (2) uses LiDAR depth labels as depth supervision. Exp. (3) uses both supervisions. Exp. (4) uses LiDAR depth labels for depth supervision and SDF loss for direct 3D supervision. Exp. (5) uses all supervisions. Zoom in for better details.

(a) Original stereo images. Note that only the left image is the model input.



(b) Visualizations of bev-density, density is accumulated along with the height axis. From left to right: Experiment setting (1, 2, 3, 4, 5), respectively. Low density means the 3D space is empty, while high density means it is occupied. Zoom in for better details.



(c) Rendered depth maps. (d) Rendered left RGB images. (e) Rendered right RGB images.

Figure 5: The second case rendered results of Experiments in Table 4. The rendered images from top to bottom: Experiments setting (1, 2, 3, 4, 5), respectively. Exp. (1) uses stereo images as depth supervision. Exp. (2) uses LiDAR depth labels as depth supervision. Exp. (3) uses both supervisions. Exp (4). uses LiDAR depth labels for depth supervision and SDF loss for direct 3D supervision. Exp (5). uses all supervisions. Zoom in for better details.

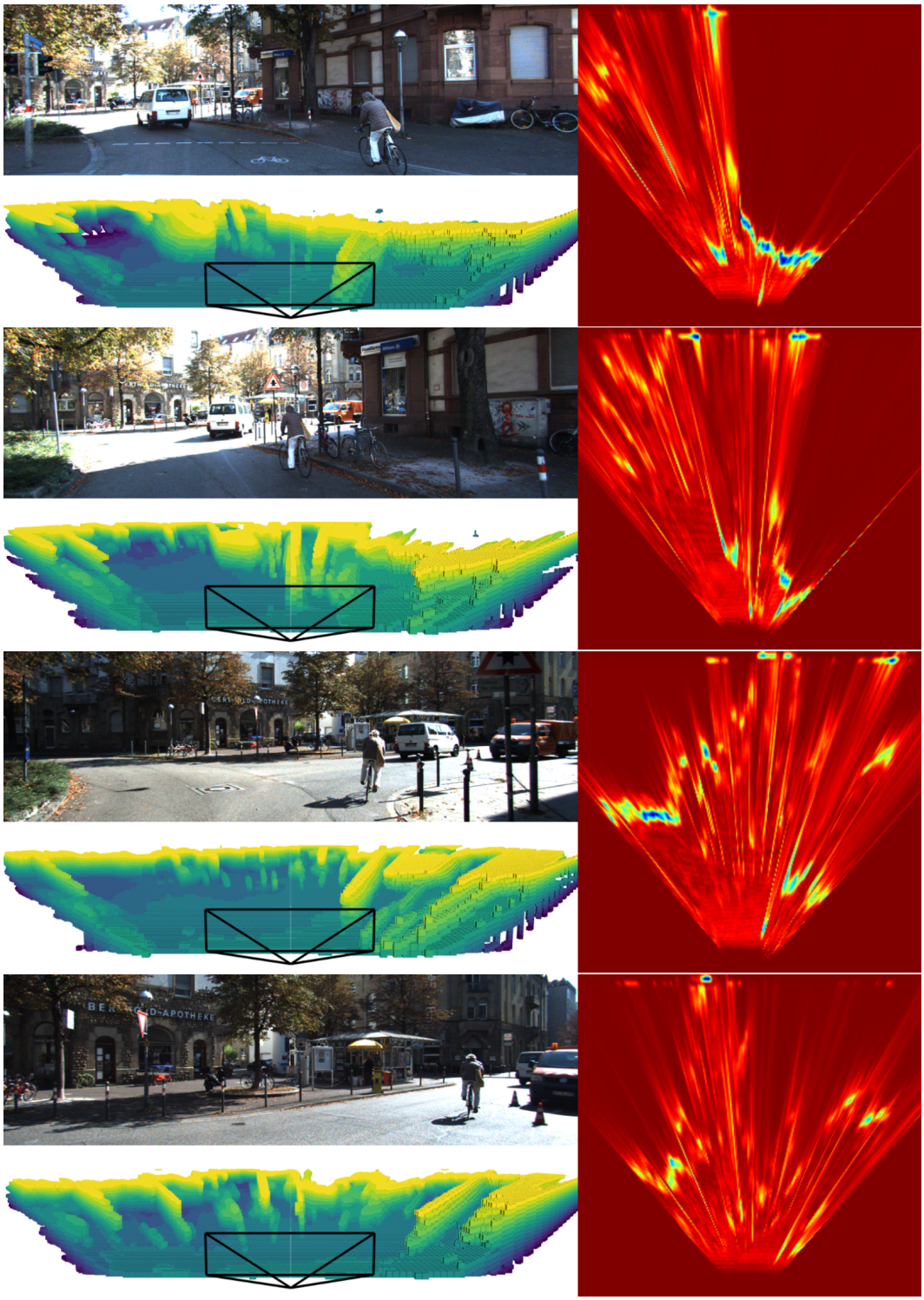


Figure 6: 3D occupancy (volume density) visualizations of Sequence (0). For each frame, top left is the input monocular image, the bottom left is the produced 3D volume density, and the right one is visualization of bev-density. Zoom in for better details. Video clip of this sequence (*sequence_0.mp4*) is provided in supplementary zip file.

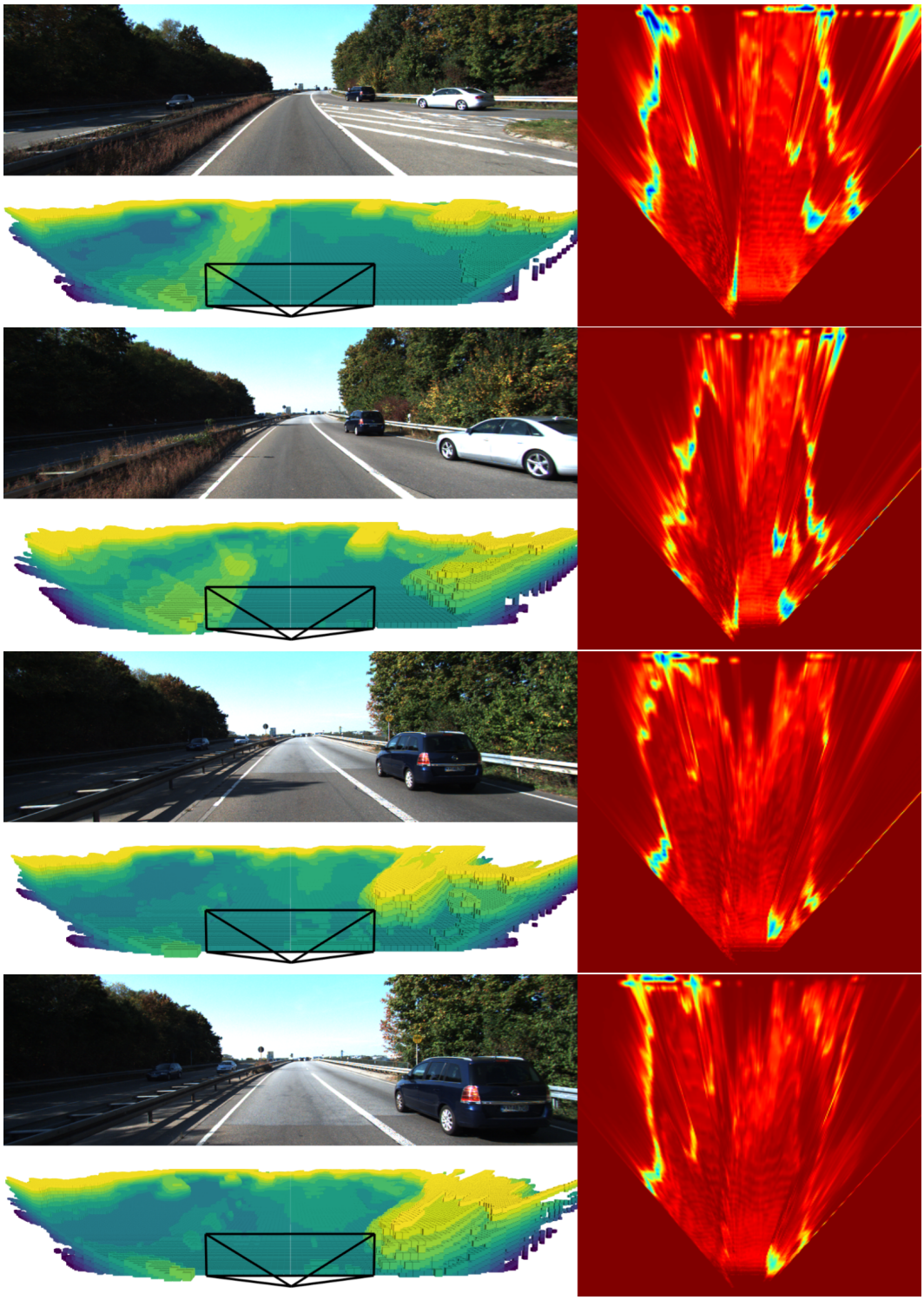


Figure 7: 3D occupancy (volume density) visualizations of Sequence (1). For each frame, top left is the input monocular image, the bottom left is the produced 3D volume density, and the right one is visualization of bev-density. Zoom in for better details. Video clip of this sequence (*sequence_1.mp4*) is provided in supplementary zip file.

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*, pages 9287–9296, 2019. 3

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1

[3] Yongjian Chen, Lei Tai, Kai Sun, and Mingyang Li. Monopair: Monocular 3d object detection using pairwise spatial relationships. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12093–12102, 2020. 3

[4] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[5] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1000–1001, 2020. 3

[6] 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, 4

[7] Xiaoyang Guo, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Liga-stereo: Learning lidar geometry aware representations for stereo-based 3d detector. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3153–3163, 2021. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1

[9] Jason Ku, Ali Harakeh, and Steven L Waslander. In defense of classical image processing: Fast depth completion on the cpu. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 16–22. IEEE, 2018. 1

[10] 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*, pages 12697–12705, 2019. 1

[11] Jiaxin Li, Zijian Feng, Qi She, Henghui Ding, Changhu Wang, and Gim Hee Lee. Mine: Towards continuous depth mpi with nerf for novel view synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12578–12588, 2021. 2

[12] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2

[13] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2

[14] Liang Peng, Fei Liu, Zhengxu Yu, Senbo Yan, Dan Deng, Zheng Yang, Haifeng Liu, and Deng Cai. Lidar point cloud guided monocular 3d object detection. In *European Conference on Computer Vision*, pages 123–139. Springer, 2022. 3

[15] 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*, pages 71–88. Springer, 2022. 2

[16] Prabhu Ramachandran and Gaël Varoquaux. Mayavi: 3d visualization of scientific data. *Computing in Science & Engineering*, 13(2):40–51, 2011. 4

[17] 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*, pages 8555–8564, 2021. 1, 2, 3

[18] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2304–2314, 2019. 2

[19] Katja Schwarz, Yiyi Liao, Michael Niemeyer, and Andreas Geiger. Graf: Generative radiance fields for 3d-aware image synthesis. *Advances in Neural Information Processing Systems*, 33:20154–20166, 2020. 2

[20] 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. 4

[21] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. 3

[22] Yan Wang, Xiangyu Chen, Yurong You, Li Erran Li, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. Train in germany, test in the usa: Making 3d object detectors generalize. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11713–11723, 2020. 2

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

[24] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d: Self-training for unsupervised domain adaptation on 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10368–10378, 2021. 2

[25] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. St3d++: denoised self-training for unsupervised domain adaptation on 3d object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[26] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems*, 34:4805–4815, 2021. 2

[27] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 2

[28] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. *arXiv preprint arXiv:2010.07492*, 2020. 2

[29] 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*, pages 3289–3298, 2021. 3