Supplementary Material of GaussianOcc

Wanshui Gan [1,2,*] Fang Liu [1,*] Hongbin Xu [3] Ningkai Mo [4] Naoto Yokoya [1,2,†]

[1]The University of Tokyo,[2] RIKEN, [3]South China University of Technology
[4]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[*] Equal contribution, [†] Corresponding author

{wanshuigan, fangliu2896, hongbinxu1013, nk.mo19941001}@gmail.com
yokoya@k.u-tokyo.ac.jp

Abstract

In this supplementary material, we provide more implementation details, experiment results with analysis, and further discussion on the limitations and future work.

1. More implementation details

The detailed parameter setting in Gaussian attributes estimation network [15]. During the joint depth and 6D pose training in stage 1, we predict the 3D Gaussian parameters alongside the 2D depth map. Since the Gaussian parameters are well-arranged in the 2D image plane prior to unprojection, we maintain equal scaling across all three dimensions of each 3D Gaussian and constrain the maximum scale to 0.02. Given that the scale s is uniform across all dimensions, we set the rotation matrix $\mathbf{R}$ to the identity matrix. Additionally, we assign an opacity value of 1 to each 3D Gaussian, ensuring that every 2D depth value corresponds to a valid point in 3D space. We do not predict the color defined by SH coefficients c, while we directly use the source RGB image as the color map the same as in [18].

The detailed parameter setting in voxel grid splatting rendering for semantic rendering. For semantic rendering, we chose a fixed scale for each grid vertex to ensure a well-arranged structure that accurately models the 3D space. If we use a learnable scale, it may lead to a situation where the scale is small but the opacity is large, which may not be captured in the rendered depth map and semantic map but could still affect the 3D occupancy result. Therefore, using a fixed scale is simple and sufficient for optimization, as demonstrated in the results presented in the main paper (Table 5) that the performance is close to the learnable scale. Since the scale $s \in \mathbb{R}^3_+$ are identical for both three dimensions, we do not need to predict the rotation $r \in \mathbb{R}^4$ and set it with the identical matrix is sufficient. Similar to the OccNeRF [16], we render the 2D feature map for the semantic regression, while we leverage the 3D Gaussian splatting rendering and OccNeRF uses volume rendering.

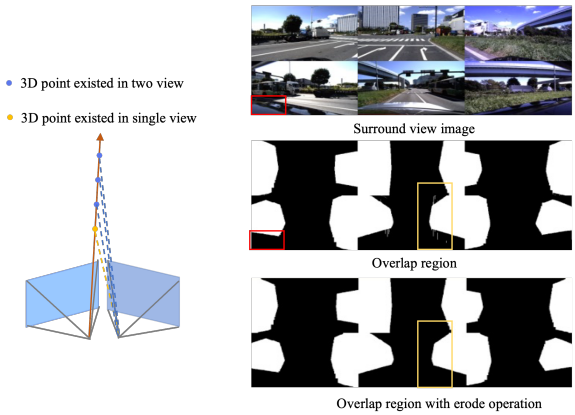

Figure 1. **Overlap mask generation on DDAD dataset**.

The detailed overlap mask generation. In Figure 1, we provide additional visualizations of the overlap mask generation process on the DDAD dataset [4]. To ensure accuracy, we exclude self-occluded regions, such as parts of the vehicle body, which are highlighted by the red rectangle. Additionally, we observe that the generated mask contains noise, as indicated by the yellow rectangle. To address this, we apply an erosion operation using the OpenCV library [1] with a threshold of 20.

The detailed parameter setting in training. We follow the training setting as OccNeRF [16], the resolution of input images and rendered depth maps are set as 384×640 and 180×320 respectively. All experiments are conducted on 8 NVIDIA A100 (40 GB).

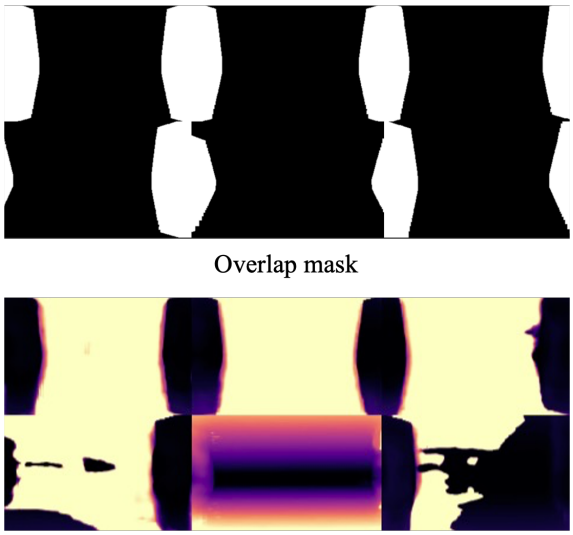

Figure 2. **One-stage training analysis**. This is the visualization of the overlap mask and the rendered depth map with one-stage training.

The detailed training strategy of the self-supervised pretraining setting. We first do the self-supervised training with 12 epochs with the learned pose from stage 1 and the 2D pseudo semantic label, which does not require the 3D occupancy label. Then, we finetune the model with 12 epochs with the 3D occupancy label. We add the RayIoU metric [11] in Table 4 for a comprehensive comparison.

The detailed definition of depth map metric. Following the depth estimation task [14], we report the depth map evaluation with the following metrics,

$$
\begin{aligned}
\text{Abs Rel:} & \frac{1}{|M|} \sum_{d \in M} \left| \hat{d} - d^* \right| / d^*, \\
\text{Sq Rel:} & \frac{1}{|M|} \sum_{d \in M} \left\| \hat{d} - d^* \right\|^2 / d^*, \\
\text{RMSE:} & \sqrt{\frac{1}{|M|} \sum_{d \in M} ||\hat{d} - d^*||^2}, \\
\text{RMSE log:} & \sqrt{\frac{1}{|M|} \sum_{d \in M} ||\log \hat{d} - \log d^*||^2}, \\
\delta < t : \% \text{ of } d \text{ s.t. } & \max \left(\frac{\hat{d}}{d^*}, \frac{d^*}{\hat{d}} \right) = \delta < t,
\end{aligned}
\tag{1}
$$

where M is the valid pixel, $\hat{d}$ is the ground truth depth and d^* is the predicted depth.

2. More experiment results and analysis

Why we need two-stage training. We made extensive efforts to develop one-stage training that directly applies the cross-view loss to the rendered depth map but were unsuccessful. As suggested in Figure 2, the cross-view supervision signals are effective only in overlapping regions. The rendered depth map learned from 3D CNN has lower generalization ability in non-overlap regions compared with the decoder depth learned by the 2D CNN, which led to local minima.

mIoU metric. Due to the limited space in the main paper, the full table of mIoU results is presented in Table 2 for reference.

RayIoU metric. In addition to the mIoU metric for 3D occupancy estimation, we also evaluate our method with a novel metric, RayIoU, introduced by the recent work [11]. The RayIoU is a ray-based evaluation metric that resolves the inconsistency penalty along the depth axis introduced in the traditional voxel-level mIoU criteria. As shown in Table 3, our approach also outperforms OccNeRF [16] in this metric as well. It's important to note that the FPS is calculated excluding rendering time. Since GaussianOcc and OccNeRF utilize the same network architecture, they share the same inference time when the rendering process is not taken into account.

More visualization. We provide more visualization for nuScenes dataset in Figure 3. Please check the videos for sequence visualization in https://github.com/GANWANSHUI/GaussianOcc.git.

More analysis on 3D occupancy and depth map result on different supervision types.

3D occupancy analysis: In Figure 4, we present visualizations of different supervision types. These visualizations highlight key differences in the results for the invisible regions (marked with red rectangles) and the rendered depth quality (marked with green rectangles).

Experiments (1) and (2) involve supervision using ground truth (GT) occupancy labels. Specifically:

Experiment (1) is trained without the visible mask provided by Occ3D-nuScenes [13], which defines the visibility of the occupancy labels. Without this mask, the invisible regions are treated as empty, and the loss function is applied to these regions as well. Experiment (2), on the other hand, excludes the loss computation in invisible regions. From the results, we observe that in Experiment (1), the model tends to predict empty values for invisible regions due to empty loss penalty. In contrast, Experiment (2), by ignoring the loss in invisible areas, shows more non-empty predictions in these regions.

Self-supervised experiments (3) and (4) rely on rendering techniques, which inherently cannot optimize predictions in invisible regions. This limitation leads to non-empty predictions in the red-highlighted areas. Notably, Ex-

Render resolution	Render time (s) with different voxel resolutions (Gaussians number)		
180 ×320	16 ×200 × 200	24 ×320 × 320	32 ×512 × 512
VR	≈ 0.50	≈ 0.85	≈ 1.52
SR	≈ 0.06	≈ 0.17	≈ 0.44

Table 1. Comparison of rendering efficiency under different Gaussians number between volume rendering (VR) [16] and splatting rendering (SR, Ours).

periment (4) frequently predicts invisible regions as related to foreground categories, as shown in the dark rectangles. Conversely, Experiment (3) demonstrates a consistent tendency to classify invisible regions as man-made structures, likely because the surrounding environment predominantly consists of man-made elements.

Render depth map analysis: In Tables 2 and 6 of main paper, we observe an interesting phenomenon that the semantic information is helpful for the depth estimation with our GaussianOcc whereas it worsens the result in OccNeRF. In Figure 4, we visualize the depth map and highlight with green rectangles that our Gaussian splatting rendering produces higher-quality depth predictions compared to volume rendering. This should be concluded to the biased sampling strategy of OccNeRF, where only 25% of the sample points are used for faster semantic map rendering compared to depth map rendering. Here is the piece of the code in OccNeRF [16]. In contrast, our proposed Gaussian splatting method, which renders directly from the voxel vertices, eliminates this issue. At last, since Experiments (1) and (2) do not involve rendering-based training, they fail to produce reasonable depth predictions.

Gaussians number and its related render time: (1) In stage 1, Gaussians number depends on the depth map resolution from the 2D decoder, where each pixel is a Gaussian primitive after unprojection. We use the depth map resolution in 224×352, resulting in 78,848 Gaussian primitives in one image. In stage 2, Gaussians number depends on the voxel resolution, where each voxel grid is a Gaussian primitive. In Table 7 of the main paper, we follow the voxel resolution the same as OccNeRF [52] in $24 \times 300 \times 300$, resulting in 2,160,000 Gaussian primitives. (2) We revealed the rendering time under different render image resolutions compared with volume rendering in Table 7 of the main paper. We conducted the extra experiment for render time comparison under the same render image resolution (180×320) but with different Gaussians number (voxel resolutions) as shown in Table 1. From Table 7 of the main paper and Table 1, we observe: (1) The render time of splatting rendering (SR) is mainly affected by the Gaussians number, not the render image resolution. (2) SR is 3–8 times faster than volume rendering (VR) across different voxel settings.

Bonus of the fully self-supervised setting: The fully self-supervised setting of our method could be a general pretraining solution for supervised learning. After the self-supervised training on the DDAD and nuScenes datasets, we further finetune the model with the 3D occupancy label from Occ3D [13]. As shown in Table 4, experiments with self-supervised pretraining outperform the baseline. In particular, we find that pretraining on nuScenes is better than the DDAD dataset, which may own to the domain gap factors, such as differences in the scenarios (RGB images) and sensor configurations (camera extrinsics).

3. Limitation and future work

The proposed method achieves reasonable predictions in most scenes; however, we observe that some cases still present challenges, as shown in Figure 5. Specifically, in the DDAD dataset, incorrect predictions occur in the back camera in certain situations as marked with the red circle, where the drivable surface is mistakenly projected into the car due to extensive self-occlusion. Notably, this issue is absent in the nuScenes dataset, which has less self-occlusion. We believe that this problem could be mitigated with better 2D semantic maps for supervision, which warrants further investigation. The proposed method is for the surround view setting which is not suitable for the monocular images. Additionally, in stage 1, we leverage the spatial cross-view constraint for scale-aware training through the proposed Gaussian splatting method. In the future, we aim to explore its potential benefits for temporal view synthesis as well.

References

[1] Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000. 1

[2] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022. 4

[3] Wanshui Gan, Ningkai Mo, Hongbin Xu, and Naoto Yokoya. A comprehensive framework for 3d occupancy estimation in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2024. 4, 7

[4] Vitor Guizilini, Rares Ambrus, Sudeep Pillai, Allan Raventos, and Adrien Gaidon. 3d packing for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2485–2494, 2020. 1

[5] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 4

[6] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 4

[7] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, pages 9223–9232, 2023. 4

Method	GT Occ.	GT Pose	mIoU*	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	sidewalk	terrain	manmade	vegetation
MonoScene [2]	✓	×	6.33	6.06	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	7.92	7.43	1.01	7.65
BEVDet [6]	✓	×	20.03	19.38	30.31	0.23	32.26	34.47	12.97	10.34	10.36	6.26	8.93	23.65	52.27	26.06	22.31	15.04	15.10
BEVFormer [9]	✓	×	24.64	23.67	38.79	9.98	34.41	41.09	13.24	16.50	18.15	17.83	18.66	27.70	48.95	29.08	25.38	15.41	14.46
OccFormer [17]	✓	×	22.39	21.93	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	34.66	22.73	6.76	6.97
RenderOcc [12]	✓	×	24.53	23.93	27.56	14.36	19.91	20.56	11.96	12.42	12.14	14.34	20.81	18.94	68.85	42.01	43.94	17.36	22.61
TPVFormer [7]	✓	×	28.69	27.83	38.90	13.67	40.78	45.90	17.23	19.99	18.85	14.30	26.69	34.17	55.65	37.55	30.70	19.40	16.78
CTF-Occ [13]	✓	×	**29.54**	**28.53**	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	37.98	33.23	20.79	18.00
SimpleOcc [3]	×	✓	7.99	7.05	0.67	1.18	3.21	7.63	1.02	0.26	1.80	0.26	1.07	2.81	40.44	18.30	17.01	13.42	10.84
SelfOcc [8]	×	✓	10.54	9.30	0.15	0.66	5.46	12.54	0.00	0.80	2.10	0.00	0.00	8.25	55.49	26.30	26.54	14.22	5.60
OccNeRF [16]	×	✓	10.81	9.54	0.83	0.82	5.13	12.49	3.50	0.23	3.10	1.84	0.52	3.90	52.62	20.81	24.75	18.45	13.19
GaussianOcc	×	×	**11.26**	**9.94**	1.79	5.82	14.58	13.55	1.30	2.82	7.95	9.76	0.56	9.61	44.59	20.10	17.58	8.61	10.29

Table 2. **3D occupancy prediction performance on the Occ3D-nuScenes dataset in mIoU metric.** Since 'other' and 'other flat' classes are the invalid prompts for open-vocabulary models, we also calculate 'mIoU*' as the result ignoring the classes that do not consider these two classes during evaluation, while 'mIoU' is the original result. GT Occ. refers to the use of the ground truth occupancy label for supervision. GT Pose is the ground truth pose from the sensor for self-supervised geometry learning.

Method	GT Occ.	GT Pose	Backbone	Input Size	Epoch	RayIoU	RayIoU$_{1m, 2m, 4m}$			mIoU	FPS
BEVFormer (4f) [9]	✓	×	R101	1600×900	24	32.4	26.1	32.9	38.0	39.2	3.0
RenderOcc [12]	✓	×	Swin-B	1408×512	12	19.5	13.4	19.6	25.5	24.4	-
SimpleOcc [3]	✓	×	R101	672×336	12	28.2	22.3	28.7	33.7	37.3	9.7
BEVDet-Occ (2f) [5]	✓	×	R50	704×256	90	29.6	23.6	30.0	35.1	36.1	2.6
BEVDet-Occ-Long (8f)	✓	×	R50	704×384	90	32.6	26.6	33.1	38.2	**39.3**	0.8
FB-Occ (16f) [10]	✓	×	R50	704×256	90	33.5	26.7	34.1	39.7	39.1	10.3
SparseOcc (8f)	✓	×	R50	704×256	24	34.0	28.0	34.7	39.4	30.1	**17.3**
SparseOcc (16f)	✓	×	R50	704×256	48	**36.1**	**30.2**	**36.8**	**41.2**	30.9	12.5
OccNeRF [16]	×	✓	R101	640×384	12	10.49	6.93	10.28	14.26	9.54	10.8
GaussianOcc	×	×	R101	640×384	12	**11.85**	**8.69**	**11.90**	**14.95**	**9.94**	10.8

Table 3. **3D Occupancy prediction performance on the Occ3D-nuScenes dataset in RayIoU metric.** GT Occ. means using the ground truth occupancy label for the supervision. GT Pose is the ground truth pose from the sensor for self-supervised geometry learning. "8f" and "16f" mean fusing temporal information from 8 or 16 frames. mIoU is the mean Intersection over Union for all categories. FPS means frame per second for each method, which is measured on a Tesla A100 GPU.

[8] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19946–19956, 2024. 4

[9] 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 multi-camera images via spatiotemporal transformers. In *ECCV*, 2022. 4

[10] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023. 4

[11] Haisong Liu, Haiguang Wang, Yang Chen, Zetong Yang, Jia Zeng, Li Chen, and Limin Wang. Fully sparse 3d panoptic occupancy prediction. *arXiv preprint arXiv:2312.17118*, 2023. 2

[12] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*, 2023. 4

[13] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023. 2, 3, 4

[14] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Yong-

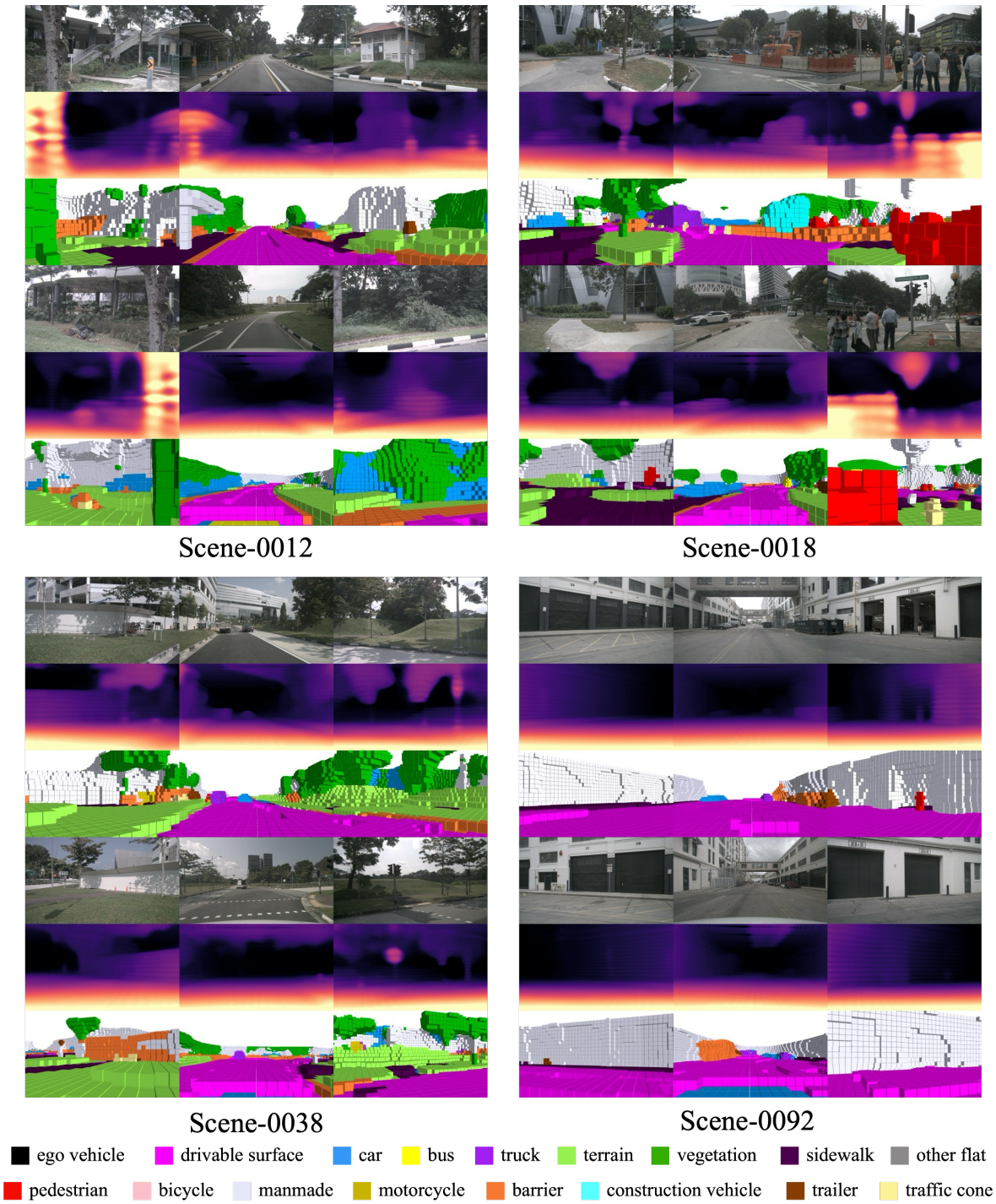

Figure 3. **The visualization of the render depth map and 3D occupancy prediction on nuScenes dataset.**

ming Rao, Guan Huang, Jiwen Lu, and Jie Zhou. Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation. In *CoRL*, pages 539–549. PMLR, 2023. 2

[15] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. Newcrfs: Neural window fully-connected crfs for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 1

[16] Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural

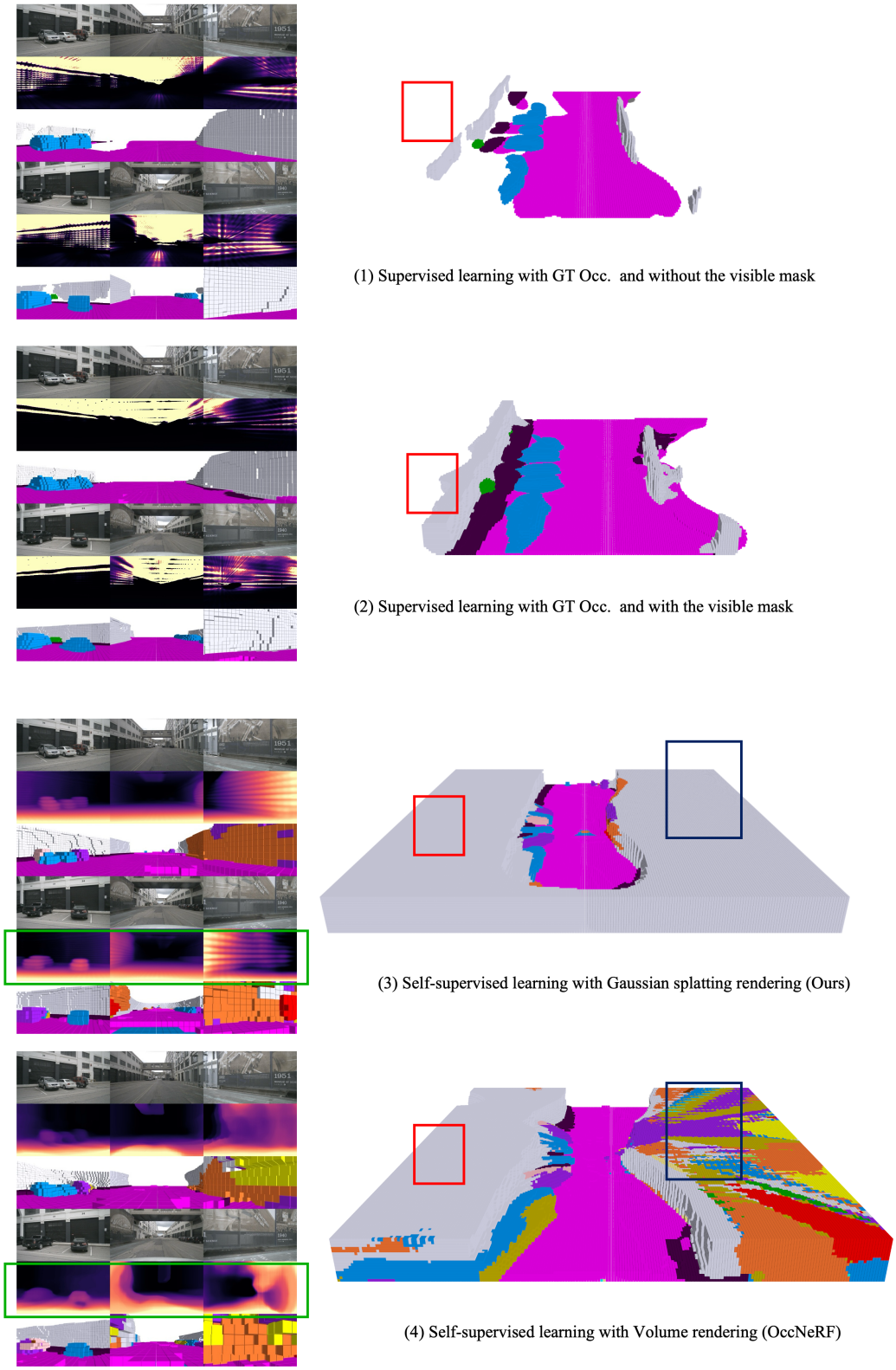

Figure 4. **The visualization of the different supervision types (1-4) comparison on nuScenes dataset.**

Method	Pretraining setting	mIoU	RayIoU	$\text{RayIoU}_{1m, 2m, 4m}$		
Baseline	None	37.29	28.2	22.3	28.7	33.7
Self-supervised pretrain	DDAD	37.40	28.7	22.9	29.1	34.0
	nuScenes	**38.45**	**29.9**	**23.9**	**30.4**	**35.5**

Table 4. The study on SimpleOcc [3] with fully self-supervised pretrain. The baseline is directly training the model with 3D occupancy label. The self-supervised pretraining is conducted on DDAD and nuScenes and then finetuned the model with 3D occupancy label. The number with bold typeface means the best.

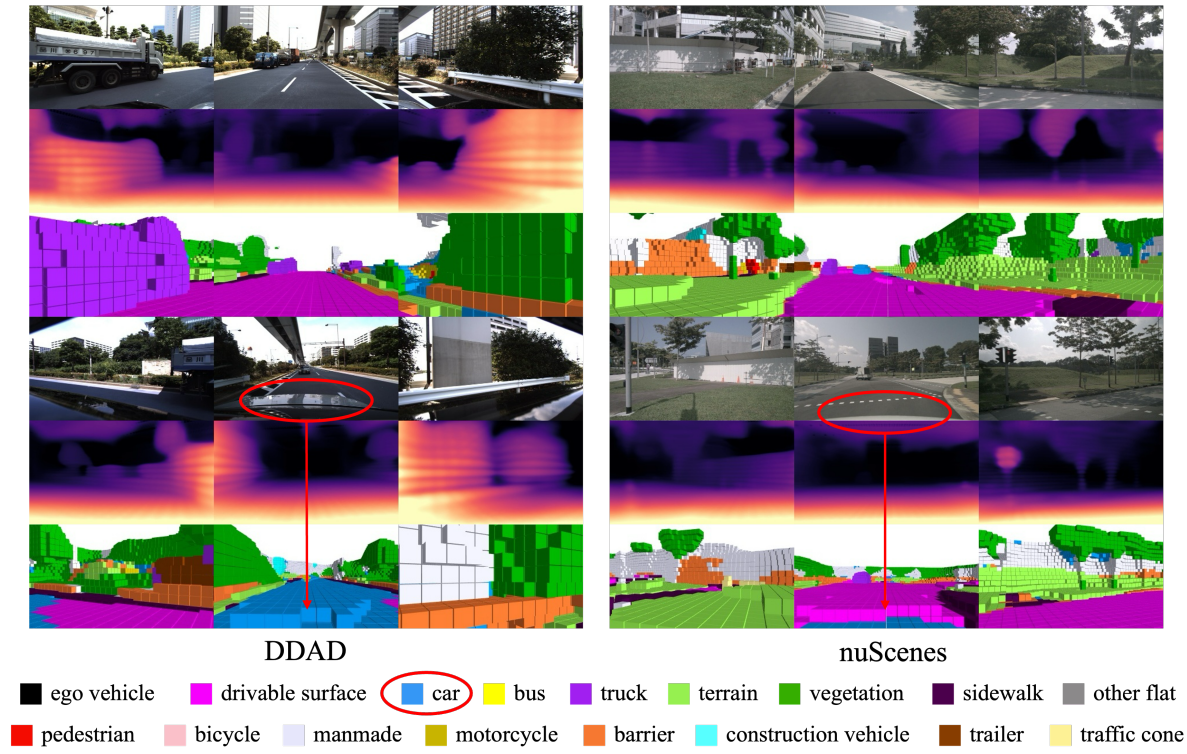

Figure 5. **Some wrong predictions due to the large self-occlusion on DDAD dataset.**

radiance fields. *arXiv preprint arXiv:2312.09243*, 2023. 1, 2, 3, 4

[17] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *ICCV*, 2023. 4

[18] Shunyuan Zheng, Boyao Zhou, Ruizhi Shao, Boning Liu, Shengping Zhang, Liqiang Nie, and Yebin Liu. Gps-gaussian: Generalizable pixel-wise 3d gaussian splatting for real-time human novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19680–19690, 2024. 1