2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

RenderOcc: Vision-Centric 3D Occupancy Prediction with 2D Rendering Supervision

Mingjie Pan[1,2*], Jiaming Liu[1*], Renrui Zhang[3*], Peixiang Huang[2],
Xiaoqi Li[1], Hongwei Xie[4], Bing Wang[5], Li Liu[2†], Shanghang Zhang[1†]

***Abstract*— 3D occupancy prediction holds significant promise in the fields of robot perception and autonomous driving, which quantifies 3D scenes into grid cells with semantic labels. Recent works mainly utilize complete occupancy labels in 3D voxel space for supervision. However, the expensive annotation process and sometimes ambiguous labels have severely constrained the usability and scalability of 3D occupancy models. To address this, we present RenderOcc, a novel paradigm for training 3D occupancy models only using 2D labels. Specifically, we extract a NeRF-style 3D volume representation from multi-view images, and employ volume rendering techniques to establish 2D renderings, thus enabling direct 3D supervision from 2D semantics and depth labels. Additionally, we introduce an Auxiliary Ray method to tackle the issue of sparse viewpoints in autonomous driving scenarios, which leverages sequential frames to construct comprehensive 2D rendering for each object. To our best knowledge, RenderOcc is the first attempt to train multi-view 3D occupancy models only using 2D labels, reducing the dependence on costly 3D occupancy annotations. Extensive experiments demonstrate that RenderOcc achieves comparable performance to models fully supervised with 3D labels, underscoring the significance of this approach in real-world applications. Our code is available at https://github.com/pmj110119/RenderOcc.**

I. Introduction

Perceiving the 3D world plays an important role in vision-based robotic systems and autonomous driving [1], [2]. One of the currently popular tasks in 3D vision is semantic occupancy prediction. It requires quantifying continuous 3D space into grid cells and predicting semantic labels for each voxel. Compared to 3D object detection [3], [4], 3D occupancy provides more fine-grained geometric perception, allowing for a detailed understanding of object shapes and scene-level geometries, rather than the coarse-grained 3D bounding box. To this point, many solutions have been proposed and gained widespread adoption in both industry and academia [5], [6].

Existing methods mostly rely on complete 3D occupancy labels for supervision. However, directly annotating 3D occupancy is extremely challenging and expensive. For one thing, after creating $\sim$30000 frames of 3D occupancy labels based on other pre-annotated 3D labels, it still requires costly $\sim$4000 human hours for purification [7]. For another, the complexity of 3D space would cause ambiguous labels, distracting the training of occupancy models. We conducted a comparison on several benchmarks and found that even when they use identical raw data, the occupancy labels they produce still exhibit a 10-15% difference [7], [8], [9]. This reflects the inherent ambiguity of occupancy annotation, which further restricts the practical application of 3D occupancy tasks in real-world scenarios.

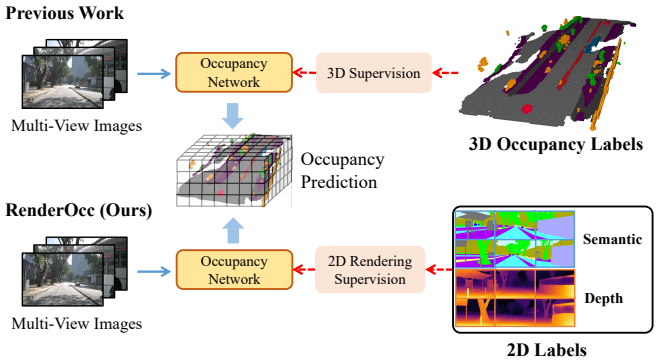

Fig. 1. **RenderOcc represents a new training paradigm.** Unlike previous works that focus on supervising with costly 3D occupancy labels, our proposed RenderOcc utilizes 2D labels to train the 3D occupancy network. Through 2D rendering supervision, the model benefits from fine-grained 2D pixel-level semantic and depth supervision.

Towards these issues aforementioned, we introduce RenderOcc, a novel paradigm for training 3D occupancy models using 2D labels, free from any 3D-space annotations. As shown in Fig. 1, the goal of RenderOcc is to eliminate the dependency on 3D occupancy labels and rely solely on pixel-level 2D semantics for network supervision during training. In particular, it constructs a NeRF-style 3D volume representation from multi-view images and utilizes advanced volume rendering techniques to generate 2D renderings. This approach enables us to provide direct 3D supervision using only 2D semantics and depth labels. With such a 2D rendering supervision, the model learns multi-view consistency by analyzing intersecting frustum rays from various cameras, obtaining a deeper understanding of geometric relationships in 3D space. Importantly, it is worth noting that autonomous driving scenarios often involve limited viewpoints, which can hinder the effectiveness of rendering supervision. Considering this, we introduce the concept of Auxiliary-Ray, which leverages rays from adjacent frames to enhance the multi-view consistency constraints for the current frame. Moreover, we have developed a dynamic sampling training strategy for auxiliary rays, which not only screens out misaligned rays, but also simultaneously mitigates the additional training costs associated with them.

[1] Mingjie Pan, Jiaming Liu, Xiaoqi Li and Shanghang Zhang are with National Key Laboratory for Multimedia Information Processing, School of CS, Peking University. [2] Mingjie Pan, Peixiang Huang and Li Liu are with Xiaomi Car. [3] Renrui Zhang is with CUHK MMLAB. [4] Hongwei Xie is with Nanjing University. [5] Bing Wang is with Nanyang Technological University.

$*$ The first three authors contributed equally.

† Corresponding to shanghang@pku.edu.cn or liuli.ll9412@gmail.com.

To evaluate the effectiveness of our proposed method, we conduct extensive experiments on two widely recognized benchmarks, NuScenes [10] and SemanticKiTTI [11]. Remarkably, with only 2D supervision, our method achieved competitive performance compared with models supervised by 3D labels. This highlights the considerable potential of our approach in real-world perception systems. Meanwhile, our method outperforms previous state-of-the-art 3D occupancy prediction methods by leveraging both 2D labels and corresponding 3D labels. The main contributions are summarized as follows:

1) We introduce RenderOcc, a 3D occupancy framework based on 2D rendering supervision. We make the first attempt to train multi-view 3D occupancy networks solely using 2D labels, discarding the costly and challenging 3D annotation.

2) To learn a favorable 3D voxel representation from limited viewpoints, we introduce Auxiliary-Rays to tackle the challenge of sparse viewpoints in autonomous driving scenarios. Meanwhile, we design a dynamic sampling training strategy for balancing and purifying auxiliary rays.

3) Extensive experiments show that RenderOcc achieves competitive performance when using only 2D labels, compared to baselines that supervised by 3D labels. This showcases the feasibility and potential of 2D image supervision for 3D occupancy training.

II. Related work

3D Object Detection. 3D object detection is a classic perception task in the fields of robot perception and autonomous driving [12], [13], [14], [15], [16], [17]. Recently, vision-based 3D object detection has gained increased attention due to its low cost and rich semantic content, [4], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27] realize efficient transformation from multiple perspective views to a unified 3D space in a single frame, achieving cross-view 3D perception.

3D Occupancy Prediction. 3D occupancy prediction, which can generate a dense 3D voxelized semantic representation of a scene, is an ideal capability for autonomous vehicles. The release of the SemanticKITTI dataset has drawn attention to 3D occupancy prediction [28], [29], [30], [31], [32], [33], but it lacks diversity in driving scenes and only evaluates front-view predictions [11]. Recent work has extended 3D occupancy prediction to multi-view surrounding scenes and produce large-scale benchmarks, which have significantly propelled the development of the field [8], [7], [9], [6]. Due to the difficulty of manually annotating dense 3D occupancy, existing work relies on extra 3D labels such as LiDAR segmentation to produce 3D occupancy labels Recent works [34], [35] discussed occupancy estimation based on rendering, but they do not account for semantic predictions.

3D Reconstruction and Rendering Inferring the 3D geometry of objects or scenes from 2D images is a challenging task. Recent popular methods model 3D scenes through neural radiance fields and supervise them using volume rendering based on multi-view 2D images [36], [37], [38], [39]. To improve training efficiency, significant results have been achieved by further using voxel-based explicit representations [40], [41], [42], [43]. Unlike 3D Occupancy Prediction, these methods focus on rendering quality, pay less attention to semantic understanding, and lack generalization. However, their training ideas can be enlightening for 3D occupancy.

III. Methods

A. Problem Setup

We aim to predict a dense semantic volume, termed as 3D occupancy, of surrounding scenes with multi-camera RGB images. Specifically, for the vehicle at timestamp t, we take N images $\{I^1, I^2, \cdots I^N\}$ as input and predict the 3D occupancy $O \in \mathbb{R}^{H\times W\times D\times L}$ as output, where H, W, D denote the resolution of the volume and L denotes the number of categories (including empty). Formally, the 3D occupancy prediction can be formulated as

$$V = \mathbb{G}(I^1, I^2, \cdots, I^N), \quad O = \mathbb{F}(V), \tag{1}$$

where $\mathbb{G}$ is an neural network that extracts 3D volume feature $V \in \mathbb{R}^{H\times W\times D\times C}$ from N-view images, where C denotes the feature dimension. $\mathbb{F}$ is responsible for transforming V into occupancy representation, for which previous works [7], [8] tend to use MLP to achieve per-voxel classification. Considering that all existing approaches require complete 3D occupancy labels to supervise the voxel-level classification, we design a new concept to implement $\mathbb{F}$ and supervise $\{\mathbb{G}, \mathbb{F}\}$ with only 2D pixel-level labels.

B. Overall Framework

Our overall framework is shown in Fig. 2. In Sec. III-C, we first extract 3D volume features V from multi-view RGB images with a 2D-to-3D network $\mathbb{G}$. Note that our framework is insensitive to the implementation of $\mathbb{G}$ and can flexibly switch between various BEV/Occupancy encoders such as [18], [19], [31]. Next in Sec. III-D, we predict a volume density σ and semantic logits S for each voxel to generate the semantic density field (SDF). Subsequently, we perform volume rendering from SDF and optimize the network with 2D labels. Finally in Sec. III-E, we illustrate the Auxiliary-Ray training strategy for volume rendering to address the sparse viewpoints issue in autonomous driving scenarios.

C. Semantic Density Field

Existing 3D occupancy methods extract volume features V from multi-view images and perform voxel-wise classification [31], [32], [9], [6] to generate 3D semantic occupancy. To employ 2D pixel-level supervision, our RenderOcc innovatively transforms V into a versatile representation termed the Semantic-Density-Field (SDF). Given a volume feature map $V \in \mathbb{R}^{H\times W\times D\times C}$, the SDF encodes the scene by two representations: volume density $\sigma \in \mathbb{R}^{H\times W\times D}$ and semantic logits $S \in \mathbb{R}^{H\times W\times D\times L}$. Specifically, we simply adopt two MLPs $\{\phi_d, \phi_s\}$ to construct the SDF, formulated as

$$\sigma = softplus(\phi_d(V)); \quad S = \phi_s(V), \tag{2}$$

where σ additionally employs softplus activation to ensure that density values do not become negative. Based on SDF,

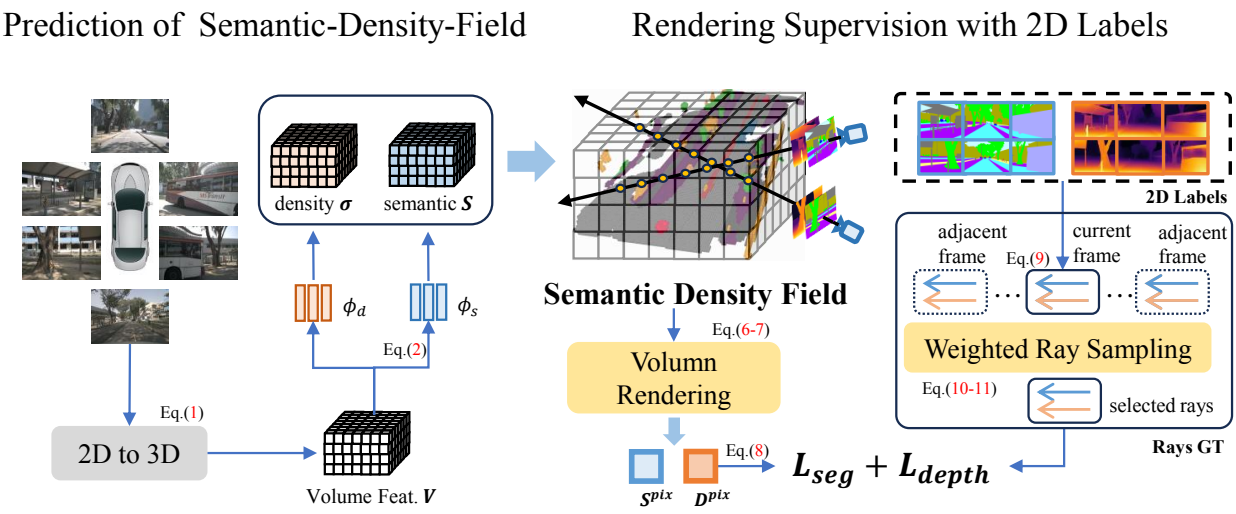

Fig. 2. **Overall framework of RenderOcc.** We extract volume features V and predict density σ and semantic S for each voxel through a 2D-to-3D network. As a result, we generate the Semantic Density Field, which can perform volume rendering to generate rendered 2D semantics and depth $\{S^{pix}, D^{pix}\}$. For the generation of Rays GT, we extract auxiliary rays from adjacent frames to supplement the rays of the current frame, and purify them using the proposed Weighted Ray Sampling strategy. Then, we calculate the loss with rays GT and $\{S^{pix}, D^{pix}\}$, achieving rendering supervision with 2D labels.

we gain the capability to perform semantic rendering from any viewpoints and get 2D supervision for optimization during training, which will be explained in Sec. III-D.

After optimized, SDF can be directly converted to 3D occupancy results. We filter out occupied voxels with σ and determine their semantic categories based on S. The process can be formalized as follows:

$$O(x,y,z) = \begin{cases} argmax(S(x,y,z)), & \sigma(x,y,z) \geq \tau \\ empty\ label, & \sigma(x,y,z) < \tau \end{cases}, \quad (3)$$

where τ serves as the threshold value of σ determining whether a voxel is occupied.

D. Rendering Supervision with 2D Labels

We utilize volume rendering to form a bridge between the SDF and 2D pixels, thereby facilitating the supervision through 2D labels. Specifically, we extract 3D rays from the current frame using camera intrinsic and extrinsic parameters, with each 2D pixel corresponding to a 3D ray originating from the camera. Each ray $\mathbf{r}$ carries the semantic and depth labels $\{\hat{S}^{pix}(r), \hat{D}^{pix}(r)\}$ of the corresponding pixel. Meanwhile, we perform volume rendering [44] based on the SDF to obtain the rendered semantic $S^{pix}(r)$ and depth $D^{pix}(r)$, which are used to compute the loss with 2D labels $\{\hat{S}^{pix}(r), \hat{D}^{pix}(r)\}$.

To render the semantic and depth of a pixel, K points $\{z_k\}_{k=1}^{K} \in r$ are sampled on the ray r in a pre-defined range. Then the accumulated transmittance T and the probability of termination α of the point z_k can be computed by

$$\alpha(z_k) = 1 - exp(-\sigma(z_k)\beta_k), \quad (4)$$

$$T(z_k) = exp(-\sum_{t=1}^{k-1} \sigma(z_t)\beta_t), \quad (5)$$

where $\beta_k = z_{k+1} - z_k$ is the distance between two adjacent points. Finally, we query the SDF with $\{z_k\}$ and accumulated

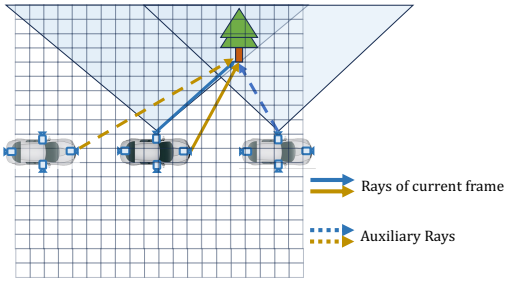

Fig. 3. **Auxiliary Rays:** Images from single frame cannot capture multi-view information of objects well. There is only a small overlap area between two adjacent cameras, and the difference in perspective is limited. By introducing auxiliary rays from adjacent frames, the model will significantly benefit from multi-view consistency constraints.

them to get rendered semantic and depth:

$$S^{pix}(r) = \sum_{k=1}^{N} T(z_k)\alpha(z_k)S(z_k), \quad (6)$$

$$D^{pix}(r) = \sum_{k=1}^{N} T(z_k)\alpha(z_k)z_k, \quad (7)$$

For loss fuctions, cross-entropy loss L_{seg} and SILog loss L_{depth} [45] are leveraged to supervise the semantic and depth, respectively. We also introduce distortion loss [39] and TV loss [46] as the regularization of SDF, which termed L_{reg}. Therefore, the overall loss can be computed by

$$L = L_{seg}(S^{pix}, \hat{S}^{pix}) + L_{depth}(D^{pix}, \hat{D}^{pix}) + L_{reg}(\sigma), \quad (8)$$

E. Auxiliary Rays: Boosting Multi-view Consistency

With the 2D rendering supervision in Sec. III-D, the model can benefit from multi-view consistency constraints and learn to account for spatial occlusion relationships among voxels. However, the viewpoint coverage of the

surrounding cameras in a single frame is very sparse, and their overlapping range is limited. As a result, most voxels cannot be simultaneously sampled by multiple rays with significant viewpoint differences, which easily leads to local optima. Therefore, we introduce auxiliary rays from adjacent frames to complement the multi-view consistency constraints as shown in Fig. 3.

Generation of Auxiliary Rays. Specifically, for the current frame with index t, we select nearby M_{aux} adjacent frames. For each adjacent frame, we generate rays individually and transform them to the current frame to obtain the final auxiliary rays r_{aux}:

$$r_{aux} = \{\mathrm{T}_t^{t-k}(r_{t-k}), \quad k = -1, 1, \cdots, -\frac{M_{\text{aux}}}{2}, \frac{M_{\text{aux}}}{2}\} \tag{9}$$

where T_t^{t-k} is the transformation matrix from adjacent-frame coordinates to the current frame. Given ego pose matrices E_{t-k} from adjacent frame and E_t from the current frame, we can calculate $\mathrm{T}_t^{t-k} = E_t^{inv} \cdot E_{t-k}$, where E_t^{inv} is the inverse matrix of E_t.

Weighted Ray Sampling. The introduction of auxiliary rays significantly enhances 2D supervision but raises two challenges: (a) Increased rays lead to high memory and computational costs, necessitating random sampling during training, which discards many valuable rays. (b) Due to the presence of dynamic objects, many auxiliary rays exhibit temporal mismatches, introducing unnecessary errors. To address this, we have devised the Weighted Ray Sampling strategy, focusing on sampling high-information-density and relative correct rays. This not only significantly enhances training efficiency but also optimizes performance. **(a) Category Density Balance:** In outdoor autonomous driving, extreme category imbalance is common. Most rays correspond to large, low-information-density background objects like roads and buildings, while rays associated with pedestrians, bicycles are scarce but valuable. For this, we calculate weight W_b based on class occurrence frequency, formulated as

$$W_b(r) = \exp(\lambda_s * (\frac{max(M)}{N(\mathbf{C}(r))} - 1)) \tag{10}$$

where λ_s represents a smoothing coefficient, M represents the numbers of rays of all categories, and $\mathbf{C}(r)$ denotes the category of the ray r. **(b) Temporal Misalignment Purification:** By Eq. 9, auxiliary rays can be aligned with the current frame for SDF supervision. However, the movement of dynamic objects can cause misalignment, leading some auxiliary rays to point to incorrect voxels in the SDF. We employed a simple yet effective strategy to reduce the sampling probability of mismatched rays, by masking dynamic objects and preserving the rays from the current frame as much as possible.

$$W_t(r) = \begin{cases} \lambda_{\text{dyn}}, & C(r) \in C_{\text{dynamic}} \\ \lambda_{\text{adj}}, & C(r) \notin C_{\text{dynamic}} \end{cases} \tag{11}$$

where λ_{dyn} and λ_{adj} are coefficients less than 1 and C_{dynamic} denotes the set of categories for dynamic objects. By utilizing λ_{adj}, we can lower the likelihood of rays from the current frame being omitted. Setting λ_{dyn} to a value close to 0 allows us to significantly mitigate misalignment issues caused by dynamic objects.

Finally, we calculate a weight W for each ray as $W = W_b \cdot W_t$. This weight acts as the probability weight of random sampling. Throughout the training process, we sample a fix number of rays for each batch using W, and the remaining rays are discarded and do not contribute to the loss computed by Eq. 8. Through proposed Weighted Ray Sampling, we can significantly reduce memory and computational costs of RenderOcc during training while achieving superior performance. Detailed discussions are provided in Sec. IV-D.

IV. Experiments

We evaluate proposed RenderOcc by comparing it with other baselines on NuScenes and SemanticKiTTI. For a deeper understanding of RenderOcc, we also conducted extensive ablation experiments on NuScenes dataset in Sec. IV-D.

A. Dataset

We evaluate our method on NuScenes and SemanticKiTTI, respectively for surrounding-view and front-view setting. The NuScenes dataset includes 1000 outdoor driving scenes with six surrounding-view cameras, and the 3D occupancy ground truth provided by [8] of each sample covers a range of [-40m, -40m, -1m, 40m, 40m, 5.4m] with a voxel size of [0.4m,0.4m,0.4m]. It has 17 classes. The SemanticKiTTI dataset includes 22 outdoor driving scenes, and the benchmark is interested in areas ahead of the car. Each sample covers a range of [0.0m, -25.6m, -2.0m, 51.2m, 25.6m, 4.4m] with a voxel size of [0.2m,0.2m,0.2m]. It has 19 classes. Neither NuScenes nor SemanticKITTI directly provide ground truth (GT) for 2D segmentation and depth. Therefore, we project 3D LiDAR points with segmentation labels onto images to generate 2D labels.

B. Architecture and Implementation Details

We use the available network BEVStereo [21] as $\mathbb{G}$ to test the performance of RenderOcc. We maintain the majority of the original structure and only replace the classification head $\mathbb{F}$ with $\{\phi_d, \phi_s\}$ to predict the Semantic Density Field, thereby supporting the training by 2D rendering supervision. For $\mathbb{G}$, we use Swin Transformer [48] as the image backbone, resize the image to 512x1408, use Adam as the optimizer, set the Batchsize to 16, and train approximately $\sim 10K$ iters with a learning rate of 1e-4. All experiments are conducted on NVIDIA A100 GPUs.

C. Main Results

Results on NuScenes. In this section, we provide experimental results on the NuScenes dataset, as shown in Tab. I. We employs 6 adjacent frames to generate auxiliary rays and sample 38400 rays for a batch. RenderOcc, when supervised solely with 2D labels, achieves a mean mIoU of 23.93, demonstrating competitive results when compared to Baselines supervised by 3D occupancy labels. In details, RenderOcc exhibits only a marginal decline of 0.58 mIoU

TABLE I

3D OCCUPANCY PREDICTION PERFORMANCE ON THE OCC3D-NUSCENES DATASET. GT REPRESENTS WHICH LABELS WE USED DURING TRAINING. MEAN IS THE AVERAGE VALUE OF MIOU ACROSS ALL CATEGORIES.

Method	GT	Mean	Others	barrier	bicycle	bus	car	cons. veh	motorcycle	pedestrian	traffic cone	trailer	truck	dri. sur	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [28]	3D	6.06	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65
BEVFormer	3D	23.67	5.03	38.79	9.98	34.41	41.09	13.24	16.50	18.15	17.83	18.66	27.70	48.95	27.73	29.08	25.38	15.41	14.46
BEVStereo	3D	24.51	5.73	38.41	7.88	38.70	41.20	17.56	17.33	14.69	10.31	16.84	29.62	54.08	28.92	32.68	26.54	18.74	17.49
OccFormer [47]	3D	21.93	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97
RenderOcc (Ours)	2D	23.93	5.69	27.56	14.36	19.91	20.56	11.96	12.42	12.14	14.34	20.81	18.94	68.85	33.35	42.01	43.94	17.36	22.61
RenderOcc*	2D+3D	26.11	4.84	31.72	10.72	27.67	26.45	13.87	18.2	17.67	17.84	21.19	23.25	63.2	36.42	46.21	44.26	19.58	20.72

TABLE II

3D OCCUPANCY PREDICTION PERFORMANCE ON THE SEMANTICKITTI DATASET.

Method	GT	Mean	car	bicycle	motorcycle	truck	other-veh.	person	bicyclist	motorcyclist	road	parking	sidewalk	other-grnd	building	fence	vegetation	trunk	terrain	pole	traf.-sign
MonoScene	3D	11.30	23.29	0.28	0.59	9.29	2.63	2.00	1.07	0.00	55.89	14.75	26.50	1.63	13.55	6.60	17.98	2.44	29.84	3.91	2.43
LMSCNet [29]	3D	9.94	23.62	0.00	0.00	1.69	0.00	0.00	0.00	0.00	54.90	9.89	25.43	0.00	14.55	3.27	20.19	1.06	32.30	2.04	0.00
AICNet [30]	3D	6.73	15.30	0.00	0.00	0.70	0.00	0.00	0.00	0.00	39.30	19.80	18.30	1.60	9.60	5.00	9.60	1.90	13.50	0.10	0.00
VoxFormer [31]	3D	12.35	25.79	0.59	0.51	5.63	3.77	1.78	3.32	0.00	54.76	15.50	26.35	0.70	17.65	7.64	24.39	5.08	29.96	7.11	4.18
RenderOcc (Ours)	2D	8.24	14.83	0.42	0.17	2.47	1.78	0.94	3.20	0.00	43.64	12.54	19.10	0.00	11.59	4.71	17.61	1.48	20.01	1.17	0.88
RenderOcc*	2D+3D	12.87	24.90	0.37	0.28	6.03	3.66	1.91	3.11	0.00	57.2	16.11	28.44	0.91	18.18	9.10	26.23	4.87	33.61	6.24	3.38

compared with BEVStereo, surpassing other baselines such as MonoScene, OccFormer and BEVFormer. Under our proposed 2d rendering supervision, RenderOcc performs significant variations in its performance across different categories compared to 3D supervised baselines. For static background categories like driveable surfaces, terrain, and vegetation, RenderOcc achieves exceptionally high mIoU, effectively identifying road structures, benefiting from an explicit understanding of 3D spatial relationships. However, RenderOcc's performance is relatively poorer for dynamic objects, often predicting artifacts such as ghosting. Our proposed Auxiliary Rays strategy can alleviate this issue to some extent. For small foreground objects like bicycles, motorcycles, and traffic cones, RenderOcc also outperforms Baselines, thanks to the fine-grained supervision provided by 2D pixel-level labels. Additionally, RenderOcc supports simultaneous supervision using both 2D and 3D labels, achieving the best result with an mIoU of 26.11. The result demonstrates that our method of 2D rendering supervision can aid in improving the construction of voxel representations for existing 3D occupancy labels and enhance overall understanding of 3D scenes. Finally, we conducted a qualitative analysis in Fig .4, and observed that our method yields improved accuracy in capturing semantic information and the shapes of objects within the 3D scene.

Results on SemanticKITTI. In this section, we provide experimental results on the SemanticKITTI dataset, as shown in Tab. II. We employs 4 adjacent frames to generate auxiliary rays and sample 25600 rays for a batch. RenderOcc achieves a mean mIoU of 8.24 supervised solely using 2D label, showing competitive result with other baseline methods that supervised by 3D occupancy GT. Unfortunately, SemanticKITTI relies on a single front-facing camera, leading

Fig. 4. **Qualitative results on NuScenes.** Compared to the baseline that uses 3D labels for supervision, our proposed RenderOcc exhibits a more acute perception of object boundaries and small objects as shown in the red boxes. The crane's arm in the image is finely perceived by RenderOcc, while BEVStereo supervised by 3D labels fails to perceive the arm floating in the air. At the same time, RenderOcc successfully identifies distant traffic cones that the baseline overlooks.

to a lack of multi-view consistency constraints due to its fixed and singular viewpoint. This limitation causes 2D rendering supervision to become trapped in local optima. While auxiliary rays strategy partially mitigate this issue, it validates the effectiveness of our approach in monocular 3D occupancy prediction. Furthermore, when incorporating existing 3D labels, RenderOcc overcomes the optimization problem stemming from the single viewpoint, resulting in a 12.87 mIoU. This result showcases the practicality of our method in real-world applications, whether the input images are from multi-view cameras or single-view camera.

TABLE III

ABLATION STUDY FOR EACH COMPONENT ON THE OCC3D-NUSCENES.

RenSup-S	RenSup-D	Aux Rays	WRS	mIoU ↑
✓				16.94
✓	✓			19.28
✓	✓	✓		22.41
✓	✓	✓	✓	**23.93**

TABLE IV

ABLATION STUDY FOR RENDERING SAMPLING WAYS.

Method	step size	point number	mIoU ↑
unified-sampling	1.0*voxel_size	/	19.42
unified-sampling	0.5*voxel_size	/	21.10
hierarchical-sampling	/	64→64	22.34
hierarchical-sampling	/	64→128	22.98
mip360-sampling	1.0*voxel_size	/	22.61
mip360-sampling	0.5*voxel_size	/	**23.93**

TABLE V

ANALYSIS OF DEPTH SUPERVISION. "LIDAR PROJECTION" REFERS EXTRACTING DEPTH LABELS FROM LIDAR POINT CLOUDS. "STRUCT FROM MOTION" REPRESENTS DERIVING DEPTH LABELS FROM IMAGE-BASED ALGORITHMS [49] WITHOUT LIDAR DATA.

Depth Label	Label Modality	mAP ↑
/	/	16.92
LiDAR Projection	L	**23.93**
Struct From Motion	C	21.11

D. Ablation Study & Analysis

Ablation Study for each Component To clarify RenderOcc's component contributions, we present ablation results in Table III. When supervised solely with 2D semantic labels (RenSup-S), RenderOcc achieves an mIoU of 16.94. Adding depth supervision (RenSup-D) increases mIoU by 2.34. Introducing the auxiliary rays strategy (Aux Rays) further boosts performance by 3.13 mIoU but at the cost of increased training overhead. Utilizing the Weighted Ray Sampling strategy yields a total mIoU improvement of 1.52. Importantly, WRS effectively addresses the training overhead introduced by auxiliary rays, which we will discuss in detail later.

Selection for Points Sampling Ways We compared three typical point sampling methods for z_k and make corresponding adjustments to the rendering formula., namely unified-sampling (perform sampling with a fixed step size) , hierarchical-sampling [36] and mip360-sampling (introduced in mip-NeRF 360 [39] to address unbounded scenes). As shown in Table IV, both hierarchical-sampling and MN360-sampling methods outperform unified-sampling. Moreover, with an increase in sampling frequency (step size), there is a noticeable improvement in performance. Therefore, we finally select mip360-sampling for volumn rendering.

Auxiliary Rays and Weighted Ray Sampling As shown in Table. III, the usage of auxiliary rays can yield significant benefits, but there are clear drawbacks: they consume more GPU memory and computational resources during training and inevitably introduce misaligned rays. We conducted comprehensive ablation experiments, and the results are presented in Fig .5. Without using Weighted Ray Sampling,

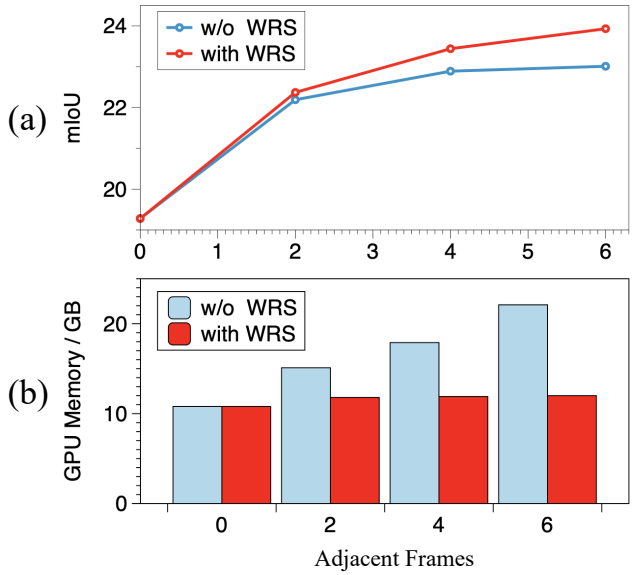

Fig. 5. **Ablation Study For Auxiliary-Ray.** (a) With the increased utilization of adjacent frames, there is a corresponding rise in mIoU. (b) Weighted Ray Sampling (WRS) effectively mitigates the additional training cost associated with auxiliary rays while improving performance.

as the number of auxiliary frames increases from 0 to 6, the mIoU gradually improves from 19.28 to 23.01. However, as shown in Fig. 5(b), GPU memory usage increases significantly with the doubling of ray numbers, incurring additional training costs. With the introduction of Weighted Ray Sampling, we keep the Ray numbers fixed at 38400 and focus on sampling valuable rays with high information density, achieving higher mIoU without incurring additional overhead.

Training Without LiDAR In Table V, we compare various depth supervision methods. Without depth supervision, RenderOcc scores 16.9 mIoU. Leveraging Raw LiDAR for depth supervision yields an mIoU of 23.44. However, expensive LiDAR data is often inaccessible in many scenarios. Therefore, we also attempted to compute depth labels directly from images using a struct-from-motion approach, following [50]. Although the generated depth labels are quite sparse, RenderOcc still achieves 21.11 mIoU. This demonstrates potential in training 3D Occupancy models without LiDAR data, meriting deeper investigation in future work.

V. CONCLUSION

We dive into the potential of using 2D image labels to train 3D occupancy networks, and propose a general framework RenderOcc to effectively implement this idea. This approach circumvents the production of costly and ambiguous 3D occupancy labels, training vision-centric models in a cheap and more intuitively direct manner. The extensive experiments validate the effectiveness of our method, offering a new perspective for the community.

VI. ACKNOWLEDGEMENT

This research was partly supported by the foundation of the National Key R&D Program of China (2022ZD0116305).

REFERENCES

[1] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3782–3795, 2019.

[2] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[3] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *European Conference on Computer Vision*. Springer, 2020, pp. 194–210.

[4] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," in *Conference on Robot Learning*. PMLR, 2022, pp. 180–191.

[5] "Tesla ai day," https://www.youtube.com/watch?v=j0z4FweCy4M, 2021.

[6] W. Tong, C. Sima, T. Wang, L. Chen, S. Wu, H. Deng, Y. Gu, L. Lu, P. Luo, D. Lin *et al.*, "Scene as occupancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8406–8415.

[7] X. Wang, Z. Zhu, W. Xu, Y. Zhang, Y. Wei, X. Chi, Y. Ye, D. Du, J. Lu, and X. Wang, "Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception," *arXiv preprint arXiv:2303.03991*, 2023.

[8] X. Tian, T. Jiang, L. Yun, Y. Mao, H. Yang, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[9] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, J. Zhou, and J. Lu, "Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 21 729–21 740.

[10] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[11] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2019. [Online]. Available: http://dx.doi.org/10.1109/iccv.2019.00939

[12] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2019. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2019.01298

[13] A. Garcia-Garcia, F. Gomez-Donoso, J. Garcia-Rodriguez, S. Orts-Escolano, M. Cazorla, and J. Azorin-Lopez, "Pointnet: A 3d convolutional neural network for real-time object class recognition," in *2016 International Joint Conference on Neural Networks (IJCNN)*, Jul 2016. [Online]. Available: http://dx.doi.org/10.1109/ijcnn.2016.7727386

[14] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, p. 3337, Oct 2018. [Online]. Available: http://dx.doi.org/10.3390/s18103337

[15] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking." in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr46437.2021.01161

[16] T. Wang, X. Zhu, J. Pang, and D. Lin, "Fcos3d: Fully convolutional one-stage monocular 3d object detection," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct 2021. [Online]. Available: http://dx.doi.org/10.1109/iccvw54120.2021.00107

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[18] J. Huang, G. Huang, Z. Zhu, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," *arXiv preprint arXiv:2112.11790*, 2021.

[19] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *European conference on computer vision*. Springer, 2022, pp. 1–18.

[20] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," *arXiv preprint arXiv:2206.10092*, 2022.

[21] Y. Li, H. Bao, Z. Ge, J. Yang, J. Sun, and Z. Li, "Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo," *arXiv preprint arXiv:2209.10248*, 2022.

[22] J. Li, M. Lu, J. Liu, Y. Guo, Y. Du, L. Du, and S. Zhang, "Bev-lgkd: A unified lidar-guided knowledge distillation framework for multi-view bev 3d object detection," *IEEE Transactions on Intelligent Vehicles*, 2023.

[23] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," *arXiv preprint arXiv:2203.05625*, 2022.

[24] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, and X. Zhang, "Petrv2: A unified framework for 3d perception from multi-camera images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3262–3272.

[25] X. Chi, J. Liu, M. Lu, R. Zhang, Z. Wang, Y. Guo, and S. Zhang, "Bev-san: Accurate bev 3d object detection via slice attention networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 461–17 470.

[26] J. Park, C. Xu, S. Yang, K. Keutzer, K. Kitani, M. Tomizuka, and W. Zhan, "Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection," *arXiv preprint arXiv:2210.02443*, 2022.

[27] X. Lin, T. Lin, Z. Pei, L. Huang, and Z. Su, "Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion," *arXiv preprint arXiv:2211.10581*, 2022.

[28] A.-Q. Cao and R. de Charette, "Monoscene: Monocular 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3991–4001.

[29] L. Roldao, R. de Charette, and A. Verroust-Blondet, "Lmscnet: Lightweight multiscale 3d semantic completion." in *2020 International Conference on 3D Vision (3DV)*, Nov 2020. [Online]. Available: http://dx.doi.org/10.1109/3dv50981.2020.00021

[30] J. Li, K. Han, P. Wang, Y. Liu, and X. Yuan, "Anisotropic convolutional networks for 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3351–3359.

[31] Y. Li, Z. Yu, C. Choy, C. Xiao, J. M. Alvarez, S. Fidler, C. Feng, and A. Anandkumar, "Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9087–9098.

[32] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Tri-perspective view for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9223–9232.

[33] R. Miao, W. Liu, M. Chen, Z. Gong, W. Xu, C. Hu, and S. Zhou, "Occdepth: A depth-aware method for 3d semantic scene completion," *arXiv preprint arXiv:2302.13540*, 2023.

[34] W. Gan, N. Mo, H. Xu, and N. Yokoya, "A simple attempt for 3d occupancy estimation in autonomous driving," *arXiv preprint arXiv:2303.10076*, 2023.

[35] F. Wimbauer, N. Yang, C. Rupprecht, and D. Cremers, "Behind the scenes: Density fields for single view reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9076–9086.

[36] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[37] K. Zhang, G. Riegler, N. Snavely, and V. Koltun, "Nerf++: Analyzing and improving neural radiance fields," *arXiv preprint arXiv:2010.07492*, 2020.

[38] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5855–5864.

[39] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-nerf 360: Unbounded anti-aliased neural radiance fields," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5470–5479.

[40] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[41] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[42] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (ToG)*, vol. 41, no. 4, pp. 1–15, 2022.

[43] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," in *European Conference on Computer Vision*. Springer, 2022, pp. 333–350.

[44] N. Max, "Optical models for direct volume rendering," *IEEE Transactions on Visualization and Computer Graphics*, p. 99–108, Jun 1995. [Online]. Available: http://dx.doi.org/10.1109/2945.468400

[45] D. Eigen, C. Puhrsch, and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network," *Advances in neural information processing systems*, vol. 27, 2014.

[46] L. Rudin and S. Osher, "Total variation based image restoration with free local constraints," in *Proceedings of 1st International Conference on Image Processing*, Dec 2002. [Online]. Available: http://dx.doi.org/10.1109/icip.1994.413269

[47] Y. Zhang, Z. Zhu, and D. Du, "Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction," *arXiv preprint arXiv:2304.05316*, 2023.

[48] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows." in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct 2021. [Online]. Available: http://dx.doi.org/10.1109/iccv48922.2021.00986

[49] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.445

[50] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, Y. Rao, G. Huang, J. Lu, and J. Zhou, "Surrounddepth: Entangling surrounding views for self-supervised multi-camera depth estimation," in *Conference on Robot Learning*. PMLR, 2023, pp. 539–549.