RenderWorld: World Model with Self-Supervised 3D Label

Ziyang Yan[1,2,3*], Wenzhen Dong[4*], Yihua Shao[5*], Yuhang Lu[1], Haiyang Liu[5], Jingwen Liu[5], Haozhe Wang[6], Zhe Wang[4], Yan Wang[4✉], Fabio Remondino[2], Yuexin Ma[1✉]

Abstract— End-to-end autonomous driving with vision-only is not only more cost-effective compared to LiDAR-vision fusion but also more reliable than traditional methods. To achieve a economical and robust purely visual autonomous driving system, we propose RenderWorld, a vision-only end-to-end autonomous driving framework, which generates 3D occupancy labels using a self-supervised gaussian-based Img2Occ Module, then encodes the labels by AM-VAE, and uses world model for forecasting and planning. RenderWorld employs Gaussian Splatting to represent 3D scenes and render 2D images greatly improves segmentation accuracy and reduces GPU memory consumption compared with NeRF-based methods. By applying AM-VAE to encode air and non-air separately, RenderWorld achieves more fine-grained scene element representation, leading to state-of-the-art performance in both 4D occupancy forecasting and motion planning from autoregressive world model.

I. INTRODUCTION

With the wide application of autonomous driving [1], [2], [3], researchers gradually focus on better perception and forecasting methods [4], which are related to the decision-making ability and robustness of the system [5], [6]. Most current frameworks consist of perception [7], forecasting, and planning separately [8]. The most commonly used perception method is 3D target detection using vision and LIDAR fusion [3], [9], [1], allowing the model to better forecast future scenes and do motion planning. Since most 3D target detection methods [10], [11], [12] are unable to obtain fine-grained information in the environment, they are non-robust in planning [13] in the subsequent model, which affects the system safety. Current perception methods primarily utilize both LiDAR [14], [15] and cameras [16], but the high cost of LiDAR and the computational demands of multimodal fusion pose challenges to the real-time performance and robustness of autonomous driving systems.

In this paper, we introduce **RenderWorld**, an autonomous driving framework for prediction and motion planning, which is trained on 3D occupancy labels generated by a Gaussian-based Img2Occ module. RenderWorld proposes an self-supervised Img2Occ module with Gaussian Splatting [17], trained on 2D multi-view depth and semantic images to generate 3D occupancy labels required for the world model. To enable the world model to better understand the scene represented by 3D occupancy, we propose the Air Mask Variational Autoencoder (AM-VAE) upon a vector-quantized variational autoencoder (VQ-VAE) [18]. This improves the inference capability of our world model by enhancing the granularity of the scene representation.

In order to verify the efficiency and reliability of RenderWorld, we evaluate the 3D occupancy generation and motion planing on NuScenes [19] separately. In summary, our contributions are mainly as follows:

1) We propose RenderWorld, a pure 2D autonomous driving framework that uses labeled 2D images to train a Gaussian-based occupancy prediction module (Img2Occ) for generating the 3D labels required by the world model.

2) To improve spatial representation abilities, we introduce AM-VAE, which improves forecasting and planning in world models while reducing memory consumption by separately encoding air and non-air voxels.

II. RELATED WORK

A. 3D Occupancy Prediction

3D occupancy is gaining attention as a viable alternative to LiDAR perception [20]. Most previous works [21], [22], [1], [23] utilize 3D Occupancy Ground Truth for supervision, which is challenging to annotate. With the widespread adoption of Neural Radiance Fields (NeRF) [24], some methods [25], [20], [26], [27], [28], [29] have attempted to use 2D depth and semantic labels for training. However, using continuous implicit neural fields to predict occupancy probabilities and semantic information often leads to high memory cost. Recently, GaussianFromer [30] leverages sparse Gaussian points as a means of reducing GPU consumption to describe 3D scenes while GaussianOcc [31] utilizes a 6D pose network to eliminate the reliance on ground truth poses, but both of them suffers from a significant drop in overall segmentation accuracy. In our work, we employ an anchor-based Gaussian initialization method to gaussianize voxel fratures and represent the 3D scenes with denser Gaussian points that achieving higher segmentation accuracy while avoiding the excessive memory consumption of ray sampling in NeRF-based methods.

B. World Model in Autonomous Driving

World models [32] are often used for future frame prediction and to assist robots in making decisions [33]. As end-to-end autonomous driving [8], [34] is gradually evolving,

[1]Ziyang Yan, Yuhang Lu, Yuexin Ma are with ShanghaiTech University, Shanghai, China. [2]Ziyang Yan, Fabio Remondino are with Fondazione Bruno Kessler, Trento, Italy. [3]Ziyang Yan is with University of Trento, Trento, Italy. [4]Wenzhen Dong, Zhe Wang, Yan Wang are with the Institute for AI Industry Research (AIR), Tsinghua University, Beijing, China. [5]Yihua Shao, Haiyang Liu, Jingwen Liu are with The University of Science and Technology Beijing, Beijing, China. [6]Haozhe Wang is The Hong Kong University of Science and Technology, Hong Kong, China.

* The first three authors contributed equally.
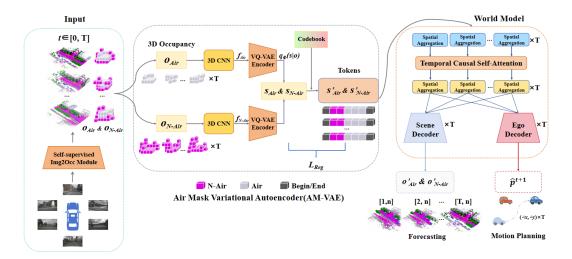
✉Corresponding to mayuexin@shanghaitech.edu.cn

Fig. 1. **General pipeline of RenderWorld.** We firstly generate the 3D occupancy labels through an Img2Occ Module (Figure 2). Then, using Air Mask Variational Autoencoder (AM-VAE) described in Section III-B, the separated air and non-air voxels are independently encoded into latent representations (i.e., discrete tokens). Finally, these latent representations are processed according to the specifications in Section III-C, and based on this, the voxels and trajectories are predicted, ultimately outputting the predicted occupancy and self-planning.

world models are also applied for predicting future scenarios and decisions making [35]. Unlike traditional autonomous driving approaches [36], [37], the world model approaches integrate perception, prediction and decision making. Many current approaches perform fusion of camera-LiDAR data and input into world model, which is used to forecast [38], [39] and make motion planning [40]. Among them, OccWorld [41] proposes to utilize 3D occupancy as world model's input. However, OccWorld is less effective at utilizing pure 2D input and struggles to accurately predict future scenes due to information loss during the encoding process. Hence, we design an Img2Occ Module to convert 2D labels into 3D occupancy labels to enhance the world modeling capabilities.

III. METHODOLOGY

In this section, We describe the overall implementation of RenderWorld. We firstly propose an Img2Occ Module for occupancy prediction and 3D occupancy labels generation (Sec III-A). Subsequently, we introduce a module based on the Air Mask Variational Autoencoder (AM-VAE) to optimize occupancy representation and enhance data compression efficiency (Section III-B). Finally, we elaborate on how to integrate the World Model for accurate prediction of 4D scene evolution (Section III-C).

A. 3D Occupancy prediction with Multi-frame 2D Labels

To enable 3D semantic occupancy prediction and future 3D occupancy labels generation, we design an Img2Occ Module which is illustrated in Figure 2. Using images from multi-cameras $\{Img_i\}_{i=1}^N$ as inputs, we firstly extract 2D image features using a pretrained BEVStereo4D [42] backbone and Swin Transformer [43]. Then, these 2D messages are interpolated into 3D space to produce volume features by leveraging the known intrinsic parameters $\{I_i\}_{i=1}^N$ and extrinsic parameters $\{E_i\}_{i=1}^N$. To project the 3D occupancy voxels onto multi-camera semantic maps, we apply Gaussian Splatting [17], an advanced real-time rendering pipeline.

Inspired by [44], we initialize anchor points with a learnable scale at the center of each voxel to approximate scene occupancy. The attributes of each anchor are determined based on the relative distance and viewing direction between the camera and the anchor. This anchor set is then used to initialize a Gaussian set with semantic labels $\{G_x\}_{x=1}^N$. Each Gaussian point x is then represented by a full 3D covariance matrix Σ in world space and its center position μ, and the color of each point is decided by the semantic label at that point.

$$G(x) = e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1} (x-\mu)} \tag{1}$$

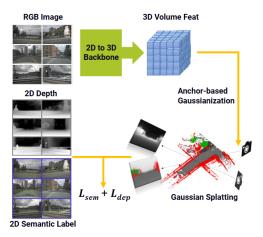Directly optimizing Σ may lead to infeasible matrices as it

Fig. 2. **Training paradigm of 2D-to-3D occupancy prediction Module.** Our proposed Img2Occ Module utilizes 2D labels to train the 3D occupancy network that allowing the model to take advantage of detailed 2D pixel-level semantics and depth supervision.

must be positive semi-definite. To ensure the validity of Σ, it is decomposed into the scaling matrix S and the rotation matrix R to characterize the geometry of a 3D Gaussian ellipsoid:

$$\Sigma = RSS^T R^T \tag{2}$$

Then the 3D Gaussians are projected to 2D for rendering by computing the camera space covariance matrix Σ' :

$$\Sigma' = JW\Sigma W^T J^T, \quad (3)$$

where J is the Jacobian matrix of the affine approximation of the projection transformation and W is the viewing transformation. The semantic / depth of each pixel can then be calculated by applying alpha blending onto sorted Gaussians:

$$D = \sum_i^N (d_i) a_i \prod_j^{i-1} (1 - a_j), \quad (4)$$

$$S = \sum_i^N (s_i) a_i \prod_j^{i-1} (1 - a_j), \quad (5)$$

where s_i/d_i is the rendered semantic / depth of a 3D Gaussian, a_i is the product of an evaluated 2D Gaussian projection and its corresponding opacity.

To calculate the difference between ground truth depth and rendered depth, we utilize the Pearson correlation which can measure the distribution difference between 2D depth maps follows the following function:

$$L_{dep}^i = Corr(\bar{d}_i, \ d_i) = \frac{Cov(\bar{d}_i, \ d_i)}{Var(\bar{d}_i, \ d_i)}, \quad (6)$$

where $\bar{d}_i$ is the ground truth depth image and d_i is the rendered depth image.

Finally, we construct the loss function with a cross-entropy loss L_{sem} for supervising semantic segmentation and L_{dep} for depth supervision, the overall loss can be computed as follows:

$$L_{i2o}^i = L_{sem}^i + L_{dep}^i \quad (7)$$

Using the well-trained checkpoint, we generate 3D occupancy labels, which are then input into the subsequent AM-VAE module.

B. Air Mask Variational Autoencoder (AM-VAE)

Traditional Variational Autoencoders (VAEs) fail to encode the distinct features of non-air voxels which hampers the model to represent scene elements as fine-grained level. To address this issue, we introduce the Air Mask Variational Autoencoder (AM-VAE), a novel VAE involves training two distinct Vector Quantized Variational Autoencoders (VQ-VAE) [18] to encode and decode air and non-air occupancy voxels separatly.

Assuming o represents the input occupancy representation, and o_{Air} and o_{N-Air} represent the air and non-air voxels. We first utilize a 3D convolutional neural network to encode the occupancy data, with the output being a continuous latent space representation denoted as f. The encoder $q_\phi(s|o)$ maps the input f to the latent space s. Then, we use two latent variables s_{Air} and s_{N-Air} to represent the air and non-air voxels, respectively:

$$s_{Air} \sim q_\phi(s_{Air}|o_{Air}), \quad s_{N-Air} \sim q_\phi(s_{N-Air}|o_{N-Air}) \quad (8)$$

Each encoded latent variable s_{Air} or s_{N-Air} uses learnable codebook C_{Air} or C_{N-Air} to obtain discrete token, which is then replaced by the most similar codebook entry before being fed into the decoder. This process is represented as:

$$s'_{Air} = \arg \min_{c_{Air} \in C_{Air}} \|s_{Air} - c_{Air}\|,$$
$$s'_{N-Air} = \arg \min_{c_{N-Air} \in C_{N-Air}} \|s_{N-Air} - c_{N-Air}\| \quad (9)$$

Then, the decoder $p_\theta(o|s)$ reconstructs the input occupancy from the quantized latent variables s'_{Air} and s'_{N-Air}:

$$\hat{o}_{Air} = p_\theta(o_{Air}|s'_{Air}), \quad \hat{o}_{N-Air} = p_\theta(o_{N-Air}|s'_{N-Air}) \quad (10)$$

To facilitate the separation of air and non-air elements within the occupancy representation, we denote M as the set of non-air categories. Then the indicator function for air and non-air in the modified occupancy can be defined as follows:

$$I_M(o) = \begin{cases} 1 & if \, o \in M, \\ 0 & otherwise. \end{cases} \quad (11)$$

The modified air occupancy o'_{Air} and non-air occupancy o'_{N-Air} are given by the following equations:

$$o'_{Air} = (1 - I_M(o)) \cdot o_{Air},$$
$$o'_{N-Air} = I_M(o) \cdot o_{N-Air} + (1 - I_M(o)) \cdot o_{Air} \quad (12)$$

To reconstruct the original occupancy representation, we use a $mask = (\hat{o}_{Air} \neq 0)$ to distinguish areas filled only with air. Then the reconstructed occupancy $\hat{o}$ combines the air and non-air components as follows:

$$\hat{o} = \hat{o}_{Air} \cdot mask + \hat{o}_{N-Air} \cdot (1 - mask) \quad (13)$$

We then build the loss function L_{VAE} for training the AM-VAE with reconstruction loss L_{Recon} and commitment loss L_{Reg}:

$$L_{Recon} = \mathbb{E}_{q_\phi(s_{Air}|o_{Air})}[\log p_\theta(o_{Air}|s'_{Air})] \\ + \mathbb{E}_{q_\phi(s_{N-Air}|o_{N-Air})}[\log p_\theta(o_{N-Air}|s'_{N-Air})], \quad (14)$$

$$L_{Com} = \|s_{Air} - s'_{Air}\|^2 + \|s_{N-Air} - s'_{N-Air}\|^2, \quad (15)$$

$$L_{VAE} = L_{Recon} + \beta L_{Com} \quad (16)$$

AM-VAE utilizes separate codebooks for air and non-air voxels within a unified encoder-decoder setup. This method effectively captures the unique features of each voxel type, thereby improving both reconstruction accuracy and generalization potential.

C. World Model

By applying a world model in autonomous driving to encode 3D scenes into high-level tokens, our framework can effectively capture environmental complexity, enabling

accurate autoregressive anticipation of future scenarios and vehicle decisions.

Inspired by OccWorld [45], we use a 3D occupancy to represent the scene and employ a self-supervised tokenizer to derive high-level scene tokens $\mathbf{T}$, and encode the spatial position of vehicles by aggregating the vehicle token $\mathbf{z}_0$. The world model is defined as w based on the current timestamp T and the number of historical frames t, then we establish the prediction with the following formula:

$$w(\mathbf{T}^T, \cdots, \mathbf{T}^{T-t}) = \mathbf{T}^{T+1}, \qquad (17)$$

where $\mathbf{T}^{T+1}$ represents the scene tokens at the next time step.

At the same time, a temporal generative transformer architecture is adopted to effectively predict the future scene. It firstly processes scene tokens through spatial aggregation and downsampling, and then generates a hierarchical set of tokens $\{\mathbf{T}_0, \cdots, \mathbf{T}_K\}$. So as to predict the future at different spatial scales, we take multiple sub-world models $w = \{w_0, \cdots, w_K\}$ to achieve it and each sub-model w_i applies temporal attention to the tokens at each position j using the following formula:

$$\hat{\mathbf{z}}_{j,i}^{T+1} = \mathrm{TA}(\mathbf{z}_{j,i}^T, \cdots, \mathbf{z}_{j,i}^{T-t}), \qquad (18)$$

where TA represents masked temporal attention, which predicts future tokens from influencing previous tokens. $\mathbf{z}_{j,i}^t \in \mathbf{T}_i^t$ denotes the j-th world token at scale i and timestamp t.

In the prediction module, we firstly utilize a self-supervised tokenizer e to convert the 3D scene into high-level scene tokens $\mathbf{T}$, and a vehicle token $\mathbf{z}_0$ to encode the spatial position of the vehicle. After predicting the future scene tokens, a scene decoder d is applied to decode the predicted 3D occupancy $\hat{\mathbf{y}}^{T+1} = d(\hat{\mathbf{z}}^{T+1})$, and learn a vehicle decoder d_{ego} which is for generating the vehicle displacement that relative to the current frame $\hat{p}^{T+1} = d_{ego}(\hat{z}_0^{T+1})$. The prediction module provides decision support for trajectory optimization of the autonomous driving system by generating continuous predictions of future vehicle displacements and scenario changes, ensuring safe and adaptive path planning.

We have implemented a two-stage training strategy to effectively train our prediction module. In the first phase, we train the scene tokenizer e and the decoder d using a 3D occupancy loss:

$$L_{e,d} = L_{soft}(d(e(\mathbf{y})), \mathbf{y}) + \lambda_1 \cdot L_{lovasz}(d(e(\mathbf{y})), \mathbf{y}), \quad (19)$$

where L_{soft} denotes the softmax loss and L_{lovasz} represents the Lovasz-softmax loss. The term λ_1 serves as a balancing factor between them.

Then we use the learned scene tokenizer e to obtain the scene tokens $\mathbf{z}$ for all frames and constrain the difference between the predicted tokens $\hat{\mathbf{z}}$ and $\mathbf{z}$. And a softmax loss is used to enforce the correct classification of $\hat{\mathbf{z}}$ to the correct code in the codebook $\mathbf{C}$. For the vehicle token, we simultaneously learn the vehicle decoder d_{ego} and apply an L2 loss on the predicted displacement $\hat{p} = d_{ego}(\hat{\mathbf{z}}_0)$ and the ground truth displacement $\mathbf{p}$. The overall loss in phase two

can be formulated as follows:

$$\begin{aligned} L_{w,d_{ego}} = \sum_{t=1}^{T}(\sum_{j=1}^{M_0} L_{soft}(\hat{\mathbf{z}}_{j,0}^t, \mathbf{C}(\mathbf{z}_{j,0}^t) \\ + \lambda_2 \, L_{L2}(d_{ego}(\hat{\mathbf{z}}_0^t), \mathbf{p}^t)), \end{aligned} \qquad (20)$$

where T and M_0 are the number of frames and the number of spatial tokens at the original scale, respectively. $\mathbf{C}(\cdot)$ denotes the index of the corresponding code in the codebook $\mathbf{C}$. L_{L2} measures the L2 difference between the two trajectories.

IV. EXPERIMENTS

In this section we evaluate the performance of Render-World using NuScenes [19] dataset. We also performed extensive ablation experiments on the same dataset - as reported in sub-section C - to deeper understand the proposed approach.

A. Experimental Setup

We adopt NuScenes as our evaluation dataset. NuScenes is a large-scale autonomous driving dataset that includes 700 scenes for training, 150 scenes for validation, and 150 scenes for testing, totaling approximately 40,000 frames across 17 classes. For self-supervised training, we generate ground truth depths and 2D segmentation ground truths by projecting LiDAR point clouds with their 3D segmentation labels onto corresponding 2D views. During the semantic occupancy prediction, each sample covers a range of [x:(-40 m, 40 m), y:(-40 m, 40 m), z:(-1.0 m, 5.4 m)] with a voxel size of 0.4 m. The evaluation experiments of our model are conducted on the 150 validation sets with one NVIDIA A30 GPU.

B. Main Result

3D semantic occupancy prediction: To demonstrate the performance of our model, we compare it against 10 occupancy prediction models, which are the existing common models evaluated on the NuScenes dataset. The results in Table I indicate that RenderWorld outperforms most state-of-the-art occupancy prediction methods in mIoU, ranking second overall, and only surpassed by CTF-OCC [46], which uses 3D occupancy GT as input. Furthermore, our method achieves outstanding performance in vehicle segmentation, including trailers, construction vehicles, trucks, etc and surpasses all other methods in segmenting various environmental terrains, such as vegetation, sidewalk etc. This is due to the 3D Gaussian representation, which effectively leverages the sparsity and object diversity in driving scenes, scaling with flexible location and covariance properties [30].

4D occupancy forecasting: We evaluated the 4D occupancy forecasting performance under several settings as shown in Table II

In order to capture finer-grained scene features and provide precise information for predictions, air-separation technique is applied to prioritize crucial non-air components in the scene, boosting prediction accuracy and computational efficiency. The results show that RenderWorld can generate non-trivial future 3D occupancy, with results far superior to

Methods	GT	mIoU ↑	Others	barrier	bicycle	bus	car	cons. veh	motorcycle	pedestrian	traffic cone	trailer	truck	dri. sur	other flat	sidewalk	terrain	manmade	vegetation
TPVFormer [21]	3D	27.83	7.22	38.90	13.67	**40.78**	**45.90**	**17.23**	19.99	18.85	14.30	**26.69**	**34.17**	55.65	35.47	37.55	30.70	19.40	16.78
BEVFormer [1]	3D	26.88	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
OccFormer [23]	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
CTF-Occ [46]	3D	**28.53**	**8.09**	**39.33**	**20.56**	38.29	42.24	16.93	**24.52**	**22.72**	**21.05**	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.0
RenderOcc [25]	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
SurroundOcc [47]	2D	20.30	-	20.59	11.68	28.06	30.86	10.70	15.14	14.09	12.06	14.38	22.26	37.29	23.70	24.49	22.77	14.89	21.86
GaussianFormer [30]	2D	19.10	-	19.52	11.26	26.11	29.78	10.47	13.83	12.58	8.67	12.74	21.57	39.63	23.28	24.46	22.99	9.59	19.12
GaussianOcc [31]	2D	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
OccNeRF [20]	2D	9.53	-	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
SelfOcc [27]	2D	9.30	0.00	0.15	0.66	5.46	12.54	0.00	0.80	2.10	0.00	0.00	8.25	55.49	0.00	26.30	26.54	14.22	5.60
RenderWorld (Ours)	2D	27.87	6.83	32.54	7.44	21.15	29.92	16.68	11.43	17.45	16.48	24.02	27.86	**75.05**	**36.82**	**50.12**	**53.04**	**22.75**	**24.23**

TABLE I

3D OCCUPANCY PREDICTION PERFORMANCE ON THE OCC3D-NUSCENES VALIDATION SET. OUR METHOD OUTPERFORMS STATE-OF-THE-ART METHODS, PARTICULARLY EXCELLING IN ENVIRONMENT-RELATED CATEGORIES (I.E. TERRAIN, VEGETATION.).

Method	Input	Aux. Sup.	mIoU ↑				IoU ↑				Memory
			1s	2s	3s	Avg.	1s	2s	3s	Avg.	
Copy&Paste	3D-Occ	None	14.91	10.54	8.52	11.33	24.47	19.77	17.31	20.52	-
OccWorld (Original) [41]	3D-Occ	None	25.78	15.14	10.51	17.14	34.63	25.07	20.18	26.63	13500M
RenderWorld(Ours)	3D-Occ	None	28.69	18.89	14.83	20.80	37.74	28.41	24.08	30.08	13000M
TPVFormer [21]+Lidar+OccWorld-T [41]	Camera	Semantic LiDAR	**4.68**	**3.36**	**2.63**	**3.56**	9.32	8.23	7.47	8.34	15000M
TPVFormer [21]+SelfOcc [27]+OccWorld-S [41]	Camera	None	0.28	0.26	0.24	0.26	5.05	5.01	4.95	5.00	15000M
RenderWorld(Ours)	Camera	None	2.83	2.55	2.37	2.58	**14.61**	**13.61**	**12.98**	**13.73**	**14400M**

TABLE II

4D OCCUPANCY FORECASTING PERFORMANCE. AUX. SUP. DENOTES AUXILIARY SUPERVISION APART FROM THE EGO TRAJECTORY. AVG. DENOTES THE AVERAGE PERFORMANCE OF THAT IN 1S, 2S, AND 3S.

Method	Input	Aux. Sup.	L2 (m) ↓				Collision Rate (%) ↓			
			1s	2s	3s	Avg.	1s	2s	3s	Avg.
IL [45]	LiDAR	None	0.44	1.15	2.47	1.35	0.08	0.27	1.95	0.77
NMP [48]	LiDAR	Box & Motion	0.53	1.25	2.67	1.48	**0.04**	<u>0.12</u>	<u>0.87</u>	0.34
FF [49]	LiDAR	Freespace	0.55	1.20	2.54	1.43	0.06	0.17	1.07	0.43
EO [50]	LiDAR	Freespace	0.67	1.36	2.78	1.60	**0.04**	**0.09**	0.88	<u>0.33</u>
ST-P3 [7]	Camera	Map & Box & Depth	1.33	2.11	2.90	2.11	0.23	0.62	1.27	0.71
UniAD [8]	Camera	Map & Box & Motion & Tracklets & Occ	0.48	<u>0.96</u>	**1.65**	**1.03**	0.05	0.17	**0.71**	**0.31**
VAD-Tiny [34]	Camera	Map & Box & Motion	0.60	1.23	2.06	1.30	0.31	0.53	1.33	0.72
VAD-Base [34]	Camera	Map & Box & Motion	0.54	1.15	1.98	1.22	**0.04**	0.39	1.17	0.53
OccNet [51]	Camera	3D-Occ & Map & Box	1.29	2.13	2.99	2.14	0.21	0.59	1.37	0.72
OccWorld-T [41]	Camera	Semantic LiDAR	0.54	1.36	2.66	1.52	0.12	0.40	1.59	0.70
OccWorld-S [41]	Camera	None	0.67	1.69	3.13	1.83	0.19	1.28	4.59	2.02
RenderWorld(Ours)	Camera	None	0.48	1.30	2.67	1.48	0.14	0.55	2.23	0.97
OccNet [51]	3D-Occ	Map & Box	1.29	2.31	2.98	2.25	0.20	0.56	1.30	0.69
OccWorld [41]	3D-Occ	None	<u>0.43</u>	1.08	1.99	<u>1.17</u>	0.07	0.38	1.35	0.60
RenderWorld(Ours)	3D-Occ	None	**0.35**	**0.91**	<u>1.84</u>	**1.03**	<u>0.05</u>	0.40	1.39	0.61

TABLE III

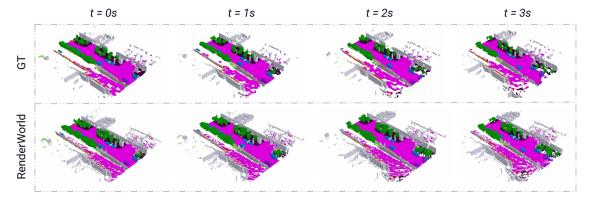**MOTION PLANNING PERFORMANCE.** AUX.SUP.DENOTES AUXILIARY SUPERVISION APART FROM THE EGO TRAJECTORY.

Fig. 3. **Visualization of the forecasting and planning results of RenderWorld.**

Setting	Forecasting mIoU (%) ↑				Planning L2 (m) ↓			
	1s	2s	3s	Avg.	1s	2s	3s	Avg.
$(50^2, 128, 512)$	**28.69**	**18.89**	**14.83**	**20.80**	**0.35**	0.91	1.84	1.03
$(50^2, 128, 256)$	27.16	18.09	14.45	19.90	**0.35**	**0.87**	**1.81**	**1.01**
$(50^2, 128, 1024)$	26.34	18.37	14.97	19.89	0.39	1.05	2.16	1.20
$(25^2, 256, 512)$	15.15	12.01	9.56	12.24	3.21	5.98	8.92	6.04
$(100^2, 128, 512)$	21.68	15.07	11.67	16.14	0.45	1.29	2.28	1.34

TABLE IV

EFFECT OF DIFFERENT HYPERPARAMETERS FOR THE SCENE TOKENIZER, THE SETTING DENOTES LATENT SPATIAL RESOLUTION, LATENT CHANNEL DIMENSION, AND THE CODEBOOK SIZE RESPECTIVELY.

OccWorld and Copy&Paste, which indicates that our model learns the underlying scene evolution.

Motion planning: As shown in Table III, We compare the motion planning performance between the proposed Render-World and state-of-the-art methods, and evaluate our model across various settings used in the 4D occupancy forecasting task. RenderWorld outperforms all compared methods in L2 metrics when takes 3D occupancy as input. Without any auxiliary support, our approach also achieves competitive results in collision rate and even outperforms OccWorld-S in when only uses 2D as input.

C. Ablation Study

With the aim of showing the effectiveness of our innovative modules, we conduct three ablation studies and the results are shown in Table V, IV and Table VI:

Efficiency comparisons among different representations: In Table V, we present the efficiency comparisons of various representations, highlighting that 3D Gaussian surpasses all competitors with significantly reduced memory usage. Leveraging its explicit representation, this approach assigns specific semantic data to individual 3D Gaussians, facilitating the transition from scene depiction to occupancy forecasts. This method also circumvents the high memory usage linked to the ray initialization step in NeRF-based techniques. Although our method has higher GPU memory overhead compared to GaussianFormer, it avoids the trade-off of reducing the number of Gaussian points to save memory, but leading to a loss of semantic information.

Methods	Query Form	Query Resolution	Memory
BEVFormer [1]	2D BEV	200×200	25100 M
TPVFormer [21]	2D Tri-Plane	$200 \times (200 + 16 + 16)$	29000 M
PanoOcc [52]	3D Voxel	$100 \times 100 \times 16$	35000 M
Fb-occ [53]	3D Voxel & 2D BEV	$200 \times 200 \times 16$ & 200×200	31000 M
OctreeOcc [54]	Octree Query	91200	26500 M
OccNeRF [20]	3D Voxel	$200 \times 200 \times 16$	79000 M
RenderOcc [25]	3D Voxel	$200 \times 200 \times 16$	23000 M
GaussianFormer [30]	3D Gaussian	144000	**6229 M**
RenderWorld (Ours)	3D Gaussian	640000	14400M

TABLE V

EFFICIENCY COMPARISON ON THE NUSCENES DATASET. THE RESULTS SHOW THAT 3D GAUSSIAN SIGNIFICANTLY REDUCES MEMORY USAGE COMPARED TO OTHER METHODS WITH OTHER REPRESENTATIONS.

Analysis of the scene tokenizer. Table IV demonstrates the impact of different hyperparameter settings on the performance of scene tokenizer, our parameters are designed like OccWorld. Larger spatial resolutions can enhance reconstruction accuracy but hinder prediction and planning, because limited token capacity for learning high-level concepts complicates future predictions [41]. Codebook sizes exceeding 512 lead to overfitting, while smaller sizes or resolutions compromise scene representation accuracy.

Effects of Mask module and VAE Separation Operation. Table VI presents the ablation study about our AM-VAE module (Separate VAE refers to dividing the potential space of the VAE into air and non-air portions). The introduction of the Air-Mask module leads to a performance improvement, achieving an mIoU of 37.68%. When applying both the Mask module and VAE separation operation together, the performance can noticeably reach to 40.25%. This indicates that our proposed Mask and VAE Separation operation offers substantial advantages in enhancing model reconstruction accuracy and reducing positional errors. Overall, the ablation study underscores the effectiveness of the proposed enhancements, especially the air separation strategy, in substantially boosting the performance of the RenderWorld framework.

Air-Mask	Separate VAE	mIoU
-	-	35.13
√	-	37.68
-	√	35.42
√	√	**40.25**

TABLE VI

ABLATION STUDIES OF AIR MASK AND VAE SEPARATION. EACH VALUE INDICATES THE PERFORMANCE ON THE VALIDATION DATASET.

V. CONCLUSIONS

In this paper, we introduce RenderWorld, a end-to-end autonomous driving framework trained on 3D occupancy labels generated by a Gaussian-based Img2Occ module and use world model for forecasting and motion planning. By leveraging Gaussian Splatting and AM-VAE, we successfully reduce GPU memory usage by at least half in 3D occupancy label generation compared to NeRF-based approaches, while simultaneously attaining minimal memory requirements in 4D occupancy forecasting. Experimental results demonstrate that our approach can achieve state-of-the-art performance in semantic segmentation of 3D occupancy, 4D occupancy forecasting with 2D input and motion planning among all input types. Our work offers a valuable contribution to the autonomous driving community, enhancing real-time, efficient robot perception, forecasting and motion planning.

REFERENCES

[1] 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.

[2] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu *et al.*, "Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 830–17 839.

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

[4] H. Yang, X. Bai, X. Zhu, and Y. Ma, "One training for multiple deployments: Polar-based adaptive bev perception for autonomous driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5602–5609.

[5] X. Tian, J. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," *arXiv preprint arXiv:2402.12289*, 2024.

[6] Y. Cui, S. Huang, J. Zhong, Z. Liu, Y. Wang, C. Sun, B. Li, X. Wang, and A. Khajepour, "Drivellm: Charting the path toward full autonomous driving with large language models," *IEEE Transactions on Intelligent Vehicles*, 2023.

[7] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision*. Springer, 2022, pp. 533–549.

[8] 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.

[9] 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," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 1477–1485.

[10] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 9939–9948.

[11] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun, "Perceive, predict, and plan: Safe motion planning through interpretable semantic representations," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*. Springer, 2020, pp. 414–430.

[12] B. Wei, M. Ren, W. Zeng, M. Liang, B. Yang, and R. Urtasun, "Perceive, attend, and drive: Learning spatial attention for safe self-driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4875–4881.

[13] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6120–6127.

[14] Y. Zhang, Z. Zhu, W. Zheng, J. Huang, G. Huang, J. Zhou, and J. Lu, "Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving," *arXiv preprint arXiv:2205.09743*, 2022.

[15] X. Peng, X. Zhu, and Y. Ma, "Cl3d: Unsupervised domain adaptation for cross-lidar 3d detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 2, 2023, pp. 2047–2055.

[16] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[17] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 1–14, 2023.

[18] A. Van Den Oord, O. Vinyals *et al.*, "Neural discrete representation learning," *Advances in neural information processing systems*, vol. 30, 2017.

[19] 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.

[20] C. Zhang, J. Yan, Y. Wei, J. Li, L. Liu, Y. Tang, Y. Duan, and J. Lu, "Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields," *arXiv preprint arXiv:2312.09243*, 2023.

[21] 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.

[22] 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.

[23] Y. Zhang, Z. Zhu, and D. Du, "Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9433–9443.

[24] F. Remondino, A. Karami, Z. Yan, G. Mazzacca, S. Rigon, and R. Qin, "A critical analysis of nerf-based 3d reconstruction," *Remote Sensing*, vol. 15, no. 14, p. 3585, 2023.

[25] M. Pan, J. Liu, R. Zhang, P. Huang, X. Li, H. Xie, B. Wang, L. Liu, and S. Zhang, "Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 12 404–12 411.

[26] X. Zhao, B. Chen, M. Sun, D. Yang, Y. Wang, X. Zhang, M. Li, D. Kou, X. Wei, and L. Zhang, "Hybridocc: Nerf enhanced transformer-based multi-camera 3d occupancy prediction," *IEEE Robotics and Automation Letters*, 2024.

[27] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu, "Selfocc: Self-supervised vision-based 3d occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 19 946–19 956.

[28] S. Boeder, F. Gigengack, and B. Risse, "Occflownet: Towards self-supervised occupancy estimation via differentiable rendering and occupancy flow," *arXiv preprint arXiv:2402.12792*, 2024.

[29] Y. Liu, L. Mou, X. Yu, C. Han, S. Mao, R. Xiong, and Y. Wang, "Let occ flow: Self-supervised 3d occupancy flow prediction," *arXiv preprint arXiv:2407.07587*, 2024.

[30] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu, "Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction," *arXiv preprint arXiv:2405.17429*, 2024.

[31] W. Gan, F. Liu, H. Xu, N. Mo, and N. Yokoya, "Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting," *arXiv preprint arXiv:2408.11447*, 2024.

[32] D. Ha and J. Schmidhuber, "World models," *arXiv preprint arXiv:1803.10122*, 2018.

[33] R. S. Sutton, "Integrated architectures for learning, planning, and reacting based on approximating dynamic programming," in *Machine learning proceedings 1990*. Elsevier, 1990, pp. 216–224.

[34] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang, "Vad: Vectorized scene representation for efficient autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8340–8350.

[35] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, and G. Corrado, "Gaia-1: A generative world model for autonomous driving," *arXiv preprint arXiv:2309.17080*, 2023.

[36] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[37] M. Bojarski, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[38] J. Gu, C. Hu, T. Zhang, X. Chen, Y. Wang, Y. Wang, and H. Zhao, "Vip3d: End-to-end visual trajectory prediction via 3d agent queries," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5496–5506.

[39] A. Hu, Z. Murez, N. Mohan, S. Dudas, J. Hawke, V. Badrinarayanan, R. Cipolla, and A. Kendall, "Fiery: Future instance prediction in bird's-eye view from surround monocular cameras," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 273–15 282.

[40] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3903–3913.

[41] W. Zheng, W. Chen, Y. Huang, B. Zhang, Y. Duan, and J. Lu, "Occworld: Learning a 3d occupancy world model for autonomous driving," *arXiv preprint arXiv:2311.16038*, 2023.

[42] J. Huang and G. Huang, "Bevdet4d: Exploit temporal cues in multi-camera 3d object detection," *arXiv preprint arXiv:2203.17054*, 2022.

[43] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[44] T. Lu, M. Yu, L. Xu, Y. Xiangli, L. Wang, D. Lin, and B. Dai, "Scaffold-gs: Structured 3d gaussians for view-adaptive rendering," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20654–20664.

[45] N. D. Ratliff, J. A. Bagnell, and M. A. Zinkevich, "Maximum margin planning," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 729–736.

[46] 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.

[47] 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. 21729–21740.

[48] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recogni-tion*, 2019, pp. 8660–8669.

[49] P. Hu, A. Huang, J. Dolan, D. Held, and D. Ramanan, "Safe local motion planning with self-supervised freespace forecasting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12732–12741.

[50] T. Khurana, P. Hu, A. Dave, J. Ziglar, D. Held, and D. Ramanan, "Differentiable raycasting for self-supervised occupancy forecasting," in *European Conference on Computer Vision*. Springer, 2022, pp. 353–369.

[51] 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.

[52] Y. Wang, Y. Chen, X. Liao, L. Fan, and Z. Zhang, "Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 17158–17168.

[53] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez, "Fb-occ: 3d occupancy prediction based on forward-backward view transformation," *arXiv preprint arXiv:2307.01492*, 2023.

[54] Y. Lu, X. Zhu, T. Wang, and Y. Ma, "Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries," *arXiv preprint arXiv:2312.03774*, 2023.