2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

GaussianPretrain: A Simple Unified 3D Gaussian Representation for Visual Pre-training in Autonomous Driving

Shaoqing Xu[1,2], Fang Li[1,2], Shengyin Jiang[2], Ziying Song[3], Zhi-Xin Yang[1†], *Member, IEEE*

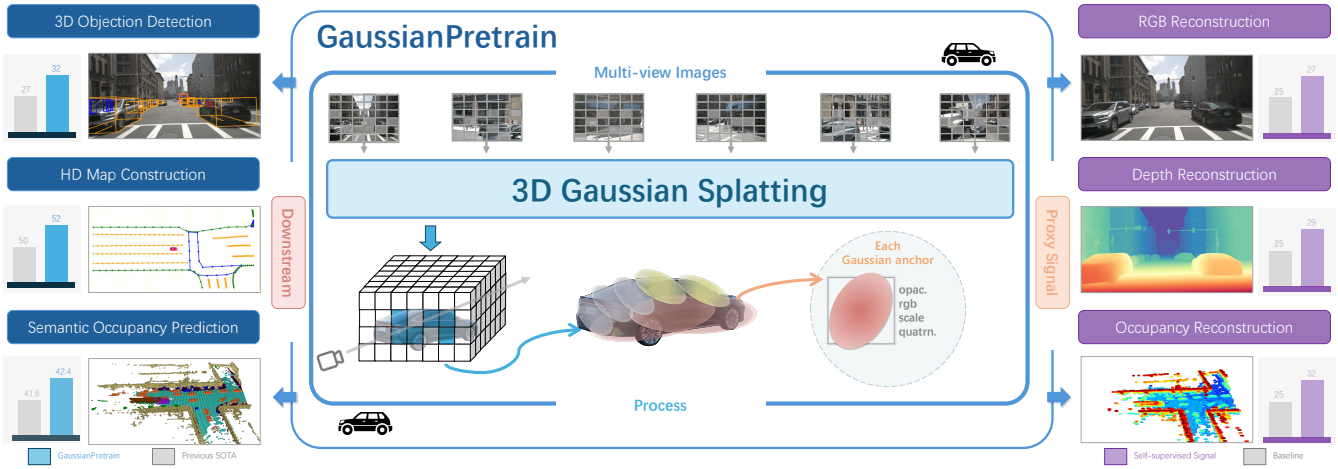

Fig. 1: Illustration of our proposed ***GaussianPretrain***. A simple, innovative, and efficient framework for vision pre-training with 3D Gaussian splatting representation. Benefits from our effective pre-training diagram, downstream application for 3D perception tasks achieved great improvement, including 3D object detection, HD-map construction, and Occupancy prediction.

***Abstract*— Self-supervised learning has made substantial strides in image processing, while visual pre-training for autonomous driving is still in its infancy. Existing methods often focus on learning geometric scene information while neglecting texture or treating both aspects separately, hindering comprehensive scene understanding. In this context, we are excited to introduce GaussianPretrain, a novel pre-training paradigm that achieves a holistic understanding of the scene by uniformly integrating geometric and texture representations. Conceptualizing 3D Gaussian anchors as volumetric LiDAR points, our method learns a deepened understanding of scenes to enhance pre-training performance with detailed spatial structure and texture, achieving that 40.6% faster than NeRF-based method UniPAD with 70% GPU memory only. We demonstrate the effectiveness of GaussianPretrain across multiple 3D perception tasks, showing significant performance improvements, such as a 7.05% increase in NDS for 3D object detection, boosts mAP by 1.9% in HD map construction and 0.8% improvement on Occupancy prediction. These significant gains highlight GaussianPretrain's theoretical innovation and strong practical potential, promoting visual pre-training development for autonomous driving. The source code is available at https://github.com/Public-BOTs/GaussianPretrain**

I. Introduction

With the development of autonomous driving technology, vision-centered solutions have gradually attracted widespread attention. Many studies focus on extracting bird's-eye view (BEV) features from multi-view input images to address various downstream applications. While supervised methods dominate current research, their reliance on accurate ground truth labels presents a significant bottleneck due to the high cost and difficulty of acquisition. Conversely, the abundance and accessibility of unlabeled data offer a promising avenue for improving performance. However, effectively harnessing this unlabeled data remains a significant and ongoing challenge in the field.

The core idea of self-supervised pre-training technology is to learn meaningful representations from abundant unlabeled data by leveraging carefully designed proxy tasks. Several approaches have been developed to explore this topic, methods like UniScene [1] and ViDAR [2] focus on predicting 3D occupancy or future LiDAR point clouds, effectively capturing geometric information but neglecting texture. Conversely, Self-Occ [3] and UniPAD [4] reconstruct 3D surfaces and RGB pixels from image-LiDAR pairs, thus capturing texture but relying solely on depth maps for geometric insights and focus on 2D appearance. Although this aids in texture learning but limits their ability to represent the detailed structure within occupied space. OccFeat [5], combines feature distillation with occupancy prediction to effectively capture both texture and geometry, but it introduces the complexity of an additional image foundation model and incurs significant pre-training costs.

Compared to Neural Radiance Fields (NeRF) methods, 3D

[1] The State Key Laboratory of Internet of Things for Smart City, and Centre for Artificial Intelligence and Robotics, and Department of Electromechanical Engineering, University of Macau, Macau SAR, China.
[2] Xiaomi EV, China. [3] Beijing Jiaotong University, China.
† Corresponding to Zhi-Xin Yang, zxyang@um.edu.mo.

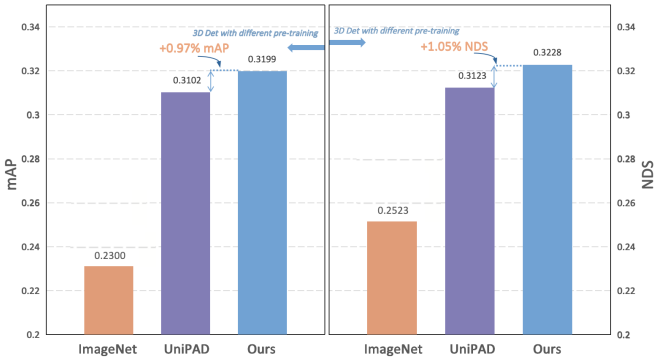

Fig. 2: Comparison of UVTR [7] model performance on the nuScenes dataset with different pre-training framework: ImageNet, UniPAD [4], and our GaussianPretrain.

Gaussian Splatting offers a powerful representation for scene reconstruction, encoding geometric and texture information through attributes like *position, color, rotation, scaling, and opacity*. Notably, 3D-GS has the advantage of both computational consumption and efficiency, particularly addressing key limitations in various tasks in robotics applications. While the NeRF-based method UniPAD[4] currently leads in 3D pertaining tasks, the potential of 3D-GS remains unexplored. In this work, we aim to investigate its performance and feasibility in 3D pre-training and verify it in the autonomous drive scene.

Inspired by the success of 3D-GS in effective scene representation and MAE [6] in 2D image self-supervised learning, we propose a novel pre-training approach **GaussianPretrain**, which combines 3D-GS with MAE method for pre-training tasks in 3D visual learning. Our approach incorporates two key innovations: (i) ***LiDAR Depth Guidance Mask Generator***. To enhance the efficiency of our approach, we only focus on learning the Gaussian information from a limited number of valid masked patches within the multi-view images or point clouds. These patches are identified by an MAE-like strategy and further filtered to include only those with LiDAR depth supervision. (ii). ***Ray-based 3D Gaussian anchor Guidance Strategy***: For each LiDAR-projected pixel, a ray-casting operation into 3D space to sample the points within the voxel. We introduce a set of learnable Gaussian anchors of these points to guide the learning of Gaussian properties from the 3D voxel as volumetric LiDAR points and predict the relevant attributes (e.g., *depth, opacity*). This enables the model to simultaneously understand the geometry and texture information of the scene through 3D Gaussian Splatting representation. Finally, we reconstruct the *RGB*, *depth*, and *occupancy* attributes solely within the valid masked patches by decoding the Gaussian parameters.

To demonstrate the effectiveness and generalizability of our GaussianPretrain, we conduct extensive experiments on various downstream vision tasks using the large-scale nuScenes[8] dataset. As shown in Figure 2, for 3D object detection, our pre-trained model significantly outperforms the ImageNet pre-trained of UVTR baseline, achieving the gain in **8.99%** mAP and **7.05%** NDS, which surpasses the previous state-of-the-art method, UniPAD [4] by 0.97% mAP and 1.05% NDS. Furthermore, we extend the experiments on HD-Map construction and occupancy prediction tasks. GaussianPretrain surpasses the HD map construction method, MapTR [9] by **1.9%** in mAP. For occupancy prediction, our approach achieves a 0.8% improvement in mIoU over PanoOCC[10], setting a new SOTA performance. Notably, compared to NeRF-based methods like UniPAD, our GaussianPretrain offers lower training time costs and memory consumption, while providing a better understanding of the scene's representation. By combining these advantages, our GaussianPretrain enables to perform an effective and efficient self-supervised pre-training paradigm. Further details are provided in Section IV.

To summarize, our main contributions are as follows:

- We introduce GaussianPretrain, a novel pre-training framework that integrates 3D Gaussian Splatting technology with a unified Gaussian representation. To the best of our knowledge, this is the first work to leverage 3D-GS for the pre-training paradigm, representing a novel contribution to this field.
- We propose a simple yet effective framework by leveraging 3D Gaussian anchors as volumetric LiDAR points, combined with Ray-based guidance and MAE strategy. Providing an efficient solution for visual pre-training, significantly reduces time consumption and GPU memory costs without sacrificing detail.
- The comprehensive experimental results demonstrate the superiority and generalizability of GaussianPretrain, showcasing significant improvements across various 3D perception tasks, including 3D object detection, HD-map construction, and Occupancy prediction. These advancements set new performance standards and underscore GaussianPretrain's transformative potential in visual pre-training tasks for autonomous driving applications.

II. Related Work

A. Pre-training for Autonomous Driving.

Pre-training on 2D images has been highly successful, mainly using contrastive learning and masked signal modeling to capture semantic and texture information. However, pre-training for visual autonomous driving demands accurate geometric representation, introducing additional challenges. Several works are currently exploring this area. For instance, UniScene [1] and OccNet [11] leverage occupancy prediction for pre-training, while ViDAR [2] predicts future LiDAR data from historical frame images. Although these methods are effective at capturing geometric information, but without detailed texture information. Conversely, methods like SelfOCC [3], UniPAD [4] and MIM4D [12] use NeRF to render RGB images and depth maps, learning texture but with limited geometric information. OccFeat [5] employs knowledge distillation to transfer texture information from an image foundation model during occupancy prediction but incurs high pre-training costs.

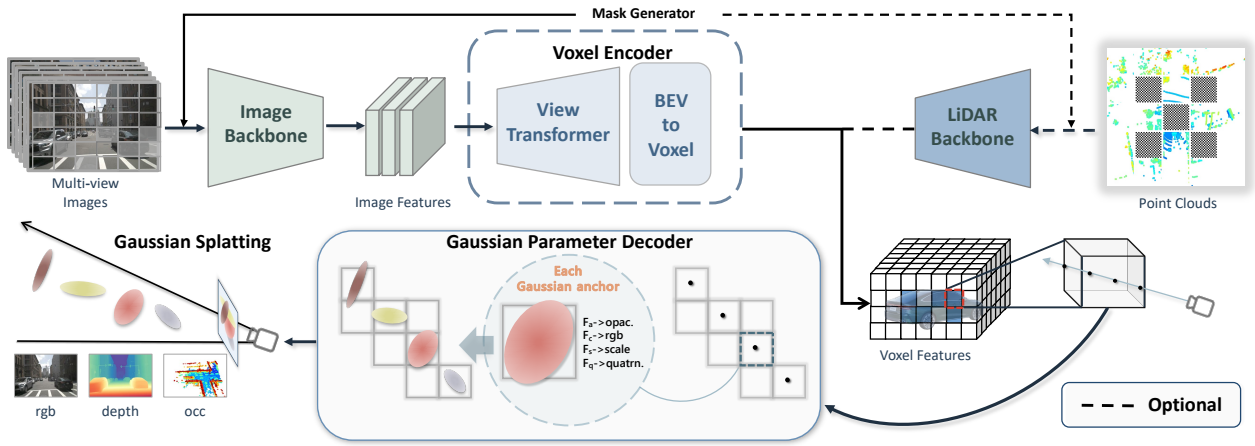

Fig. 3: The architecture of proposed GaussianPretrain. Given multi-view images or point clouds, we first extract valid mask patches using the mask generator with the LiDAR Depth Guidance strategy. Subsequently, a set of learnable 3D Gaussian anchors is generated using ray-based guidance and conceptualized as volumetric LiDAR points. Finally, the reconstruction signals of RGB, Depth, and Occupancy are decoded based on the predicted Gaussian anchor parameters.

B. 3D Gaussian Splatting and NeRF.

Neural Radiance Fields (NeRF) [13] achieve impressive rendering quality by implicitly representing scenes of color and density with volume rendering techniques. Follow-up works [14], [15], [16], [17] have successfully extended NeRF to various tasks. However, these methods still require extensive per-scene optimization, limiting their efficiency due to slow optimization and rendering speeds. In contrast, 3D Gaussian Splatting [18] explicitly represents scenes with anisotropic Gaussians, enabling real-time rendering via differentiable rasterization. For example, GPS-Gaussian [19] performs epipolar rectification and disparity estimation from image pairs, relying on stereo images and ground-truth depth maps. Similarly, Spatter Image [20] focuses on single-object 3D reconstruction from single views. Both methods are often constrained by inefficiencies, limited to object reconstruction, and dependent on specific input formats such as image pairs or single views. In this paper, we extend 3D Gaussian Splatting into visual pre-training tasks, overcoming limitations related to the number of views and the necessity of depth maps by presetting fixed-position 3D Gaussian anchors in 3D space, marking a novel application of 3D-GS.

III. Method

The pipeline of our GaussianPretrain, a simple, innovative, and efficient framework for vision pre-training using 3D-GS representation, is illustrated in Figure.3. Given multi-view images with valid masked patches, our goal is to reconstruct the signals, including *RGB*, *Depth*, and *occupancy* by decoding Gaussian parameters $\{(\mu_j, \alpha_j, \Sigma_j, c_j)\}_{j=1}^{K}$ for each scene, where μ_j, α_j, Σ_j and c_j are the 3D Gaussian's position, opacity, covariance, and color information, and K denote the max number of Gaussian Anchors.

In this section, we first detail the generation of valid masked patches under MAE method and Depth guidance in Section III-A and introduce the initialization of 3D Gaussian anchors through Ray-based Guidance in Section III-B. Subsequently, we review the process of converting multi-view images into a 3D voxel space in Section III-C and present the process of the Gaussian decoder part in Section III-D. The reconstruction task and the loss functions of GaussianPretrain will be presented in Section III-E.

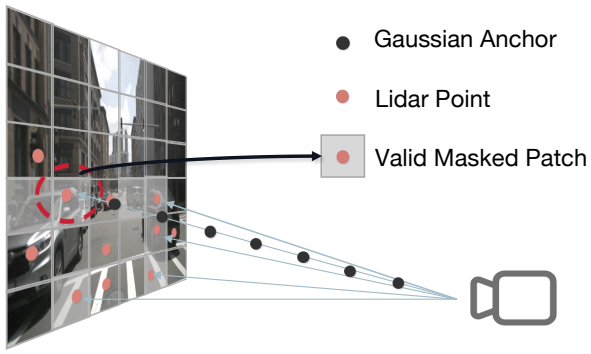

Fig. 4: Process of generating valid mask patches.

A. LiDAR Depth Guidance Mask Generator

Inspired by MAE [6], we apply random patch masking to multi-view images or point clouds, denoted as M. For image, sparse convolution is used to replace traditional convolutions in the image backbone, as implemented by SparK [21]. The valid areas of point clouds are obtained by filtering the non-masked region. For computational efficiency and generalization, we only focus on learning Gaussian parameters from a limited set of valid masked patches.

Additionally, for input images, we double-check the mask region by verifying the presence of LiDAR points within a certain depth range. As illustrated in Figure 4, if a set of points project into the masked patch M_i in the images and their depth falls within the range of [a, b], the mask

region will be marked as valid, M_i'. The process is outlined as follows:

$$M'^{n}_{i=1} = valid, \text{if } Proj(Set(pc)) \in \{[a,b], M\} \quad (1)$$

where n represents the number of valid masked patches, with n $\leq$ m. This strategy ensures that our model concentrates on the foreground, avoiding unnecessary attention to irrelevant background elements like the sky.

B. 3D Gaussian Anchor Initialization with Ray-based Guidance

To enable the model to simultaneously understand the geometry and texture information of the scene, we introduce a series of learnable Gaussian anchors in 3D space. These anchors guide the learning of Gaussian properties derived from the 3D voxel grid, treated as volumetric LiDAR points. Consider the LiDAR-projected pixel denoted by $\mathbf{u} = (u_1, u_2, 1)$ which corresponds to a ray R that extends from the camera into 3D space. Along this ray, we sample D ray points $\{p_j = \mathbf{u}d_j | j = 1, ...D,\ d_j < d_{j+1}\}$, where d_j is the corresponding depth along the ray. Each sampled ray point p in the valid mask region M' can be immediately unprojected to 3D space using the projection matrix summary as 3D Gaussian Anchors, $\mathscr{G}_p^{M'}(\cdot)$. This strategy not only eliminates the need for full image rendering, significantly reducing memory usage, but also enables simultaneous the RGB, depth, and occupancy reconstruction, a capability that has yet to be achieved by prior methods.

C. Voxel Encoder

For images, view transformer is typically used to generate Bird's Eye View (BEV) features. In our baseline model, UVTR [7], we employ the lift-splat-shoot (LSS) [22] and VoxelNet [23] to generate image or points 3D voxel features, respectively. We extend the channel dimension to incorporate a height dimension for image, producing $V \in R^{C \times Z \times H \times W}$, where C, H, W, and Z represent the channel number, dimensions along the x, y, and z axes, respectively. Additionally, for each LiDAR-projected pixel, we perform a ray-casting operation to extract N_t sampled target voxel where exists Gaussian Anchors V_t from 3D voxel grid V.

D. Gaussian Parameter Decoder

As shown in Figure 3, by conceptualizing $\mathscr{G}_p^{M'}$ as 3D Gaussian anchors, this unified representation enables the efficient capture of high-quality, fine-grained details, providing a more comprehensive understanding of the scene.

Specially, each 3D Gaussian anchor is characterized by attributes $\mathscr{G} = \{x \in \mathbb{R}^3, c \in \mathbb{R}^3, r \in \mathbb{R}^4, s \in \mathbb{R}^3, \alpha \in \mathbb{R}^1\}$ and the proposed Gaussian maps G are defined as:

$$G(x) = \{\mathscr{M}_c(x), \mathscr{M}_r(x), \mathscr{M}_s(x), \mathscr{M}_\alpha(x)\} \quad (2)$$

where x is the position of a Gaussian anchor in 3D space, $\mathscr{M}_c$, $\mathscr{M}_r$, $\mathscr{M}_s$, $\mathscr{M}_\alpha$ represents Gaussian parameters maps of color, rotation, scaling and opacity, respectively.

Due to the overlapping areas in the multi-view images, the pixel-by-pixel prediction of Gaussian parameters may lead to ambiguity due to overlapping splats. In contrast, we argue that predicting Gaussian parameters in a feed-forward manner directly from 3D voxel features is a better choice. Given voxel features V and center coordinate x , we employ trilinear interpolation to sample the corresponding feature $f(x)$ as follows:

$$f(x) = TriInter(V, x) \quad (3)$$

The Gaussian parameter maps are generated by prediction heads, defined as $h = MLP(\cdot)$, which consist of multiple MLP layers. Each prediction head is specifically designed to regress a particular parameter based on the sampled feature $f(x)$. For the parameter of color and opacity, we employed the sigmoid function for a range of [0,1] as follows:

$$\mathscr{M}_c(x) = Sigmoid(h_c(f(x)) \quad (4)$$

$$\mathscr{M}_\alpha(x) = Sigmoid(h_\alpha(f(x)) \quad (5)$$

where h_c, h_α denote the head of color and opacity.

Before being used to formulate Gaussian representations, the rotation map should be normalized since it represents a quaternion to ensure unit magnitude while the scaling map needs activations to satisfy their range as follows:

$$\mathscr{M}_r(x) = Norm(h_r(f(x)) \quad (6)$$

$$\mathscr{M}_s(x) = Softplus(h_s(f(x)) \quad (7)$$

where h_r, h_s represents the rotation head and scale head.

E. Supervise by Reconstruction Signals

To facilitate a better reconstruction of the masked region under the MAE strategy, we supervise the learning process with different reconstruction signals derived from the Gaussian representation. Specifically, the *RGB*, *Depth*, and *Occupancy* signals are decoded based on the predicted Gaussian anchor parameters within the valid mask patches.

a) ***RGB Reconstruction****:* Since we do not need to reconstruct images of arbitrary perspectives, we directly predict fixed viewpoint *RGB* instead of using *Spherical Harmonics (SH)* coefficients. After predicting the parameters of the Gaussian anchor, we decode the color information with the $\hat{C}(p) = \sum_{i=1}^{N} c_i \alpha_i \tau$ to render the RGB values map of the image, for each target reconstruct pixel. Specifically, the c_i value in the equation is replaced by the predicted RGB, α_i is the opacity-related influence of this Gaussian to the current pixel and $\tau = \prod_{j=1}^{i-1}(1-\alpha_j)$ is the transmittance.

b) ***Depth Reconstruction****:* Inspired by the depth implementation in NeRF-style volume rendering, we integrate the depth of each splat in a manner similar to RGB reconstruction and approximate each pixel z-depth from the 3D-GS parameters. The process is as: $\hat{D} = \sum_{i=1}^{n} d_i \alpha_i \prod_{j=1}^{i-1}(1-\alpha_j)$ where n is the number of Gaussian anchors, d_i is the i^{th} Gaussian anchor z-depth coordinate in view space, enabling efficient depth rendering with minimal computational overhead. $\hat{D}$ is the depth map of the image. Notably, the ground truth of depth is from sparse depth projected by LiDAR.

Methods	Backbone	CBGS	NDS↑	mAP↑	**mATE↓**	**mASE↓**	**mAOE↓**	**mAVE↓**	**mAAE↓**
LiDAR-base									
CenterPoint [24]	V0.075	✓	65.3	56.9	0.285	0.253	0.323	0.272	0.186
UVTR-L [7]	V0.075	✓	67.7	60.9	0.334	0.257	0.300	0.204	0.182
TransFusion-L [25]	V0.075	✓	70.1	65.1	-	-	-	-	-
UVTR-L+UniPAD† [4]	V0.075	✓	70.2	64.6	0.283	0.252	0.263	0.230	0.185
UVTR-L+GP	V0.075	✓	**70.5**	**65.1**	0.289	0.252	0.244	0.229	0.189
Camera-base									
UVTR-C [7]	R101		44.1	37.2	0.735	0.269	0.397	0.761	0.193
BEVFormerV2 [26]	V299		46.7	39.6	0.709	0.274	0.368	0.768	0.196
CMT-C [27]	V299		46.0	40.6	-	-	-	-	-
UVTR-C+UniPAD† [4]	ConvNeXt-S		46.4	41.0	0.671	0.277	0.382	0.867	0.211
UVTR-C+GP	ConvNeXt-S		**47.2**	**41.7**	0.676	0.278	0.394	0.815	0.200
StreamPETR† [28]	R50		47.9	38.0	0.686	0.280	0.622	0.303	0.217
StreamPETR+GP	R50		**48.8**	**38.6**	0.671	0.273	0.593	0.307	0.206
Camera + LiDAR									
FusionPainting[29]	V0.075-R50	✓	70.7	66.5	-	-	-	-	-
BEVFusion[30]	V0.075-swint	✓	71.4	68.5	-	-	-	-	-
UVTR-M	V0.075-R101	✓	70.2	65.4	0.332	0.258	0.268	0.212	0.177
UVTR-M+UniPAD† [4]	V0.075-R101	✓	72.8	69.4	0.276	0.253	0.236	0.233	0.189
UVTR-M + GP	V0.075-R101	✓	**73.7**	**70.7**	0.267	0.258	0.218	0.234	0.196

TABLE I: *3D Object Detection.* We compare with SOTA methods *without* test-time augmentation on the nuScenes *val* set. †: denotes our reproduced results based on MMDetection3D [31]. C, L, M denotes the experiment under the setting of Camera, LiDAR and both sides.

*c) **Occupancy Reconstruction**:* The opacity attribute of 3D-GS point is inherent to vision perception, particularly for occupancy prediction tasks. Unlike GaussianFormer [32], which uses opacity for semantic logits, we directly interpret opacity as an indicator of occupancy. A Gaussian anchor with full opacity signifies the presence of an occupied location at x. Formally, for each target voxel, we take the maximum opacity value among Gaussian anchors within the voxel to represent the occupancy probability, denoted by $\hat{O}$. This direct mapping of opacity to occupancy provides a natural and effective way to leverage 3D Gaussian Splatting for occupancy prediction.

$$\hat{O} = \max_{j=1}^{k} (\mathscr{M}_{\alpha}^{j}(x)) \mid x \in V_t \tag{8}$$

where k is number of Gaussian anchors in a target voxel V_t.

*d) **Loss Function**:* In summary, the overall pre-training loss function consists of color, depth and occupancy loss: $L = \frac{\lambda_{RGB}}{N_t^p} \sum_{i=1}^{N_t^p} |C_i - \hat{C}_i| + \frac{\lambda_{Depth}}{N_t^p} \sum_{i=1}^{N_t^p} |D_i - \hat{D}_i| + \frac{\lambda_{Occupacy}}{N_t^v} \sum_{i=1}^{N_t^v} |O_i - \hat{O}_i| \cdot$ where C_i, D_i are the GT of RGB, depth for each ray. O_i denotes GT of occupancy which is ***considered occupied if it contains at least one lidar point***. N_t^p and N_t^v are the counts of target pixels of P_t and target voxels of V_t, respectively. Note that, λ_{RGB}**=0** if only the LiDAR branch exsits.

IV. Experiments

A. Datasets and Evaluation Metrics

nuScenes dataset [8] contains 700/150/150 scenes for training, validation, and testing, respectively. Each sequence is captured at 20Hz frequency with 20 seconds duration. Each sample contains RGB images from 6 cameras with 360° horizontal FOV and point cloud data from 32 beam LiDAR sensor, and five radars. For HD-map task, perception ranges are $[-15.0m, 15.0m]$ for the X-axis and $[-30.0m, 30.0m]$ for the Y-axis. We calculate the AP_τ under Chamfer distance with several thresholds ($\tau \in T, T = \{0.5, 1.0, 1.5\}$).

Occ3D-nuScenes [39] occupancy scope is defined as $-40m$ to $40m$ for X and Y-axis, and $-1m$ to $5.4m$ for the Z-axis. The voxel size is $0.4m \times 0.4m \times 0.4m$ for the occupancy label. Occ3D-nuScenes benchmark calculates the mean Intersection over Union (**mIoU**) for 17 semantic categories (including 'others'). Besides, it also provides visibility masks for LiDAR and camera modality.

B. Implementation Details

Our code implementation is based on MMDetection3D [31], and all models were trained on **8** NVIDIA A100 GPUs. Unless otherwise specified, the input image resolution is set to 1600x900 by default. The scale factors for λ_{RGB} and λ_{Occ} are maintained at 10, while λ_{Depth} is set to 1. During the pre-training phase, we applied the mask to the input images with a size of 32 and a ratio of 0.3. LiDAR specific depth range for generating the valid mask is [0, 50]. The model is pre-trained for 12 epochs using the AdamW optimizer, with an initial learning rate of 2e-4 and a weight decay of 0.01. In the ablation studies, unless explicitly stated, fine-tuning is conducted for 12 epochs on 50% of the image data.

Method	Modality	Backbone	Pretrain	Epochs	mAP	AP_{ped}	$AP_{divider}$	$AP_{boundary}$
HDMapNet	C	Effi-B0	ImageNet	30	23.0	14.4	21.7	33.0
HDMapNet	L	PointPillars[33]	ImageNet	30	24.1	10.4	24.1	37.9
HDMapNet	C & L	Effi-B0 & PointPillars	ImageNet	30	31.0	16.3	29.6	46.7
VectorMapNet	C	R50	ImageNet	110	40.9	36.1	47.3	39.3
VectorMapNet	L	PointPillars	ImageNet	110	34.0	25.7	37.6	38.6
VectorMapNet	C & L	R50 & PointPillars	ImageNet	110	45.2	37.6	50.5	47.5
MapTR-tiny†	C	R50	ImageNet	24	49.9	52.0	45.3	52.4
MapTR-tiny†+GP	C	R50	Ours	24	**51.8**	**54.9**	**45.8**	**54.9**

TABLE II: *HD-Map construction*. Comparisons with state-of-the-art methods on nuScenes *val* set. "C" and "L" respectively denotes camera and LiDAR. The APs of HDMapNet [34] and VectorMapNet [35] are taken from the paper of MapTR [9]. *GP* presents GaussianPretrain framework.

Method	Input Modality	Image Backbone	mIoU	car	bus	bicycle	barrier	vegetation	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	others
MonoScene	Camera	R101-DCN	6.06	9.38	4.93	4.26	7.23	7.65	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	1.75
BEVDet	Camera	R101-DCN	11.73	12.97	4.18	0.0	15.29	15.26	1.35	0.0	0.43	0.13	6.59	6.66	52.72	19.04	26.45	21.78	14.51	2.09
TPVFormer	Camera	R101-DCN	27.83	45.90	40.78	13.67	38.90	16.78	17.23	19.99	18.85	14.30	26.69	34.17	55.65	35.47	37.55	30.70	19.40	7.22
BEVFormer	Camera	R101-DCN	23.67	41.09	34.41	9.98	38.79	14.46	13.24	16.51	18.50	17.83	18.66	27.70	48.95	27.73	29.08	25.38	15.41	5.03
BEVFormer+**GP**	Camera	R101-DCN	24.21	39.18	36.55	6.95	34.88	18.16	11.88	16.62	16.93	17.1	13.83	27.03	54.09	32.36	33.02	27.05	20.39	5.53
PanoOCC	Camera	R101-DCN	41.60	54.78	45.46	28.92	49.82	40.10	25.20	32.93	28.86	30.71	33.87	41.32	83.18	45.00	53.80	56.10	45.11	11.99
PanoOCC+**GP**	Camera	R101-DCN	**42.42**	55.21	49.88	28.81	49.30	42.54	22.27	31.30	29.42	30.37	34.29	42.05	84.06	47.76	55.90	58.13	48.20	11.58

TABLE III: *3D Occupancy prediction* performance on the Occ3D-nuScenes dataset. The results of MonoScene [36], BEVDet [37] and TPVFormer[38], is reported in Occ3d [39].

C. *Main Results on Different Vision Task*

a) ***3D Object Detection****:* We compare our GaussianPretrain with previous SOTA methods of different modalities, as shown in Table I. We adopt UVTR as our baselines. Benefits from the effective 3D-GS representation pre-training, GaussianPretrain consistently improves the baselines, UVTR-C, UVTR-L, and UVTR-M by **4.7%**, **2.8%**, and **3.1%** NDS, respectively. Even compared to the previous NeRF-based pre-trained method, our approach achieves additional gains of 0.8, 0.3, and 0.5 NDS over UniPAD-C, UniPAD-L, and UniPAD-M, respectively, reaching the level of existing state-of-the-art methods without any test time augmentation.

b) ***HD Map Construction****:* As shown in Table II, we evaluate the performance of our pre-training model on the nuScenes dataset for the HD map construction task. This task requires the model to understand road topology and traffic rules, necessitating a detailed understanding of the scene's texture information. We utilize MapTR [9] to assess the ability of GaussianPretrain to capture this information. Benefiting from our effective pre-training of Gaussian representation, MapTR achieves a **1.9%** improvement in mAP.

c) ***3D Occupancy Prediction****:* The opacity attribute of Gaussian anchor is inherently suited for occupancy prediction tasks. In Table III, we conduct the experiments of 3D occupancy prediction on the Occ3D-nuScenes. The performance of the SOTA methods in the table is reported in the work of Occ3d [39]. We implement our framework on BEVFormer [40] and PanoOCC [10], achieving an improvement of 0.6% mIoU over BEVFormer and a further improvement of 0.8% mIoU over the SOTA method, PanoOCC. This also highlights the effectiveness of our pre-training diagram.

D. *Comprehensive Analysis on Ablation Studies*

RGB-Loss	Depth-Loss	OCC-Loss	NDS↑	mAP↑	mIoU
✗	✗	✗	25.23	23.00	15.1
✓			26.84	25.73	16.3
✓	✓		29.20	26.54	17.2
✓	✓	✓	**32.28**	**31.99**	**19.3**

TABLE IV: Ablation study on different supervised losses.

a) Effect of GaussianPretrain's losses: To validate the effectiveness of each reconstructed signal, we conducted experiments on *UVTR* and *BEVFomer* for 3D detection and occupancy tasks, respectively. The RGB loss guides the model in learning texture information of the scene, while the depth loss encourages learning the geometric on a 2D plane which is insufficient for capturing complete 3D geometry. In contrast, the occupancy loss supervises the model in learning comprehensive geometric details within 3D space. As shown in Table IV, each component contributes positively, with the best results achieved when all are used together. Notably, thanks to the attribute of the opacity of Gaussian representation which inherently benefits occupancy prediction, yielding a significant improvement of **2.1%** in mIoU. These results underscore the effectiveness of Gaussian representation to the pre-training framework.

Method	Decoder	Res.	Param.	Memory	Latency
UniPAD-C	NeRF	MAE	0.46MB	1125MB	32ms
GaussianPretrain	3D-GS		0.45MB	**788MB**	**19ms**
UniPAD-C	NeRF	1/4 Full	0.46MB	30.5G	390ms
GaussianPretrain	3D-GS		0.45MB	**9.6G**	**76ms**
UniPAD-C	NeRF	1/2 Full	0.46MB	**OOM**	-
GaussianPretrain	3D-GS		0.45MB	**51.9G**	**230ms**

TABLE V: Comparison of the consumption with the NeRF-based method. Res. denote "Resolution"

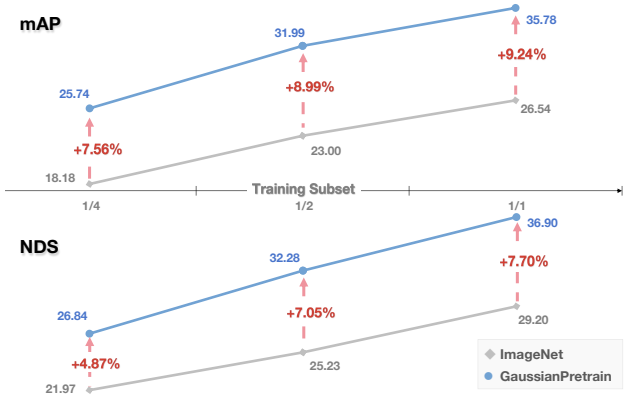

Fig. 5: Effect of GaussianPretrain on Fine-tuning. By reducing annotations from the full training set to a 1/4 subset.

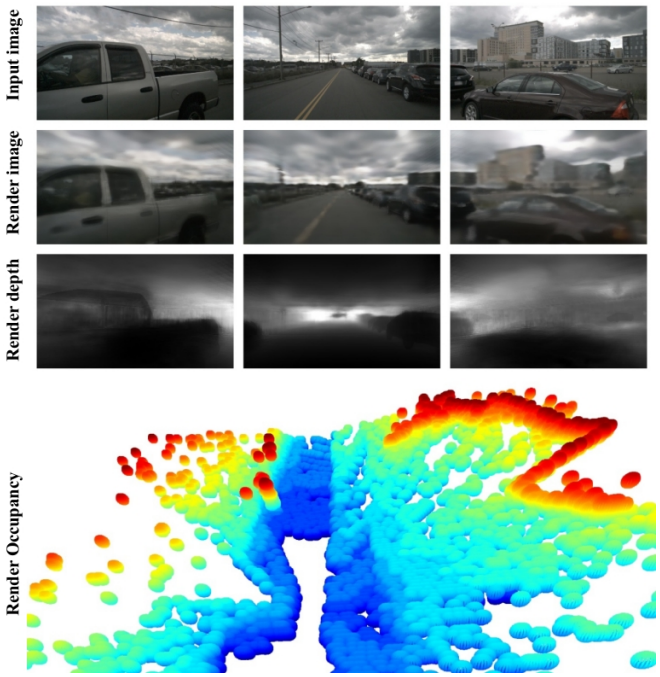

Fig. 6: Rendering results, pretrained by GaussianPretrain.

b) Efficiency & consumption: NeRF-based methods often suffer from slow convergence and high GPU memory consumption. In contrast, our 3D-GS based approach offers comparable rendering quality with significantly faster convergence and superior efficiency for free-view rendering. In the Table V, we compare the efficiency and memory consumption of the decoder module between the NeRF-base UniPAD and ours. Notably, GaussianPretrain obviously reduces memory usage by about **30%**, and decreases latency by approximately **40.6%** under MAE strategy while maintaining a similar parameter size. At 1/4 full image resolution, our method achieves a memory usage of 9.6GB with a latency of 76ms, which is **1/3** and **1/5** of UniPAD, respectively. Notably, for 1/2 full resolution, GaussianPretrain maintains functionality with 51.9GB memory usage and 230ms latency, whereas UniPAD-C fails due to an **out-of-memory error**. This highlights significant gains in resource efficiency with the GaussianPretrain approach.

*c) **Effect of Supervised Pre-training**:* We demonstrate the effectiveness of GaussianPretrain in reducing the reliance on annotations by fine-tuning UVTR, ranging from the full dataset to a ***1/4*** subset. As depicted in Figure 5, our approach surpasses the baseline under full supervision by **5.5**% mAP, ***with only half of the supervised samples***, i.e., 32.0% mAP vs. 26.5% mAP. This result indicates that GaussianPretrain can effectively leverage unlabeled data to compensate for reduced supervision, leading to improved performance even with fewer annotations.

*d) **Visualization.**:* As shown in Figure 6, we rendered the attributes of RGB, Depth, and Occupancy images from our pre-trained model. GaussianPretrain exhibits a substantial capacity for 3D scene reconstruction by means of Gaussian splitting. More visualization results on various downstream tasks are shown on the project page.

V. Conclusion and Limitations.

In this work, we introduce 3D Gaussian Splatting technology into the vision pre-training task for the first time. Our GaussianPretrain demonstrates remarkable effectiveness and robustness, achieving significant improvements across various downstream 3D vision perception tasks with efficiency and lower memory consumption.

Limitation. There still exist certain limitations in the current framework. Notably, it does not explicitly incorporate temporal information for better understanding dynamic objects or scenes, which is crucial for many autonomous driving applications. In future work, we plan to extend our GaussianPretrain to leverage such information and further enhance its performance and capabilities.

VI. Acknowledgement

This work was funded in part by the National Natural Science Foundation of China under Grant 62461160260, National Key Research and Development Program of China under grant 2023YFE0205800, the Science and Technology Development Fund, Macau SAR (Grant no. 0075/2023/AMJ, 0092/2024/AFJ, 0003/2023/RIB1, 001/2024/SKL), the Guangdong Department of Science and Technology (Grant no. 2023A0505030003), and the University of Macau (Grant No.: MYRG-GRG2024-00299-FST-UMDF, MYRG-GRG2025-00312-FST).

REFERENCES

[1] C. Min, L. Xiao, D. Zhao, Y. Nie, and B. Dai, "Uniscene: Multi-camera unified pre-training via 3d scene reconstruction for autonomous driving," *arXiv preprint arXiv:2305.18829*, 2023.

[2] Z. Yang, L. Chen, Y. Sun, and H. Li, "Visual point cloud forecasting enables scalable autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 673–14 684.

[3] 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.

[4] H. Yang, *et al.*, "Unipad: A universal pre-training paradigm for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 238–15 250.

[5] S. Sirko-Galouchenko, A. Boulch, S. Gidaris, A. Bursuc, A. Vobecky, P. Pérez, and R. Marlet, "Occfeat: Self-supervised occupancy feature prediction for pretraining bev segmentation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4493–4503.

[6] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 16 000–16 009.

[7] Y. Li, Y. Chen, X. Qi, Z. Li, J. Sun, and J. Jia, "Unifying voxel-based representation with transformer for 3d object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 18 442–18 455, 2022.

[8] 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.

[9] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized hd map construction," *arXiv preprint arXiv:2208.14437*, 2022.

[10] 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. 17 158–17 168.

[11] W. Tong, *et al.*, "Scene as occupancy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8406–8415.

[12] J. Zou, B. Liao, Q. Zhang, W. Liu, and X. Wang, "Mim4d: Masked modeling with multi-view video for autonomous driving representation learning," *arXiv preprint arXiv:2403.08760*, 2024.

[13] 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.

[14] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "Nerd: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 12 684–12 694.

[15] K. Park, U. Sinha, J. T. Barron, S. Bouaziz, D. B. Goldman, S. M. Seitz, and R. Martin-Brualla, "Nerfies: Deformable neural radiance fields," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5865–5874.

[16] G. Wang, P. Wang, Z. Chen, W. Wang, C. C. Loy, and Z. Liu, "Perf: Panoramic neural radiance field from a single panorama," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[17] M. Z. Irshad, S. Zakharov, K. Liu, V. Guizilini, T. Kollar, A. Gaidon, Z. Kira, and R. Ambrus, "Neo 360: Neural fields for sparse view synthesis of outdoor scenes," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 9187–9198.

[18] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering." *ACM Trans. Graph.*, vol. 42, no. 4, pp. 139–1, 2023.

[19] S. Zheng, B. Zhou, R. Shao, B. Liu, S. Zhang, L. Nie, and Y. 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*, 2024, pp. 19 680–19 690.

[20] S. Szymanowicz, C. Rupprecht, and A. Vedaldi, "Splatter image: Ultra-fast single-view 3d reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 10 208–10 217.

[21] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan, "Designing bert for convolutional networks: Sparse and hierarchical masked modeling," *arXiv preprint arXiv:2301.03580*, 2023.

[22] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[23] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[24] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 11 784–11 793.

[25] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 1090–1099.

[26] C. Yang, *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.

[27] J. Yan, Y. Liu, J. Sun, F. Jia, S. Li, T. Wang, and X. Zhang, "Cross modal transformer: Towards fast and robust 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 268–18 278.

[28] S. Wang, Y. Liu, T. Wang, Y. Li, and X. Zhang, "Exploring object-centric temporal modeling for efficient multi-view 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3621–3631.

[29] S. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, "Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 3047–3054.

[30] 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.

[31] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d, 2020.

[32] 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.

[33] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[34] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: An online hd map construction and evaluation framework," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4628–4634.

[35] Y. Liu, T. Yuan, Y. Wang, Y. Wang, and H. Zhao, "Vectormapnet: End-to-end vectorized hd map learning," in *International Conference on Machine Learning*. PMLR, 2023, pp. 22 352–22 369.

[36] 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.

[37] 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.

[38] 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.

[39] X. Tian, T. Jiang, L. Yun, Y. Wang, Y. Wang, and H. Zhao, "Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving," *arXiv preprint arXiv:2304.14365*, 2023.

[40] 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.