ODG: Occupancy Prediction Using Dual Gaussians

Yunxiao Shi[†] **Yinhao Zhu**[†] **Shizhong Han**[†] **Jisoo Jeong**[†] **Amin Ansari**[‡]
Hong Cai[†] **Fatih Porikli**[†]
†Qualcomm AI Research* ‡Qualcomm Technologies, Inc
{yunxshi,yinhaoz,shizhan,jisojeon,amina,hongcai,fporikli}@qti.qualcomm.com

Abstract

Occupancy prediction infers fine-grained 3D geometry and semantics from camera images of the surrounding environment, making it a critical perception task for autonomous driving. Existing methods either adopt dense grids as scene representation which is difficult to scale to high resolution, or learn the entire scene using a single set of sparse queries, which is insufficient to handle the various object characteristics. In this paper, we present ODG, a hierarchical dual sparse Gaussian representation to effectively capture complex scene dynamics. Building upon the observation that driving scenes can be universally decomposed into static and dynamic counterparts, we define dual Gaussian queries to better model the diverse scene objects. We utilize a hierarchical Gaussian transformer to predict the occupied voxel centers and semantic classes along with the Gaussian parameters. Leveraging the real-time rendering capability of 3D Gaussian Splatting, we also impose rendering supervision with available depth and semantic map annotations injecting pixel-level alignment to boost occupancy learning. Extensive experiments on the Occ3D-nuScenes and Occ3D-Waymo benchmarks demonstrate our proposed method sets new state-of-the-art results while maintaining low inference cost.

1 Introduction

3D spatial understanding forms the foundation of autonomous systems such as self-driving cars. 3D object detection [57, 23, 34, 33] has been the primary task that outputs bounding boxes to capture different entities in the scene. Concise as box representation is, it cannot deal with out-of-vocabulary or irregularly-shaped objects (*e.g.* trash can on the side of road, excavator with arms deployed) which is critical for driving safety. Calling for a unified 3D representation that can handle such cases, 3D occupancy adopts a voxel grid to partition the scene and jointly predicts the occupancy state and semantic labels of each voxel, which provides critical information for downstream planning [22].

Previous approaches have explored regular grids like voxel [58, 63], BEV ground plane [31, 61], and tri-perspective plane [25] to represent the scene followed by dense classification. Such approaches do not take into account the fact that most of the voxels are empty [3, 50] and allocate equal resource for each one, which inevitably results in severe inefficiency. To overcome such drawbacks, another line of research [47, 53] formulate the task of 3D occupancy as direct set prediction, effectively predicting 3D occupancy as set of sparse points from sparse latent vectors. Such sparse representation avoids spending resource to model empty regions and improves scalability. Recent works [27, 4, 10] utilize 3D Gaussians instead of points which have more spatial context, and also leverage the real-time rendering of 3D Gaussian Splatting (3DGS) [28, 29] to either complement learning from occupancy ground-truth or instead directly learn occupancy from 2D labels [59, 21] without relying on 3D annotations. Such sparse query-based methods have shown great promises for occupancy prediction.

*Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

But Is it really sensible to rely on a single set of queries to predict everything within a driving scene? We observe that driving scenes can be universally decomposed into static background (*e.g.* roads, buildings, etc.) and dynamic agents (*e.g.* vehicles, pedestrians, etc.), and argue that the dynamic agents should have resources dedicated to them given their importance in occupancy prediction. Hence, we introduce a dual Gaussian query design, consisting of two sets of Gaussian queries, to model the static and dynamic parts of the scene for better capturing complex scene dynamics. To establish communication between queries, we propose a simple and effective attention scheme to achieve this. Meanwhile for Gaussian-based representation, it is critical to have a sufficient number of Gaussians in order to have enough capacity to represent the scene. But existing methods [27, 4] utilize a single transformer which can only handle a smaller number of Gaussians. Therefore we propose to predict Gaussians in a coarse-to-fine manner with a series of transformer layers, enabling the use of a much larger number of Gaussians and thereby increasing model capacity. In addition to learning from 3D occupancy ground-truth, we leverage the efficient rendering capabilities of 3DGS to generate the depth and semantic maps for each camera view at every stage, which are supervised by the corresponding 2D labels and improves model consistency. Our contributions can be summarized as follows:

- **Dual Gaussian Query Design:** We propose a novel dual-query architecture comprising two distinct sets of Gaussian queries to separately model the static and dynamic parts of the scene. A cross query attention is also introduced to establish effective interaction between queries, enhancing 3D occupancy prediction.
- **Hierarchical Coarse-to-Fine Refinement:** We refine the Gaussian properties in a hierarchical coarse-to-fine fashion, allowing a much larger number of 3D Gaussians to be utilized, effectively increasing model capacity and expressiveness.
- **Multi-Stage Rendering Supervision:** We enforce depth and semantic consistency through multi-stage real-time rendering using 3D Gaussian Splatting. This allows supervision from 2D labels across views, improving spatial coherence and prediction accuracy.
- **State-of-the-Art Performance:** Extensive experiments on the Occ3D [50] benchmark demonstrates that our proposed method sets new state-of-the-art results in 3D occupancy prediction, while maintaining competitive inference runtime.

2 Related Works

2.1 3D Occupancy Prediction

3D occupancy prediction partitions the 3D space with a voxel grid and simultaneously estimates the occupancy state and semantic label of each voxel. Earlier methods [7, 31, 25, 58, 63] utilize dense grids (*e.g.* voxel, BEV) as scene representation and learns 3D occupancy from ground-truth data. Given the high cost [56] of curating occupancy annotations, [62, 41, 5, 44] advocates the idea of using 2D labels projected from LiDAR or generated by vision foundation models [59, 21] to train 3D occupancy networks eliminating reliance on direct 3D occupancy annotations. Several works [8, 62, 24, 36] also explore self-supervised learning based on the photometric consistency between neighboring frames to learn 3D occupancy. Meanwhile, multiple 3D occupancy benchmarks [3, 58, 50, 51, 13, 56, 64] have been created based on existing datasets [17, 16, 6, 45].

2.2 Set Prediction with Transformers

Direct set prediction with transformers [52, 14] for perception tasks was first introduced in DETR [9]. Follow-up works like [57, 37, 34, 33] further advanced this technique for 3D object detection and obtained impressive results. Witnessing such success, [47, 53] adapted such set prediction paradigm for 3D occupancy prediction and demonstrated a competitive alternative to the common pipeline of explicit space modeling (*e.g.*, dense grids) followed by classification. We adopt such paradigm but different from previous works [34, 47, 53] that only use one set of queries assuming scene homogeneity, we define two distinctive sets of queries to better handle the dynamic agents in the scene.

2.3 Neural Rendering and 3D Gaussian Splatting

Neural rendering [48] aims to learn 3D representations from 2D data and experienced significant growth over the past few years [49]. Neural Radiance Fields (NeRF) [40, 1, 2] is one such technique

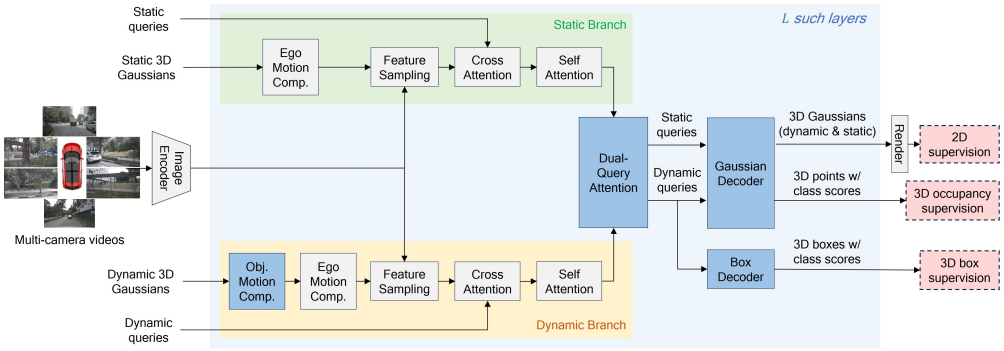

Figure 1: Overview of proposed ODG, where we model the dynamic and static elements of the scene with two separate sets of Gaussian queries. Dual-query attention aggregates information across dynamic and static queries. These queries are then decoded into 3D Gaussians, 3D points, as well as 3D bounding boxes (from dynamic queries only), which are supervised by ground-truth depth and semantic maps through rendering, ground-truth 3D occupancy, and ground-truth bounding boxes, respectively.

and has achieved impressive results. However, NeRF-based methods suffer from slow training and high memory cost. Recently, 3D Gaussian Splatting (3DGS) [28, 29] emerged as a rasterization pipeline which drastically reduced rendering time while also achieving incredible rendering quality. Earlier methods [60, 19] focused on improving per-scene rendering quality. Later generalizable 3DGS [11, 12, 46] were proposed to enable reconstructing large-scale scenes in a feed-forward manner. Given that 3D occupancy also aims to reconstruct the scene, latest research begin exploring 3D Gaussians for occupancy prediction [26, 27, 10, 4]. GaussianFormer [26] and GaussRender [10] voxelizes 3D Gaussians which incurs high compute cost. GaussTR [27] and GaussianFlowOcc [4] employs a single transformer to predict Gaussian parameters from sparse queries, which can only handle small number of Gaussians limiting model capability. In contrast, our method predicts Gaussians in a hierarchical coarse-to-fine fashion allowing a much larger number of Gaussians, effectively resulting in higher learning capacity.

3 Method

In this section, we present our proposed 3D occupancy approach, ODG, in which we adopt a dual Gaussian query design to capture the respective dynamic and static elements in the scene, as discussed in Sec. 3.2. We model object motions for the dynamic Gaussians (Sec. 3.3) and leverage attention to enable feature interaction between the dual queries (Sec. 3.4). Finally, we describe the training objectives in Sec. 3.5. Fig. 1 provides an overview of ODG. Next, we first provide a quick recap on the problem of 3D occupancy prediction.

3.1 Problem Definition

Given an ego-vehicle at time T, the task of 3D occupancy prediction takes N_c multi-camera images (with $k \times N_c$ optional history frames where $k \geq 0$), $\mathbf{I} = \{I_c^t\}_{t=T-k,c=1}^{T,N_c}$ and corresponding camera parameters as input, and predicts a 3D semantic voxel grid $\mathbf{O} = \{c_1, c_2, \ldots, c_C\}^{H \times W \times Z}$ where H, W, Z denotes the grid resolution and C is the number of classes. Formally, 3D occupancy prediction can be defined as

$$\mathbf{O} = G(\mathbf{V}), \quad \mathbf{V} = F(\mathbf{I}), \tag{1}$$

where $F(\cdot)$ consists of an image backbone that extract multi-camera features and transforms them into scene representation $\mathbf{V}$. $G(\cdot)$ is another neural network that maps $\mathbf{V}$ to final occupancy predictions. Dense grids (*e.g.* voxel grid [58, 63], BEV/TPV grid [31, 25]) are common choices for $\mathbf{V}$ which are difficult to scale and cannot differentiate different object scales. Inspired by successes in object detection [9, 57], more recent works [35, 53] cast 3D occupancy prediction as direct set prediction with transformers, without explicitly building $\mathbf{V}$. At its core, a set of learnable queries $\mathbf{Q}$ and 3D points $\mathbf{P}$ are initialized to regress point locations and corresponding semantic class scores $\mathbf{C}$ simultaneously. Such a paradigm can be described as

$$\min_{\mathbf{P},\mathbf{C}} D_p(\mathbf{P}, \mathbf{P}_g) + D_c(\mathbf{C}, \mathbf{C}_g), \tag{2}$$

where $\{\mathbf{P}_g, \mathbf{C}_g\}$ is the constructed ground-truth set for occupied voxels with $|P_g| = |C_g| = V_g$ being the number of occupied voxels. Each $p_g \in \mathbf{P}_g$ represents the voxel center coordinate and $c_g \in \mathbf{C}_g$ the class label. Correspondingly, $\{\mathbf{P}, \mathbf{C}\}$ denotes predictions where each $p_i \in \mathbf{P}$ and $c_i \in \mathbf{C}$ is predicted by query $q_i \in \mathbf{Q}$. $D_p(\cdot)$ and $D_c(\cdot)$ are geometric and semantic distances respectively.

3.2 Dual Dynamic and Static Gaussian Queries

Instead of predicting occupancy as point sets from one set of learnable queries as in Eq. 2, we define dual Gaussian queries as shown in Fig. 1 which are detailed below.

Formally, a standard 3D Gaussian $\mathbf{g}$ is parameterized by a set of properties

$$\mathbf{g} = \{\boldsymbol{\mu}, \mathbf{s}, \mathbf{r}, \sigma\}, \tag{3}$$

where $\boldsymbol{\mu} \in \mathbb{R}^3$ is the mean, $\mathbf{s} \in \mathbb{R}^3$ is the scale, $\mathbf{r} \in \mathbb{R}^4$ is the quaternion and $\sigma \in [0, 1]$ is the opacity. We define two sets of Gaussian queries for static and dynamic objects, each query contains a set of Gaussians and a feature vector

$$\mathbf{G}^s = \{\mathbf{g}^s_{i,k}\}^{S,K_\ell}_{i=1,k=1},\ \mathbf{Q}^s = \{\mathbf{q}^s_i\}^S_{i=1}, \mathbf{q}^s_i \in \mathbb{R}^M;\ \mathbf{G}^d = \{\mathbf{g}^d_{j,k}\}^{D,K_\ell}_{j=1,k=1},\ \mathbf{Q}^d = \{\mathbf{q}^d_j\}^D_{j=1}, \mathbf{q}^d_j \in \mathbb{R}^N, \tag{4}$$

where S, D is the number of static and dynamic Gaussian queries, and N, M is the embedding dimension of query features, K_ℓ is the number of Gaussians per query in stage ℓ. Here we set $N = M$ for simplicity but they can be different.

For a dynamic query $\mathbf{g}^d$, whose intended purpose is to model the dynamic agents in the scene, what additional information should they have on top of Eq. 3 that can effectively accomplish this? We observe that the box representation from 3D object detection [57, 34, 33] is a good candidate that is tailored to capture dynamic objects. Therefore we expand the standard definition in Eq. 3 and define our dynamic Gaussian queries $\mathbf{g}^d$ as

$$\mathbf{g}^d = \mathbf{g}^s \oplus \mathbf{b}, \quad \mathbf{b} = [l, w, h, \theta, v_x, v_y, v_z], \tag{5}$$

where l, w, h are the spatial dimensions of the 3D box and θ its rotation. v_x, v_y, v_z is the velocity vector. Given for driving scenes, motion along the z-axis is negligible hence we set $v_z = 0$ and treat it as constant.

We initialize Gaussian means $\mathbf{g}^s_{:\mu}$ and $\mathbf{g}^d_{:\mu}$ from $\mathcal{U}[0, 1]$ for both types of queries and the box attributes $\mathbf{b}$ for $\mathbf{g}^d$ but leave the rest of the Gaussian properties uninitialized. Instead we predict them using separate MLPs with their means and query features

$$\{\mathbf{s}^d, \mathbf{r}^d, \sigma^d\} = \Phi(\mathbf{G}^d_{:\mu}, \mathbf{Q}^d),\ \{\mathbf{s}^s, \mathbf{r}^s, \sigma^s\} = \Phi(\mathbf{G}^s_{:\mu}, \mathbf{Q}^s), \tag{6}$$

where Φ denotes respective MLPs. To enable an sufficient number of Gaussians, unlike previous methods [27, 4] which only utilizes a single transformer that maintains the same number of points over layers, we predict Gaussians in a hierarchical coarse-to-fine fashion through a series of transformer-based layers $\mathcal{T}$ akin to [34, 53]

$$\mathbf{G}^{d,b}_\ell,\ \mathbf{C}^{d,b}_\ell,\ \mathbf{G}^d_\ell,\ \mathbf{C}^d_\ell,\ \mathbf{Q}^d_\ell = \mathcal{T}_\ell(\mathbf{G}^d_{:\mu\oplus\mathrm{b},\ell-1}, \mathbf{Q}^d_{\ell-1}; \Phi_s, \Phi_r, \Phi_\sigma), \tag{7}$$

$$\mathbf{G}^s_\ell,\ \mathbf{Q}^s_\ell,\ \mathbf{C}^s_\ell = \mathcal{T}_\ell(\mathbf{G}^s_{:\mu,\ell-1}, \mathbf{Q}^s_{\ell-1}; \Phi_s, \Phi_r, \Phi_\sigma), \tag{8}$$

where $1 \le \ell \le L$ with L being the number of layers used. For each layer $\mathcal{T}_\ell$, it takes as input static Gaussian means $\mathbf{G}^s_{:\mu,\ell-1}$ and query features $\mathbf{Q}^s_{\ell-1}$ from the previous layer, and predict the current static Gaussian means $\mathbf{G}^s_{:\mu,\ell} \in \mathbb{R}^{S\times K_\ell\times 3}$ and corresponding class scores $\mathbf{C}^s_\ell \in \mathbb{R}^{S\times K_\ell\times C}$, where $K_{\ell-1} < K_\ell$ is the number of Gaussians per query at each stage which enables coarse-to-fine prediction. For dynamic Gaussians, besides decoding $\mathbf{G}^d_\ell \in \mathbb{R}^{D\times K_\ell\times 3}$ and $\mathbf{C}^d_\ell \in \mathbb{R}^{D\times K_\ell\times C}$, we also parallelly predict box attributes and class scores $\mathbf{G}^{d,b}_\ell \in \mathbb{R}^{D\times 10}$ and $\mathbf{C}^{d,b}_\ell \in \mathbb{R}^{D\times C}$ as shown in Eq. 7. We note that we do not perform coarse-to-fine prediction for boxes given the relatively small number of boxes that are typically present in the scene. The rest of the Gaussian properties is predicted by separate MLPs Φ as defined in Eqs. 6.

We aim to reduce the spatial artifacts of 3D occupancy through projective constraints. To this end, we render depth and semantic maps of each camera view at current time (keyframe) which effectively

enforces geometric and semantic consistency. We leverage 3D Gaussian splatting [28, 29] which offers real-time rendering:

$$\hat{D}_p = \sum_{i=1}^{G} T_i \sigma_i \mathbf{d}_i, \quad \hat{S}_p = \sum_{i=1}^{G} T_i \alpha_i \mathbf{c}_i, \tag{9}$$

where p indicates a certain pixel location and $G = D + S$ is the number of Gaussians. σ_i is the opacity and T_i is accumulated transmittence. $\mathbf{c}_i \in [0, 1]^C$ is the "semantic probability" of the i-th Gaussian and $\mathbf{d}_i$ is the depth of the i-th Gaussian. $\hat{D}_p, \hat{S}_p$ are the rendered depth and semantic values at location p. We refer readers to [28] for further details regarding 3D Gaussian splatting.

3.3 Motion Modeling of Dynamic Gaussians

For each Gaussian $\mathbf{g}$ with $\mathbf{g}_{:\mu}$ representing its position in 3D space, it first samples 3D points following [34, 53]. Then we sample image features by projecting sampled 3D points onto each image feature plane using available camera extrinsics and intrinsics, and aggregate corresponding image features. Under the setting of having history frames, it is critical to move the Gaussians according to its motion to sample features correctly. For driving scenes we consider two types of motions: 1) ego-motion which is the motion of the ego-vehicle as it navigates across the scene, which effectively is camera motion, and 2) object-motion which describes how the dynamic agents themselves move in the scene.

For static Gaussians, compensating ego-motion is enough. For the motion of dynamic Gaussians, we adopt the approach of approximating instantaneous velocity with average velocity in a short-time window as in [34], and warp sampling points to previous timestamps t_{-i} using the velocity vector $[v_x, v_y]$ from the dynamic Gaussian queries (Eq. 5)

$$x_{-i} = x_0 - v_x \cdot (t_0 - t_{-i}), \tag{10}$$

$$y_{-i} = y_0 - v_y \cdot (t_0 - t_{-i}), \tag{11}$$

where t_0 is the current timestamp. We note that on the z-axis, we assume there is no velocity give the nature of driving. Then we warp each dynamic Gaussian using camera motion

$$(x'_{-i}, y'_{-i}, z'_{-i}, 1)^\top = P_{-i}^{-1} P_0 (x_{-i}, y_{-i}, z_{-i}, 1)^\top,$$

where P_{-i}, P_0 are the ego poses at timestamp t_{-i} and t_0.

3.4 Attention across Dynamic and Static Queries

To enable effective interaction between dynamic Gaussian queries $\mathbf{Q}^d$ and static Gaussian queries $\mathbf{Q}^s$, we first concatenate their features representations. We then apply Self-Attention [52] to the combined features, allowing for rich information exchange cross both query types. We refer to this mechanism as Dynamic-and-Static (DaS) Attention. This approach not only facilitates self-attention within the dynamic and static queries individually but also enables bidirectional cross-attention between them, enhancing the integration of dynamic and static information.

$$\mathbf{Q} = \text{Self-Attention}(\text{Concatenate}(\mathbf{Q}^d, \mathbf{Q}^s)), \tag{12}$$

$$\mathbf{Q}^d = \mathbf{Q}_{:D}, \; \mathbf{Q}^s = \mathbf{Q}_{D:D+S}, \tag{13}$$

where : here is the index operator.

3.5 Loss Functions

We supervise predicted Gaussian means $\mathbf{G}_{:\mu}$ and corresponding class scores $\mathbf{C}$ with Chamfer distance [15] and focal loss [32]

$$\mathcal{L}_{occ} = \text{CD}(\mathbf{G}_{:\mu,0}, \mathbf{P}_g^0) + \sum_{\ell=1}^{L} \text{CD}(\mathbf{G}_{:\mu,\ell}, \mathbf{P}_g^\ell) + \text{FocalLoss}(\mathbf{C}_\ell, \mathbf{C}_g^\ell), \tag{14}$$

where $\text{CD}(\mathbf{G}_0^\mu, \mathbf{P}_g^0)$ encourages initial Gaussian means to capture global pattern of underlying data as pointed out in [53]. For the simplicity of notation, we do not differentiate between static and dynamic Gaussians in Eq. 14.

For box predictions by dynamic Gaussians, $\mathcal{L}_1$ loss is used to supervised box attributes defined in Eq. 5 and focal loss is used to supervise corresponding box class labels

$$\mathcal{L}_{box} = \sum_{\ell=1}^{L} \mathcal{L}_1(\mathbf{G}_\ell^{d,b}, \mathbf{B}_g^\ell) + \text{FocalLoss}(\mathbf{C}_\ell^{d,b}, \mathbf{C}_{g,b}^\ell), \tag{15}$$

where $\mathbb{B}, \mathbf{C}_{g,b}$ denotes ground-truth. Label assignment is done using the Hungarian algorithm [30] during training. Hence, the 3D Loss can be written as

$$\mathcal{L}_{3d} = \mathcal{L}_{occ} + \lambda_{3d}\mathcal{L}_{box}, \tag{16}$$

where λ_{3d} is the weighting factor.

For rendered depth and semantic maps from Gaussians at all stages, we supervise depth with $\mathcal{L}_1$ loss and semantics with cross-entropy loss

$$\mathcal{L}_r = \sum_{\ell=1}^{L} \mathcal{L}_1(\hat{D}_\ell, \bar{D}) + \text{CE}(\hat{S}_\ell, \bar{S}), \tag{17}$$

where $\hat{D}, \hat{S}$ are the rendered depth and semantic maps with $\bar{D}, \bar{S}$ being the ground-truth. Here we project LiDAR points with their ground-truth semantic labels that are available from datasets [6, 45] onto each camera view to obtain $\bar{D}, \bar{S}$. Therefore the final loss can be written as

$$\mathcal{L} = \mathcal{L}_{3d} + \lambda\mathcal{L}_r, \tag{18}$$

where λ is the weighting factor.

4 Experiments

4.1 Experiment Setup

Datasets: We evaluate our model on the Occ3D benchmark [50] which bootstraps the nuScenes [6] and Waymo-Open [45] dataset.[1] nuScenes consists of 1,000 scenes with a split of $700/150/150$ for training, validation and testing. Occ3D-nuScenes annotates 3D occupancy ground-truth providing 17 semantic classes. The voxel grid range is $[-40\text{m}, -40\text{m}, -1\text{m}, 40\text{m}, 40\text{m}, 5.4\text{m}]$ along the X, Y and Z axis with a grid resolution of $200 \times 200 \times 16$ and voxel size of 0.4m. The original image resolution is 900×1600. Waymo Open [45] has 798 training scenes and 202 validation scenes. Occ3D-Waymo provides 3D semantic occupancy ground-truth of 15 semantic classes with 1 class being *General Object (GO)*. The voxel grid resolution and voxel size is the same as Occ3D-nuScenes. On Waymo, the original image resolution is 1280×1920 for the front, front-left and front-right cameras. For the side-left and side-right cameras, the original image resolution is 1040×1920.

Evaluation Metrics: We evaluate our model under the mIoU and RayIoU [47] metric:

$$\text{mIoU} = \frac{|P \cap G|}{|P \cup G|}, \quad \text{RayIoU} = \frac{\sum_{r \in \mathcal{R}} |P_r \cap G_r|}{\sum_{r \in \mathcal{R}} |P_r \cup G_r|},$$

where P, G are the set of occupied voxels in prediction and ground-truth respectively. $\mathcal{R}$ is the set of all emulated LiDAR rays, and P_r, G_r are the sets of occupied voxels intersected by ray r in prediction and ground-truth respectively.

Implementation Details: We implement our proposed method in PyTorch [42]. Following previous works [35, 53, 4], we use ResNet-50 [20] as image backbone to extract multi-camera image features. On nuScenes, we resize input images to the resolution of 256×704. On Waymo, all input images are resized and padded to 640×960. For Ours-tiny, we set number of static Gaussian queries $S = 500$ and number of dynamic Gaussian queries $D = 100$. For Ours-large, we set $S = 4000$ and $D = 800$, respectively. We use $L = 6$ transformer layers to conduct coarse-to-fine prediction. We set $\lambda_{3d} = 0.2$ to balance box loss $\mathcal{L}_{box}$ and occupancy loss $\mathcal{L}_{occ}$. For rendering loss L_r, we set $\lambda = 0.05$ for stage $\ell = 1, 6$, and $\lambda = 0.01$ for the rest. We use AdamW [38] as the optimizer with weight decay of 0.01. We train all our models with an initial learning rate of 2×10^{-4} and decays with CosineAnnealing [39] schedule. For experiments on Waymo, we sample 20% of the data matching practices in previous works [53, 50]. Unless otherwise specified, we train all our models with a global batch size of 8 for 100 epochs using NVIDIA A100 GPUs. During inference, we adopt the standard practice and make use of the camera visibility masks provided by the dataset [50] and only evaluate in unoccluded regions. Inference runtime is measured on a single idle A100 GPU with PyTorch `fp32` backend.

[1] nuScenes is under a CC BY-NC-SA 4.0 license and Waymo license terms can be found here: `https://waymo.com/open/terms/`.

Table 1: 3D semantic occupancy results on Occ3D-nuScenes validation set [6, 50]. Cons. Veh stands for "Construction Vehicle" and Dri. Sur stands for "Drivable Surface". We note that for fair comparison, both ODG-T and ODG-L here are trained without using future frames. **Bold**/Underline: Best/second best results. *indicates self-supervised methods.

Method	mIoU	Others	Barrier	Bicycle	Bus	Car	Cons. Veh	Motorcycle	Pedestrian	Traffic Cone	Trailer	Truck	Dri. Sur	Other flat	Sidewalk	Terrain	Manmade	Vegetation	RayIoU	FPS
RenderOcc [41]	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	19.5	3.0
GaussRender [10]	30.38	8.87	40.98	23.25	43.76	46.37	19.49	25.2	**23.96**	19.08	25.56	33.65	58.37	33.28	36.41	33.21	22.76	22.19	37.5	-
GaussTR* [27]	12.27	-	6.5	8.54	21.77	24.27	6.26	15.48	7.94	1.86	6.1	17.16	36.98	-	17.21	7.16	21.18	9.99	-	-
SparseOcc (8f) [47]	30.1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	34.0	17.3
SparseOcc (16f) [47]	30.6	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	35.1	12.5
OPUS-T (8f) [53]	33.2	10.72	39.82	21.27	39.76	45.25	23.41	21.80	17.81	19.26	27.48	33.20	71.61	37.12	45.13	43.59	33.80	33.18	38.4	**22.4**
OPUS-L (8f) [53]	36.2	11.95	43.45	25.51	40.95	47.24	23.86	25.89	21.26	**29.06**	30.13	35.28	73.13	**41.08**	**47.01**	45.66	37.40	**35.27**	41.2	7.2
GaussianFlowOcc* [4]	16.02	-	7.23	9.33	17.55	17.94	4.5	9.32	8.51	10.66	2.00	11.80	63.89	-	31.11	35.12	14.64	12.59	16.47	10.2
ODG-T (8f)	35.54	13.69	38.97	23.02	46.75	49.33	25.79	23.63	20.73	18.54	30.01	35.61	76.84	39.33	45.01	46.78	37.45	32.24	39.2	20.1
ODG-L (8f)	**38.18**	**14.11**	**46.62**	**27.09**	**48.77**	**52.09**	**26.79**	**28.05**	23.21	27.92	**30.86**	**38.17**	**77.13**	40.35	46.94	**47.37**	**40.01**	33.52	**42.3**	4.9

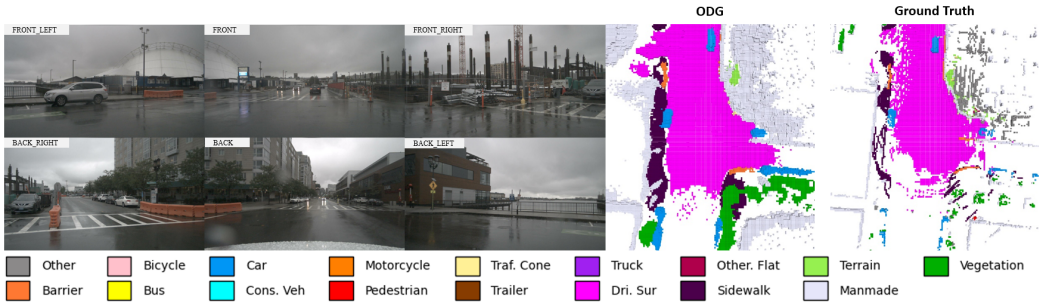

Figure 2: Visualization of ODG prediction on the Occ3D-nuScenes [50, 6] validation set. The ODG can capture all the vehicles on a gloomy rainy day.

4.2 Evaluation Results

In this section, we report evaluation results on the Occ3D benchmark [50] and compare with latest state-of-the-art methods.

nuScenes Our results on Occ3D-nuScenes is summarized in Tab. 1. One can see that our method achieves new state-of-the-art results in terms of both mIoU and RayIoU, while maintaining competitive inference speed even when compared to latest efficient approaches. Specifically, ODG-T (8f) achieves an mIoU of 35.54 with a RayIoU of 39.2, outperforming OPUS-T (8f) who has an mIoU of 33.2 (-2.34) and a RayIoU of 38.4 (-0.8), while still having an inference runtime of 20.1 FPS. Similarly, ODG-T also easily outperforms both SparseOcc (8f), SparseOcc (16f) and GaussRender with significant margins. Meanwhile, our heavy variant ODG-L sets new best result eventually obtaining an mIoU of 38.18 with a RayIoU of 42.3, surpassing previous best with a definitive margin of +1.98 and +1.1, respectively.

It is worth noting that given our specific design to attend to the dynamic agents in the scene, we show significant improvement when examining the key dynamic object classes. As shown in Tab. 2, for the classes of *Bus, Car, Construction Vehicle (Cons. Veh), Motorcycle*, and *Truck*, ODG-L carries a significant lead of +4.13 for mIoU, once again demonstrating the efficacy of our proposed strategy of handling dynamic agents.

Table 2: Occupancy prediction results over key dynamic object classes on Occ3D-nuScenes [50] validation set. ODG achieves consistent improvement across all dynamic categories. **Bold**/Underline: Best/second best results.

Method	mIoU	Bus	Car	Cons. Veh	Motorcycle	Truck
GaussRender [10]	33.69	43.76	46.37	19.49	25.2	33.65
OPUS-T (8f) [53]	32.68	39.76	45.25	23.41	21.80	33.20
OPUS-L (8f) [53]	34.64	40.95	47.24	23.86	25.89	35.28
ODG-T (8f)	36.22	46.75	49.33	25.79	23.63	35.61
ODG-L (8f)	**38.77**	**48.77**	**52.09**	**26.79**	**28.05**	**38.17**

Table 3: 3D semantic occupancy results on Occ3D-Waymo validation set [45, 50]. GO stands for "General Object". Traf. Light stands for "Traffic Light" and Cons. Cone stands for "Construction Cone".

Method	mIoU	GO	Vehicle	Bicyclist	Pedestrian	Sign	Traf. Light	Pole	Cons. Cone	Bicycle	Motorcycle	Building	Vegetation	Tree Trunk	Road	Walkable	RayIoU	FPS
BEVDet [23]	9.88	0.13	13.06	2.17	10.15	7.80	5.85	4.62	0.94	1.49	0.00	7.27	10.06	2.35	48.15	34.12	-	-
BEVFormer [31]	16.76	3.48	17.18	13.87	5.9	13.84	2.7	9.82	12.2	13.99	0.00	13.38	11.66	6.73	74.97	51.61	-	-
TPVFormer [25]	16.76	3.89	17.86	12.03	5.67	13.64	8.49	8.90	9.95	14.79	0.32	13.82	11.44	5.8	73.3	51.49	-	4.6
CTF-Occ [50]	18.73	**6.26**	28.09	14.66	8.22	15.44	**10.53**	11.78	**13.62**	**16.45**	**0.65**	18.63	17.3	8.29	67.99	42.98	-	2.6
OPUS-L [53]	19.00	4.66	27.07	19.39	6.53	**18.66**	6.41	11.44	10.40	12.90	0.00	18.73	**18.11**	7.46	72.86	50.31	24.7	**8.5**
ODG-L	**21.35**	5.09	**31.34**	**22.4**	**19.06**	15.24	6.09	**12.51**	12.77	13.59	0.00	**21.49**	17.89	**8.37**	**78.19**	**56.28**	**25.9**	5.6

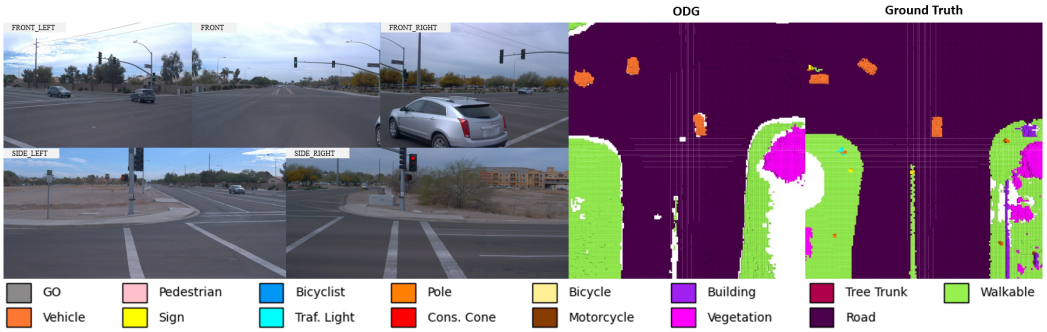

Figure 3: Visualization of ODG prediction on the Occ3D-Waymo [50, 45] validation set.

Waymo We further evaluate our ODG on the Occ3D-Waymo dataset and the results are presented in Tab. 3. We note that Occ3D-Waymo is a much less well evaluated occupancy benchmark especially for camera-only methods, given its challenging conditions (*e.g.* almost no view overlap between cameras). We train ODG-L of with 20% of the training data following practice in previous methods [53, 54]. To the best of our knowledge, the most comprehensive source which compiles vision-only methods are from [50, 53], hence we use them for comparison. It is clear from Tab. 3 that our ODG obtains a definitive lead of +2.35 for mIoU and +1.2 for RayIoU when compared to OPUS. Upon further examining Tab. 3 one can see that ODG establishes far surpass the second-best under class *Vehicle(+3.25), Bicyclist(+3.01)* and *Pedestrians(+8.91)*, which are all critical traffic agents, once again proving the superiority of our modeling of dynamic objects.

4.3 Visualization

We showcase visualizations of predicted occupancy of ODG on nuScenes and Waymo in Fig. 2 and Fig. 3. In Fig. 2, under gloomy rainy weather where the ground-truth is sparse especially in object-centric categories (*e.g.* cars in this case), ODG effectively outputs dense predictions capturing all the cars in the scene. In Fig. 3, ODG also captures all the traffic agents. We attribute this attractive behavior to our dynamic Gaussian queries dedicated to modeling the dynamic scene agents.

4.4 Ablation Studies

In this section, we conduct multiple ablation studies to analyze the effects of various components in our proposed ODG. For all our ablation studies, we adopt ODG-T and train on the Occ3D-nuScenes for 24 epochs.

Temporal Alignment As pointed in Sec. 3.3, in order to sample features correctly from the history frames, it is critical to move the Gaussians according to their respective motions before projecting them onto image feature plane. Hence, here we study the effects of performing different type of motion compensation. As shown in Tab. 5b, if we merely correct ego-motion (camera-motion) for the dynamic Gaussians, both mIoU and RayIoU suffers a noticeable drop of -0.71 and -0.5 respectively. This demonstrates it is essential to compensate object motion for dynamic agents.

Query Interaction Given we have dynamic and static Gaussians modeling the scene, it is important to make them aware of each other. We analyze the effect of different attention mechanisms in Tab. 5a. In our experiments, we first tried performing cross attention with dynamic query features serving as queries, and static query features as keys and values, which gave us a modest improvement. A more straight-forward way is to concatenate the dynamic and static query features together, and perform self attention on top of concatenated features. This lead to further better results as shown in Tab. 5a.

Table 4: Impact of different components inside ODG on model performance.

Motion compensation	Query attention	Rendering Sup	mIoU	RayIoU$_{1m}$	RayIoU$_{2m}$	RayIoU$_{4m}$
			30.80	27.9	36.6	42.0
✓			31.78	28.7	37.3	42.6
✓	✓		32.13	29.3	37.8	43.1
✓	✓	✓	32.82	31.1	40.5	43.8

Table 5: Ablation studies on components related to dynamic Gaussian queries.

(a) Effects of Query Attention.

Query Attention	mIoU	RayIoU
Cross Attn	31.95	36.3
Self Concat Attn	32.13	36.7

(b) Effects of Motion Compensation.

Ego Comp.	Dyn. Comp	mIoU	RayIoU
✓		31.17	35.7
✓	✓	31.78	36.2

We posit that running self attention on all features in an exhaustive manner makes all queries become aware of each other therefore facilitating more information flow, leading to improved results.

Rendering Supervision To reduce spatial artifacts through projective constraints, we leverage 3D Gaussian Splatting and render depth and semantic maps to each camera view. As shown in Tab. 4, this resulted in a noticeable improvement both in mIoU and RayIoU, with mIoU+0.69 and RayIoU+0.70. By injecting the auxiliary supervision signals from LiDAR points projected onto each camera view (keyframes only, constrained by sensor calibration), it effectively enforces geometric and semantic consistency especially during the early stages of rendering, which helps the model learn more effectively for the regions that the 3D occupancy ground-truth are ambiguous or noisy. It is worth noting that during inference, we turn off rendering hence incurring no extra computation cost, reaping the benefit with zero increase in inference latency.

We summarize the effect of the different components in our proposed method in Tab. 4. It is evident that by progressively enabling different modules in ODG, the model performs increasingly well, validating the soundness of the designs that we incorporated into our system.

5 Conclusions and Discussions

In this paper, we present ODG, a novel approach to 3D occupancy prediction based on dual Gaussian queries. ODG defines two distinctive sets of queries consisting of dynamic and static Gaussian queries, aiming to better model dynamic scene agents. To make ODG attend to moving objects, we expand the standard 3D Gaussian properties of dynamic queries with 3D bounding box attributes, which effectively guides queries to dedicate resource to capture complex scene dynamics. Self attention is utilized to establish connection between dynamic and static queries. To enable sufficient model capacity, ODG adopts a coarse-to-fine prediction paradigm when it comes to predicting the Gaussian parameters, which allows a large number of Gaussians to be utilized. To reduce spatial artifacts in 3D occupancy, we enforce projective constraints through rendering depth and semantic maps to each camera view leveraging 3D Gaussian Splatting, supervised by projected LiDAR points. Our extensive experiments on the Occ3D-nuScenes and Occ3D-Waymo benchmark demonstrates ODG sets new state-of-the-art results while maintaining highly competitive efficiency.

Limitations. However, as promising as ODG is, it does not come without limitations. Readers might have observed that currently the Gaussian parameters other than mean (namely scale, rotation and opacity) is only optimized through rendering loss, rather than for instance, aggregating nearby Gaussians to get occupancy and learn from occupancy ground-truth as well, albeit such aggregation incurs significant cost. Therefore exploring ways to improve optimization of Gaussians would be an interesting research direction.

Broader Impacts. Our proposed ODG provides more accurate 3D occupancy prediction while maintaining inference efficiency, which is beneficial to safe, energy-efficient autonomous driving. Furthermore, given the versatility and efficiency of our ODG, we think it is a promising venue towards building a unified perception and prediction module for autonomous driving, which will improve the homogeneity of the entire stack that will in turn lower overall operating cost, yielding economic benefit. On the other hand, during the development of this work, several experiments did not make it into the paper (*e.g.* due to lack of time), which incurred extra energy consumption.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5855–5864, 2021.

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022.

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.

[4] Simon Boeder, Fabian Gigengack, and Benjamin Risse. Gaussianflowocc: Sparse and weakly supervised occupancy estimation using gaussian splatting and temporal flow. *arXiv preprint arXiv:2502.17288*, 2025.

[5] Simon Boeder, Fabian Gigengack, and Benjamin Risse. Occflownet: Towards self-supervised occupancy estimation via differentiable rendering and occupancy flow. *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[7] Anh-Quan Cao and Raoul De Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022.

[8] Anh-Quan Cao and Raoul de Charette. Scenerf: Self-supervised monocular 3d scene reconstruction with radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9387–9398, 2023.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Loick Chambon, Eloi Zablocki, Alexandre Boulch, Mickael Chen, and Matthieu Cord. Gaussrender: Learning 3d occupancy with gaussian rendering. *arXiv preprint arXiv:2502.05040*, 2025.

[11] David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19457–19467, 2024.

[12] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European Conference on Computer Vision*, pages 370–386. Springer, 2024.

[13] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. `https://github.com/OpenDriveLab/OpenScene`, 2023.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[16] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[18] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

[19] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5354–5363, 2024.

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

[21] Mu Hu, Wei Yin, Chi Zhang, Zhipeng Cai, Xiaoxiao Long, Hao Chen, Kaixuan Wang, Gang Yu, Chunhua Shen, and Shaojie Shen. Metric3d v2: A versatile monocular geometric foundation model for zero-shot metric depth and surface normal estimation. *arXiv preprint arXiv:2404.15506*, 2024.

[22] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.

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

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

[25] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9223–9232, 2023.

[26] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In *European Conference on Computer Vision*, pages 376–393. Springer, 2024.

[27] Haoyi Jiang, Liu Liu, Tianheng Cheng, Xinjie Wang, Tianwei Lin, Zhizhong Su, Wenyu Liu, and Xinggang Wang. Gausstr: Foundation model-aligned gaussian transformer for self-supervised 3d spatial understanding. *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[28] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[29] Bernhard Kerbl, Andreas Meuleman, Georgios Kopanas, Michael Wimmer, Alexandre Lanvin, and George Drettakis. A hierarchical 3d gaussian representation for real-time rendering of very large datasets. *ACM Transactions on Graphics (TOG)*, 43(4):1–15, 2024.

[30] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[31] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, pages 1–18. Springer, 2022.

[32] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017.

[33] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion. *arXiv preprint arXiv:2211.10581*, 2022.

[34] Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18580–18590, 2023.

[35] Haisong Liu, Haiguang Wang, Yang Chen, Zetong Yang, Jia Zeng, Li Chen, and Limin Wang. Fully sparse 3d panoptic occupancy prediction. In *Proceedings of the European Confernece on Computer Vision*, 2024.

[36] Yili Liu, Linzhan Mou, Xuan Yu, Chenrui Han, Sitong Mao, Rong Xiong, and Yue Wang. Let occ flow: Self-supervised 3d occupancy flow prediction. *The Conference on Robot Learning (CoRL)*, 2024.

[37] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022.

[38] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[39] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

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

[41] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12404–12411. IEEE, 2024.

[42] A Paszke. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[43] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506, 2020.

[44] Yunxiao Shi, Hong Cai, Amin Ansari, and Fatih Porikli. H3o: Hyper-efficient 3d occupancy prediction with heterogeneous supervision. *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[45] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[46] Stanislaw Szymanowicz, Chrisitian Rupprecht, and Andrea Vedaldi. Splatter image: Ultra-fast single-view 3d reconstruction. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10208–10217, 2024.

[47] Pin Tang, Zhongdao Wang, Guoqing Wang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma. Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15035–15044, 2024.

[48] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, et al. State of the art on neural rendering. In *Computer Graphics Forum*, volume 39, pages 701–727. Wiley Online Library, 2020.

[49] Ayush Tewari, Justus Thies, Ben Mildenhall, Pratul Srinivasan, Edgar Tretschk, Wang Yifan, Christoph Lassner, Vincent Sitzmann, Ricardo Martin-Brualla, Stephen Lombardi, et al. Advances in neural rendering. In *Computer Graphics Forum*, volume 41, pages 703–735. Wiley Online Library, 2022.

[50] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023.

[52] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[53] Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang Yan, Ke Wang, Jie Yang, Wei Liu, Qibin Hou, and Mingming Cheng. Opus: Occupancy prediction using a sparse set. In *Advances in Neural Information Processing Systems*, 2024.

[54] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3621–3631, 2023.

[55] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[56] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17850–17859, 2023.

[57] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022.

[58] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21729–21740, 2023.

[59] Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3d: Towards zero-shot metric 3d prediction from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9043–9053, 2023.

[60] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19447–19456, 2024.

[61] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023.

[62] Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *arXiv preprint arXiv:2312.09243*, 2023.

[63] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9433–9443, 2023.

[64] Benjin Zhu, Zhe Wang, and Hongsheng Li. nucraft: Crafting high resolution 3d semantic occupancy for unified 3d scene understanding. In *European Conference on Computer Vision*, pages 125–141. Springer, 2024.

Appendix

We provide more experiment analysis regarding various aspects of ODG.

A. Coarse-to-fine Prediction of Gaussian Parameters

We present details of the coarse-to-fine refinement used in ODG. For notation simplicity, we denote motion compensation, feature sampling, cross and self attention, along with dual query-attention presented in Figure 1 as a single layer $\hat{\mathcal{T}}_l$, rest of the notations are the same as defined in Section 3.2. To further enhance clarity, since coarse-to-fine refinement doesn't apply to bounding box attributes (i.e. $K_0 = 1$ throughout all the layers $\hat{\mathcal{T}}_l$), we have omitted them in Algorithm 1 below.

Input: images features $\mathbf{F}$; static Gaussian queries $\{\mathbf{G}_0^s, \mathbf{Q}_0^s\}$; dynamic Gaussian queries $\{\mathbf{G}_0^d, \mathbf{Q}_0^d\}$;

Output: refined Gaussian queries $\{\mathbf{G}_L^s, \mathbf{Q}_L^s\}$, class predictions $\mathbf{C}_L^s$; refined dynamic Gaussian queries $\{\mathbf{G}_L^d, \mathbf{Q}_L^d\}$, class predictions $\mathbf{C}_L^d$.

Function *ODGRefine(*$\mathbf{F}, \mathbf{G}_0^s, \mathbf{Q}_0^s, \mathbf{G}_0^d, \mathbf{Q}_0^d$*)*

$\mathbf{G}_{:\mu,0}^s \in \mathbb{R}^{S\times K_0\times 3} \leftarrow \mathcal{U}(0,1), K_0 = 1$

$\mathbf{G}_{:\mu,0}^d \in \mathbb{R}^{D\times K_0\times 3} \leftarrow \mathcal{U}(0,1), K_0 = 1$

// initialize Gaussian means with uniform distribution; rest of the Gaussian parameters left uninitialized

for $l \leftarrow 1$ **to** L **do**

static

$\mathbf{G}_{:\mu,l}^s, \mathbf{Q}_l^s, \mathbf{C}_l^s = \hat{\mathcal{T}}_l(\mathbf{G}_{:\mu,l-1}^s, \mathbf{Q}_{l-1}^s, \mathbf{C}_{l-1}^s)$

$\mathbf{G}_{\mu:,l}^s = \Phi(\mathbf{G}_{:\mu,l}^s, \mathbf{Q}_l^s)$ // rest of Gaussian params predicted by MLP

// $\mathbf{G}_{:\mu,l}^s \in \mathbb{R}^{S\times K_l\times 3}, \mathbf{G}_{\mu:,l}^s \in \mathbb{R}^{S\times K_l\times 10}, \mathbf{C}_l^s \in \mathbb{R}^{S\times K_l\times C}, K_{l-1} < K_l$ # coarse-to-fine

dynamic

$\mathbf{G}_{:\mu,l}^d, \mathbf{Q}_l^d, \mathbf{C}_l^d = \hat{\mathcal{T}}_l(\mathbf{G}_{:\mu,l-1}^d, \mathbf{Q}_{l-1}^d, \mathbf{C}_{l-1}^d)$

$\mathbf{G}_{\mu:,l}^d = \Phi(\mathbf{G}_{:\mu,l}^d, \mathbf{Q}_l^d)$ // rest of Gaussian params predicted by MLP

// $\mathbf{G}_{:\mu,l}^d \in \mathbb{R}^{D\times K_l\times 3}, \mathbf{G}_{\mu:,l}^d \in \mathbb{R}^{D\times K_l\times 10}, \mathbf{C}_l^d \in \mathbb{R}^{D\times K_l\times C}, K_{l-1} < K_l$

coarse-to-fine

end

return $\{\mathbf{G}_L^s, \mathbf{Q}_L^s, \mathbf{C}_L^s\}, \{\mathbf{G}_L^d, \mathbf{Q}_L^d, \mathbf{C}_L^d\}$

Algorithm 1: Coarse-to-fine refinement in ODG.

B. Runtime Efficiency

We profiled ODG-L at inference time with DeepSpeed [43]. Results are summarized in Table 6a below.

Table 6: Runtime analysis of ODG-L (FP32).

(a) Runtime of different components in ODG-L

Component	Runtime (ms)	Percentage
img_backbone (ReNet50)	33.82	16.58%
img_neck (FPN)	9.83	4.82%
ODG-L transformer	160.32	78.59%

(b) Runtime of different components in ODG-L transformer.

Component	Runtime (ms)	Percentage
Self-attention	70.77	44.14%
Point sampling	25.88	16.14%
Cross-attention	24.06	15.01%
FFN	8.55	5.33%
...	...	...

We can see that the transformer part takes up most of the inference cost in ODG-L. We provide further profiling results on ODG-L transformer in Table 6b above. Evidently query self-attention takes up almost half the runtime. One straight-away optimization can be replacing self-attention with efficient attention schemes such as linear attention [55] or state space models (SSMs) [18]. We plan to look into this as part of future work.

C. Evaluation of Prediction from Each Layer

To provide further insight of the coarse-to-fine scheme in ODG, we evaluate predictions of all 6 layers of ODG-T against ground-truth occupancy. The results are summarized in Table 7 below. In the beginning since the prediction is too coarse, the model behaves poorly in terms of mIoU and RayIoU. Then with our coarse-to-fine refinement, ODG starts gradually learning the scene and gives progressively better results, which validates the effectiveness of our coarse-to-fine refinement design.

Table 7: Evaluation of each layer's prediction in ODG-T.

ODG-T	# Predicted Gaussians	mIoU	RayIoU_{1m}	RayIoU_{2m}	RayIoU_{4m}	RayIoU
layer 1	600	0.35	1.9	2.8	3.8	2.8
layer 2	2400	3.17	9.5	13.5	16.3	13.0
layer 3	9600	18.68	25.9	32.6	36.6	31.7
layer 4	12800	26.48	29.9	37.9	42.2	36.6
layer 5	38400	31.02	30.5	39.9	43.3	37.9
layer 6	76800	32.82	31.1	40.5	43.8	38.5

D. Ablation on Supervision Signals

In Table 8, we show how ODG performs when there is only occupancy labels available. Compared to full supervision with bbox and rendering labels, ODG suffers a slight drop in terms of mIoU and rayIoU, but still delivers strong performance, which validates our design.

Table 8: Ablation on supervision signal.

Supervision	mIoU	rayIoU
Occupancy only	31.46	36.1
Occupancy & bbox & rendering supervision	32.82	38.5

E. Effect of Query Composition

We study the effect of utilizing different types of queries. The results are summarized in Table 9.

Table 9: Effect of query composition on model performance.

Method	# Queris	mIoU	RayIoU	FPS
OPUS-T [53]	600 (static)	33.2	38.4	22.4
OPUS-L [53]	4800 (static)	36.2	41.2	7.2
ODG-T	500 (static)+100 (dynamic)	35.5	39.2	20.1
ODG-L	4000 (static)+800 (dynamic)	38.2	42.3	4.9
ODG-T*	600 (dynamic)	34.8	38.9	17.6
ODG-L*	4800 (dynamic)	37.6	42.1	3.7

One can see that by taking into account object motion, ODG definitely outperforms baseline OPUS [53] which only considers ego motion. Interestingly, when we treat all our queries as dynamic (denoted as ODG-*), it still performs better than OPUS. We attribute this gain to the object detection task present in our system which effectively improves dynamic object predictions, and through rendering constraint we also learn better 3D geometry.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In the paper, we have provided sufficient details for reproducing our work, including detailed design and experiment setups.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will aim to release the code, which will be subject to our institution's review and approval. The datasets used in our experiments are publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In this paper, we have specified all the training and test details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: In our experiments, we use established benchmarks and training & testing protocols, which do not include reporting of error bars or statistical significance tests. In the literature, it is not a common practice to report error bars in the 3D occupancy prediction problem considered in this paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have provided information on the compute resources in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and our research conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the potential societal impacts of this work.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not introduce data or models that have a high risk for misuse and as such, poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: In the paper, we have cited the original papers that produced the data or models, and have mentioned/provided links to the licenses and terms of use. We have properly respected the licenses and terms of use of the data and model used in this work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.