EmbodiedOcc: Embodied 3D Occupancy Prediction for Vision-based Online Scene Understanding

Yuqi Wu Wenzhao Zheng✉ Sicheng Zuo Yuanhui Huang Jie Zhou Jiwen Lu
Department of Automation, Tsinghua University, China
wuyq24@mails.tsinghua.edu.cn; wenzhao.zheng@outlook.com

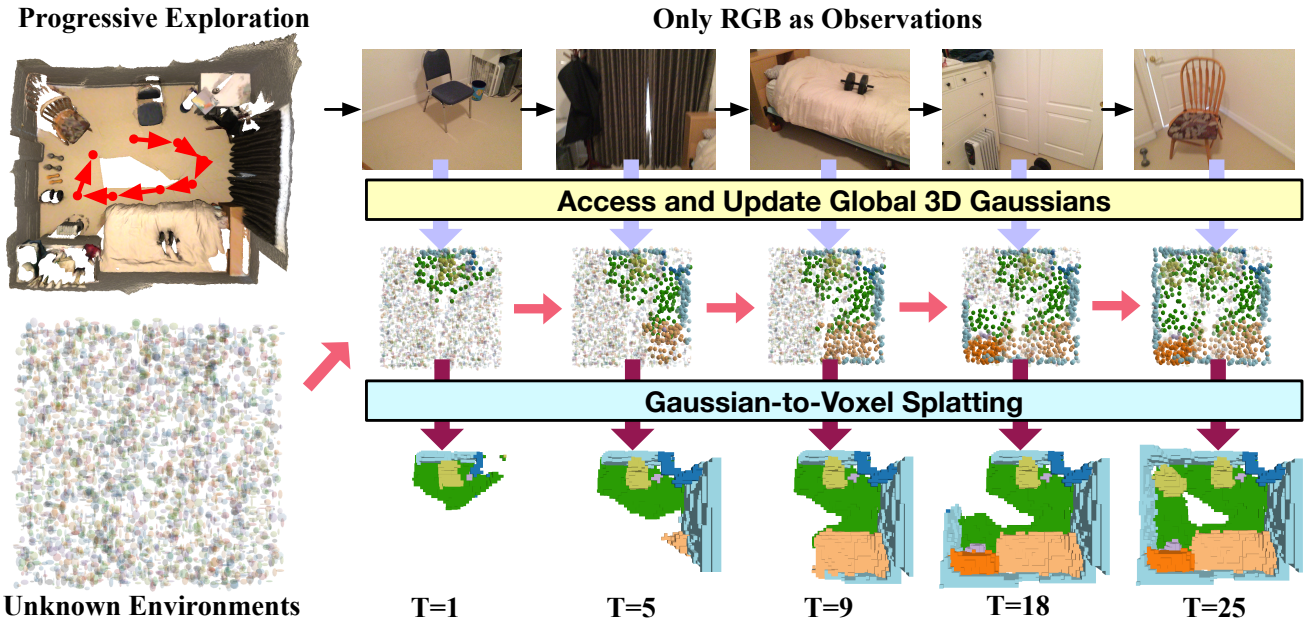

Figure 1. Given streaming monocular RGB inputs, our EmbodiedOcc conducts embodied occupancy prediction in an online manner for indoor scenes. Different from existing methods which focus on offline perception from monocular images, we focus on the scene-level occupancy prediction from embodied observations. We initialize the scene to be explored with uniform 3D semantic Gaussians and progressively update them based on new observations, similar to how humans explore unknown scenes.

Abstract

3D occupancy prediction provides a comprehensive description of the surrounding scenes and has become an essential task for 3D perception. Most existing methods focus on offline perception from one or a few views and cannot be applied to embodied agents that demand to gradually perceive the scene through progressive embodied exploration. In this paper, we formulate an embodied 3D occupancy prediction task to target this practical scenario and propose a Gaussian-based EmbodiedOcc framework to accomplish it. We initialize the global scene with uniform 3D semantic Gaussians and progressively update local regions observed by the embodied agent. For each update, we extract semantic and structural features from the observed image and efficiently incorporate them via deformable cross-attention to refine the regional Gaussians. Finally, we employ Gaussian-to-voxel splatting to obtain the global 3D occupancy from the updated 3D Gaussians. Our EmbodiedOcc assumes an unknown (i.e., uniformly distributed) environment and maintains an explicit global memory of it with 3D Gaussians. It gradually gains knowledge through the local refinement of regional Gaussians, which is consistent with how humans understand new scenes through embodied exploration. We reorganize an EmbodiedOcc-ScanNet benchmark based on local annotations to facilitate the evaluation of the embodied 3D occupancy prediction task. Our EmbodiedOcc outperforms existing methods by a large margin and accomplishes the embodied occupancy prediction with high accuracy and efficiency. Code: https://github.com/YkiWu/EmbodiedOcc.

1. Introduction

With the rapid development of embodied intelligence and active agents [14, 17, 32], 3D scene perception [30, 34, 41, 42] has become a crucial task in computer vision. Intelligent agents first perceive their surrounding environments and then make decisions based on the perception results.

✉ Corresponding author.

Due to the low costs of camera sensors, vision-based 3D occupancy prediction is gaining increasing popularity and produces a comprehensive understanding of both semantics and structures of the scene [2, 11, 13, 46, 56].

While vision-based 3D occupancy prediction has made significant progress in outdoor driving scenes [11, 13, 22, 40, 43, 45, 46, 52, 58, 59], the application to indoor scenarios is still challenging due to the diversity and complexity of indoor scenes. Most existing methods [2, 54, 56] still focus on local 3D occupancy prediction by integrating semantic and depth information extracted from the visual inputs. However, different from outdoor scenarios, it is important to obtain a global understanding of the room for indoor scenarios, as it usually requires multiple traversals for embodied agents. Also, it is more practical to progressively explore and update the global occupancy of the 3D scene in an online manner from embodied vision-based observations with different positions and perspectives.

To bridge this gap, we formulate a new embodied 3D occupancy prediction task to evaluate the ability to progressively explore an unknown scene using only visual inputs. We propose an EmbodiedOcc framework based on Gaussian memories to accomplish this task, considering the explicity and structural nature of 3D Gaussians. We initialize the global scene with uniform 3D semantic Gaussians and progressively update the Gaussians within the field of view observed by the agent. Throughout the exploration process, we maintain an explicit global memory of 3D Gaussians as the global understanding and derive the global 3D occupancy with Gaussian-to-voxel splatting [13]. Specifically, we propose a structure-aware local refinement module to update the relevant Gaussians within the current frustum. We employs a simple yet effective depth-aware branch to introduce explicit structural information for each Gaussian, ensuring the update of these Gaussians to better align with the global representation. During the continuous exploration, we read out Gaussians within the current frustum from the memory as inputs to the local module for refinement. We assign high confidence values for updated Gaussians and use them to reweight information from the memory and the current input. This ensures the consistency of the 3D representation during the fusion and update process. We reorganize an EmbodiedOcc-ScanNet benchmark for the embodied 3D occupancy prediction task based on the locally annotated Occ-ScanNet dataset [3, 47, 56]. Experiments show that our EmbodiedOcc outperforms existing methods by a large margin and accomplishes embodied occupancy prediction with high accuracy and efficiency.

2. Related Work

3D Occupancy Prediction. Benefiting from its compactness and versatility, 3D occupancy prediction based on multi-view images or additional 3D information [1, 11–13, 21, 37, 46] has gained great popularity over the last few years. MonoScene [2] was the first to derive 3D occupancy prediction from a single image, propelling the original 3D Semantic Scene Completion (SSC) [4, 8, 18, 19, 33, 37] into a more challenging stage with vision-only inputs and more universal scenarios (both indoor and outdoor scenes). Subsequent works [54, 56] further focused on addressing the depth ambiguity in this monocular setting. However, most of these efforts were confined to local and offline prediction. SCFusion [47] proposed an incremental framework based on RGB-D inputs. EmbodiedScan [28, 44] introduced an offline global prediction framework using multi-modal sequential inputs. Differently, the proposed embodied 3D occupancy prediction aims at online prediction from RGB-only inputs, which is more challenging and practical.

Online 3D Scene Perception. Accurate comprehension of 3D scenes is an indispensable capability for embodied agents, such as 3D occupancy prediction [2, 56] and object detection [16, 31, 42]. Most existing works on indoor 3D scene perception [9, 30, 41, 55] take pre-acquired and reconstructed 3D data as inputs and perceive the scene in an offline manner. To achieve online perception, Online3D [49] introduced an adapter-based model that equips mainstream offline frameworks with the competence to perform online scene perception, enabling the process of real-time RGB-D sequences. However, this framework still requires depth information as inputs and mainly targets point segmentation and 3D detection. Differently, we target online vision-based 3D occupancy prediction which can provide a more comprehensive understanding of the scene.

3D Gaussian Splatting. 3D Gaussian Splatting [15] uses 3D Gaussians to model a 3D scene and benefits from fast speed and high quality in the field of neural rendering. The physical characteristics of 3D Gaussians and the splat-based rasterization also motivated rapid advancements in research fields such as scene editing [10, 24, 29, 36], dynamic scenarios [7, 27, 38, 48, 53], and SLAM [5, 20, 50, 57]. GaussianFormer [13] pioneers the application of 3D Gaussians in outdoor 3D occupancy prediction and uses features from multi-view images to update 3D Gaussians, which can be converted into 3D occupancy prediction through a Gaussian-to-voxel splatting module. However, it is still unclear how to employ 3D Gaussians for online global indoor scene understanding from local observations. We achieve this by designing a Gaussian memory mechanism and progressively updating it with structure-aware interaction.

3. Proposed Approach

3.1. Embodied 3D Occupancy Prediction

Conventional methods in indoor scenarios for occupancy prediction accepted RGB-D as inputs to predict the semantic occupancy of a 3D scene which requires depth sensors.

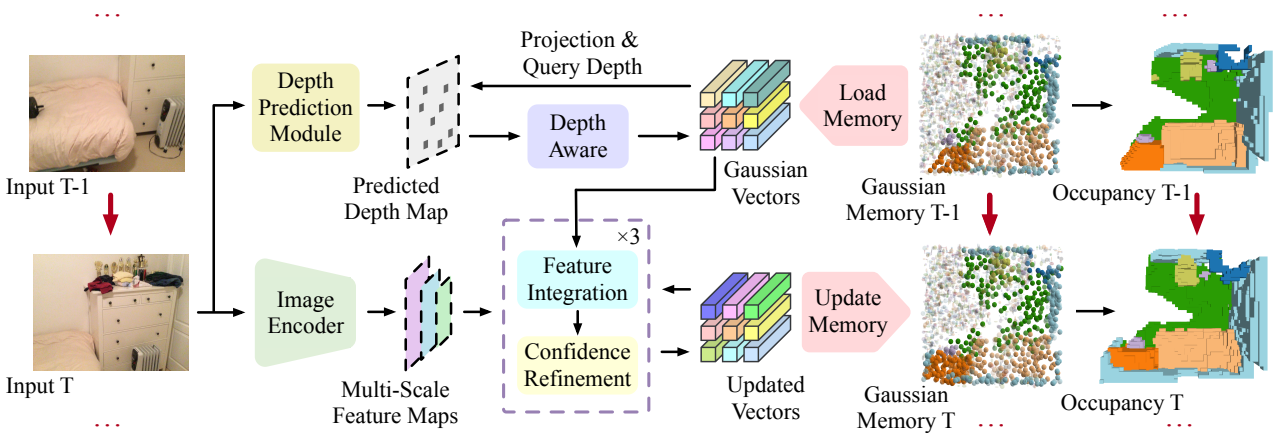

Figure 2. **Framework of our EmbodiedOcc for embodied 3D occupancy prediction.** We maintain an explicit global memory of 3D Gaussians during the exploration of the current scene. For each update, the Gaussians within the current frustum are taken from the memory and updated using semantic and structural features extracted from the monocular RGB input. Each Gaussian has a confidence value to integrate information from both the memory and the current input. Then we detach and put these updated Gaussians back into the memory. We can obtain the current 3D occupancy prediction using a Gaussian-to-voxel splatting module whenever we need.

However, we humans are capable of effortlessly processing the visual information from a single view to obtain 3D perception of our surroundings. Recent methods begin to consider endowing models with the same competence, which accept a monocular RGB image as input and derive a 3D occupancy prediction within the current frustum:

$$\mathbf{Y}_{mono} = \mathcal{F}_{mono}(I_{mono}), \tag{1}$$

where $\mathcal{F}_{mono}$ is the proposed monocular prediction model, $I_{mono} \in \mathbb{R}^{H\times W\times 3}$ and $\mathbf{Y}_{mono} \in \mathbb{R}^{X\times Y\times Z\times C}$ refer to the monocular RGB input and the obtained 3D occupancy prediction. X, Y, Z represent the dimensions of the local 3D scene and C represents the total number of semantics.

This is only the initial step towards practical scenarios. The essence of human intelligence is the capacity to analyze and respond immediately based on real-time perception of the surroundings. Correspondingly, superior embodied agents are anticipated to process egocentrically gathered real-time visual input to update the 3D occupancy prediction of the current scene. This capability facilitates the execution of downstream tasks based on real-time perception.

Motivated by this, we propose an embodied 3D occupancy prediction task in this paper. Let $\mathcal{X}_t = \{x_1, x_2, ..., x_t\}$ be an RGB sequence and the corresponding extrinsics collected by the embodied agent up to the present, where $x_t = (I_t, M_t)$, $I_t \in \mathbb{R}^{H\times W\times 3}$, $M_t \in \mathbb{R}^{3\times 4}$. It is worth noting that the variation in the subscripts merely represents the change in the position and perspective of the agent when exploring the current scene continuously. Different subscripts may correspond to similar positions and perspectives, indicating that the agent has returned to a previously explored location. In embodied occupancy prediction, re-exploration of the same area should maintain global consistency and even demonstrate improved performance, akin to we humans always possessing a more comprehensive understanding of sights that have been encountered repeatedly.

We formulate the function of an embodied occupancy prediction model as follows:

$$\mathbf{Y}_t = \mathcal{F}_{embodied}(\mathbf{Y}_{t-1}, x_t), \tag{2}$$

where $\mathcal{F}_{embodied}$ is the embodied prediction model, $\mathbf{Y}_t \in \mathbb{R}^{X_{room}\times Y_{room}\times Z_{room}\times C}$ refers to the current occupancy prediction of the whole scene ($\mathbf{Y}_0$ is the initialization). X_{room}, Y_{room}, Z_{room} denote the scene dimensions.

3.2. Local Refinement Module

Different from conventional methods that conducted feature integration in a voxelized space, we use a set of 3D semantic Gaussians to represent an indoor scene [13]. In this subsection, we will first explain our local refinement module, which extracts semantic and structural features from the monocular input and integrates them to update the Gaussian-based representation of the current frustum.

Initialization. We first initialize a set of semantic Gaussians to represent the current frustum. Each semantic Gaussian $\mathbf{G}$ is represented by a vector comprising mean $\mathbf{m} \in \mathbb{R}^3$, scale $\mathbf{s} \in \mathbb{R}^3$, rotation quaternion $\mathbf{r} \in \mathbb{R}^4$, opacity $\mathbf{o} \in \mathbb{R}$, and semantic logits $\mathbf{c} \in \mathbb{R}^C$ (C denotes the total number of semantic categories). We use an embedding layer to lift each Gaussian vector $\mathbf{G}$ to its corresponding high-dimensional feature vectors $\mathbf{Q}$, and derive $\mathcal{Q} = \{\mathbf{Q}_i \in \mathbb{R}^m, i = 1, ..., N\}$, where m is the dimension of $\mathbf{Q}_i$ and N is the total number of the Gaussians.

Depth-Aware Branch. Due to the variable scales and tight arrangements of indoor objects, depth ambiguity has always been one of the core challenges limiting the performance of indoor occupancy prediction models in monocular settings. Previous work has consistently focused on how to

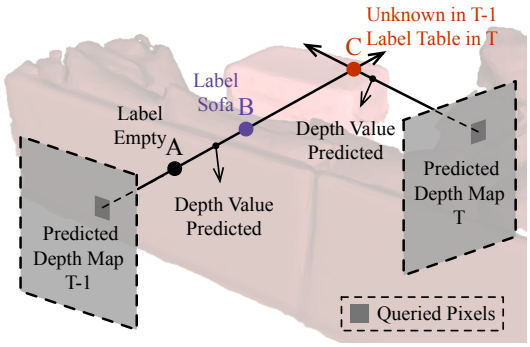

Figure 3. **Motivation of the depth-aware branch.** Along a specific ray, Gaussians distributed in front of the true depth point are likely to model the empty semantic (A). Gaussians distributed behind the true depth point closely are likely to model valid semantics (B). Gaussians that are distributed behind the true depth point but are too far away require more information to guide their updates (C). During the embodied exploration, the subsequent frames can make up for this lack of information in the current frame.

better extract and utilize depth information from the input image. We design a depth-aware branch to provide more accurate and effective guidance for the update of 3D semantic Gaussians in our local refinement module.

We first use a depth prediction network to obtain a relatively accurate depth map D_{metric} from input I_{mono}. A naive approach can explicitly utilize this depth information when initializing the Gaussians, e.g., we can randomly sample some points from the pseudo point cloud recovered from the depth map and use these coordinates to initialize the means of some Gaussians. Although providing direct hints for the means of some Gaussians, this cannot exploit the potential of the depth information. We design a simple yet effective depth-aware layer to accomplish this. We still uniformly initialize a number of Gaussians within the current frustum. For each Gaussian, we project its mean $\mathbf{m}$ into the pixel coordinate system through the intrinsics $K_{mono} \in \mathbb{R}^{3\times3}$ and obtain the depth value d. The sampled depth value d, along with the z-component z of the Gaussian mean in the camera coordinate system, are fed into the depth-aware layer, which is a multi-layer perceptron (MLP) that outputs the depth-aware feature $\mathbf{Q}_{depth}$ for this Gaussian. Then we add the depth-aware feature to the original feature vector $\mathbf{Q}$, injecting additional information into the subsequent feature integration. In this way, depth information not only affects the means of the Gaussians but also promotes the update of other properties:

$$\begin{aligned} \mathbf{Q}_{depth} &= \mathcal{M}_{depthaware}((D_{metric}(u,v),z), \\ \hat{\mathcal{Q}} &= \{\hat{\mathbf{Q}}_i, i=1,...,N|\hat{\mathbf{Q}}_i = \mathbf{Q}_i + \mathbf{Q}_i^{depth}\}, \end{aligned} \tag{3}$$

where $\mathcal{M}_{depthaware}$ is the depth-aware layer, (u,v) are pixel coordinates of each Gaussian. We illustrate our depth-aware branch in Figure 3.

Feature Integration and Gaussian Refinement. Feature integration in our local refinement module includes the interactions among Gaussians as well as the interactions between image features and Gaussians. We voxelize the Gaussian centers and conduct 3D sparse convolution on the generated grid to allow interactions among Gaussian vectors $\hat{\mathcal{Q}}$. We project the Gaussian centers onto the image feature map and use the deformable attention function to integrate the queried features and the Gaussian vectors $\hat{\mathcal{Q}}$. After the prior feature integration, these feature vectors with aggregated information will be used to obtain the update amounts $\Delta\mathbf{G} = (\Delta\mathbf{m}, \Delta\mathbf{s}, \Delta\mathbf{r}, \Delta\mathbf{o}, \Delta\mathbf{c})$ of each Gaussian. We use the update amounts $\Delta\mathbf{G}$ to refine the Gaussian properties:

$$\mathbf{G}_{new} = (\Delta\mathbf{m}+\mathbf{m}, \Delta\mathbf{s}+\mathbf{s}, \Delta\mathbf{r}\otimes\mathbf{r}, \Delta\mathbf{o}+\mathbf{o}, \Delta\mathbf{c}+\mathbf{c}), \tag{4}$$

where $\otimes$ refers to the special composition of quaternions.

We conduct the feature integration and the refinement of Gaussians multiple times. After the final refinement, we use a Gaussian-to-voxel splatting module [13] to obtain the final occupancy within the frustum.

3.3. Gaussian Memory Updated Online

To explore unknown scenes, we humans continuously update the objects within the scene and their relationships to gradually construct a global scene memory. When revisiting this scene for further exploration, we use the visual information to refine this memory. Inspired by this, we design an online framework (shown in Figure 2) and maintain a Gaussian memory for global understanding.

Memory Initialization. Our local refinement module initializes and updates Gaussians in the camera coordinate system, as the extrinsics in indoor scenarios are constantly changing, which will pose additional difficulties for our local module. But in the final embodied framework, we initialize the entire scene with uniform Gaussians in the world coordinate system. For a novel scene to be explored, we have: $\mathcal{G}_{room} = \{(\mathbf{G}_i, \gamma_i), i=1,...,N|\mathbf{G}_i = (\mathbf{m}_i, \mathbf{s}_i, \mathbf{r}_i, \mathbf{o}_i, \mathbf{c}_i), \gamma_i = 0,1\}$, where N refers to the number of Gaussians to initialize this scene, $\mathbf{m}_i$ and $\mathbf{r}_i$ are the means and rotation quaternions of these Gaussians in the world coordinate system ($\mathbf{s}_i$, $\mathbf{o}_i$ and $\mathbf{c}_i$ maintain consistency between the world and camera coordinate systems). We introduce an additional tag γ for all the Gaussians in the memory. When initializing a novel scene, tags of these Gaussians are set to 0. Every time we put some updated Gaussians back into the memory, their tags are set to 1.

Memory Update. At the current step t, our embodied occupancy prediction framework receives a posed visual input $x_t = (I_t, M_t)$ to perform the update. During the current update, we use a mask from coordinate system transformation to get all Gaussians $\mathcal{G}_t$ within the current frustum from the memory. These Gaussians will interact and be refined using a tailored confidence refinement module. Then we detach these Gaussians and put them back into the memory.

Confidence Refinement. Apart from the initial update for each scene which is akin to the local refinement, sub-

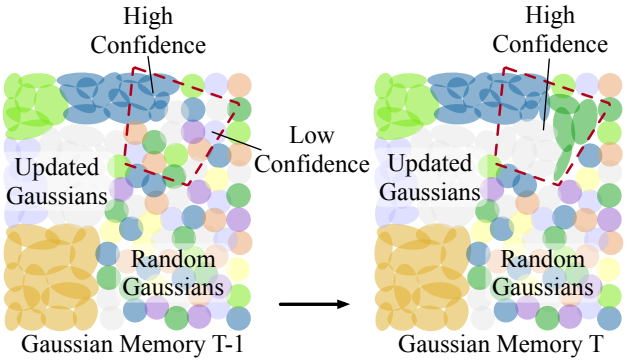

Figure 4. **Illustration of our Gaussian memory.** During each update, the Gaussians within the current frustum are taken from the memory. Confidence values of those well-updated Gaussians are used to integrate information from both the memory and the current input. Then we put these Gaussians back into the memory.

sequent exploration involves the update of Gaussians from the Gaussian memory, among which some have been well-updated by previous frames (if we can derive an acceptable local occupancy prediction from these Gaussians, we believe that they have been "well-updated") and some still remain random. It is unreasonable to update these Gaussians equally. For those Gaussians deemed well-updated, we only need to refine them slightly based on the semantic and structural features extracted from the current image, which is exactly the essence of maintaining the Gaussian memory. As for those random Gaussians that have never been updated, we can directly update them with a fresh perspective.

To elaborate, we generate a confidence value θ for each Gaussian taken from the memory. For those having been previously updated ($\gamma = 1$), we set their confidence values to a certain value between 0 and 1, which means we will integrate information from both the memory and the newly observed image to update these Gaussians. For those that have never been updated, we set their confidence values to 0 and follow the same refinement module in Sec. 3.2:

$$\begin{aligned} \Delta\mathbf{G}_{online} &= (1-\theta)\Delta\mathbf{G}, \\ \mathbf{G}_{after} &= \Delta\mathbf{G}_{online} \oplus \mathbf{G}_{before}, \end{aligned} \tag{5}$$

where we use $\oplus$ to represent the composition of rotation quaternions and the add operation of other parts. Figure 4 illustrates how we maintain the Gaussian memory.

Stopping Mechanism. We propose a simple stopping mechanism to consider a room as having been effectively explored. At the step t, we first calculate a confidence ratio α to measure the exploration of the current room:

$$\alpha = \textstyle\sum_{i=0}^{N} \mathbb{I}_{\gamma_i=1}/N, \tag{6}$$

where $\mathbb{I}_{\gamma_i=1}$ takes the value of 1 if $\gamma_i = 1$. If α exceeds a certain threshold we set before, the model can decide to enter an adjacent room to begin a new exploration or stay here to get a better perception of the current room.

3.4. EmbodiedOcc: An Embodied Framework

We present the training framework of our EmbodiedOcc model for indoor embodied occupancy prediction. During the whole prediction process, we use the current monocular input to update our Gaussian memory in real time, which can be easily converted into 3D occupancy prediction.

We first train our local refinement module using the focal loss L_{focal}, the lovasz-softmax loss L_{lov}, the scene-class affinity loss L^{geo}_{scal} and L^{sem}_{scal} following RetinaNet [23], TPVFormer [11] and MonoScene [2]. We use monocular occupancy within the frustum $\mathbf{Y}^{fov}_{mono}$ and the corresponding ground truth $\mathbf{Y}^{fov}_{gt}$ to compute the loss:

$$\begin{aligned} \mathcal{L} = \lambda_1 \mathcal{L}_{focal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) + \mathcal{L}_{lov}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) \\ + \mathcal{L}^{geo}_{scal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) + \mathcal{L}^{sem}_{scal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}), \end{aligned} \tag{7}$$

where λ_1 is a balance factor.

We then use the trained local module to train our EmbodiedOcc. For efficient training, we initialize the Gaussian memory before the first update and compute the current loss following the equation 7 after each update. To ensure consistency, the local occupancy ground truth used here is obtained from the occupancy of the whole scene. After a certain number of updates, we re-initialize the memory and come to the next scene. Trained with such a pipeline, our EmbodiedOcc can effectively perform the embodied occupancy prediction task while ensuring consistency within the same scene. We expect that our EmbodiedOcc can have an improving prediction with continuous exploration rather than undermining previous predictions when encountering parts that have been explored before. Therefore, we conduct some tailored tests to validate the capability of our model.

4. Experiments

4.1. EmbodiedOcc-ScanNet Benchmark

Task Descriptions. We conducted two tasks to evaluate our EmbodiedOcc framework: local occupancy prediction and embodied occupancy prediction. Local occupancy prediction shares the same setting with previous works, which accept monocular image as input and obtain the occupancy within the current frustum. Embodied occupancy prediction accepts real-time visual inputs continuously and updates the occupancy of the current scene online. The visual input at a certain step t during embodied occupancy prediction is still monocular, which is a more challenging setting compared with multi-view input or input with 3D information.

Datasets. We trained and evaluated our local refinement module on the Occ-ScanNet dataset [56], which provides frames in $60 \times 60 \times 36$ voxel grids (a $4.8m \times 4.8m \times 2.88m$ box in front of the camera). These frames are labeled with 12 semantics, including 11 for valid semantics (ceiling, floor, wall, window, chair, bed, sofa, table, tvs, furniture, objects) and 1 for empty space.

Table 1. **Local Prediction Performance on the Occ-ScanNet dataset.**

Method	Input	IoU	ceiling	floor	wall	window	chair	bed	sofa	table	tvs	furniture	objects	mIoU
TPVFormer [11]	x^{rgb}	33.39	6.96	32.97	14.41	9.10	24.01	41.49	45.44	28.61	10.66	35.37	25.31	24.94
GaussianFormer [13]	x^{rgb}	40.91	20.70	42.00	23.40	17.40	27.0	44.30	44.80	32.70	15.30	36.70	25.00	29.93
MonoScene [2]	x^{rgb}	41.60	15.17	44.71	22.41	12.55	26.11	27.03	35.91	28.32	6.57	32.16	19.84	24.62
ISO [56]	x^{rgb}	42.16	19.88	41.88	22.37	16.98	29.09	42.43	42.00	29.60	10.62	36.36	24.61	28.71
Surroundocc [46]	x^{rgb}	42.52	18.90	49.30	24.80	18.00	26.80	42.00	44.10	32.90	18.60	36.80	26.90	30.83
Ours	x^{rgb}	**53.55**	**39.60**	**50.40**	**41.40**	**31.70**	**40.90**	**55.00**	**61.40**	**44.00**	**36.10**	**53.90**	**42.20**	**45.15**

Table 2. **Embodied Prediction Performance on the EmbodiedOcc-ScanNet dataset.**

Method	Dataset	IoU	ceiling	floor	wall	window	chair	bed	sofa	table	tvs	furniture	objects	mIoU
TPVFormer [11]	EmbodiedOcc	35.88	1.62	30.54	12.03	13.22	35.47	51.39	49.79	25.63	3.60	43.15	16.23	25.70
SurroundOcc [46]	EmbodiedOcc	37.04	12.70	31.80	22.50	22.00	29.90	44.70	36.50	24.60	11.50	34.40	18.20	26.27
GaussianFormer [13]	EmbodiedOcc	38.02	17.00	33.60	21.50	21.70	29.40	47.80	37.10	24.30	15.50	36.20	16.80	27.36
SplicingOcc	EmbodiedOcc	49.01	**31.60**	38.80	35.50	36.30	47.10	54.50	57.20	34.40	32.50	51.20	29.10	40.74
EmbodiedOcc	EmbodiedOcc	**51.52**	22.70	**44.60**	**37.40**	**38.00**	**50.10**	**56.70**	**59.70**	**35.40**	**38.40**	**52.00**	**32.90**	**42.53**

Based on this dataset, we reorganized an EmbodiedOcc-ScanNet dataset to train and evaluate our EmbodiedOcc framework [35, 56]. Our EmbodiedOcc-ScanNet comprises 537/137 scenes in the train/val splits. Each scene consists of 30 posed frames with their corresponding local occupancies. The resolutions of global occupancy of each scene are calculated by $l_x \times l_y \times l_z/(0.08m)^3$, where $l_x \times l_y \times l_z$ is the range of this scene in the world coordinate system. In addition, we associate grid points that can be projected onto the camera plane for each frame as the global mask of this frame. By splicing the global mask of all processed frames, we can easily obtain the occupancy ground truth of the explored part in the current scene.

Apart from Occ-ScanNet and EmbodiedOcc-ScanNet datasets in the original scale, we sampled a small set from the EmbodiedOcc-ScanNet dataset as the EmbodiedOcc-ScanNet-mini dataset which comprises 64/16 scenes in the train/val splits. We sampled from the Occ-ScanNet dataset accordingly and obtained an Occ-ScanNet-mini2 dataset, which comprises 5504/2376 frames in the train/val splits. We conducted the local occupancy prediction task on the Occ-ScanNet and Occ-ScanNet-mini2 datasets and conducted the embodied prediction task on the EmbodiedOcc-ScanNet and EmbodiedOcc-ScanNet-mini datasets.

Evaluation Metrics. We use mIoU and IoU as the evaluation metrics. For local occupancy prediction, we calculate the mIoU and IoU using the occupancy within the current frustum (same with the evaluation in ISO [56]). For embodied occupancy prediction, we calculate the mIoU and IoU using the global occupancy of the current scene. It is worth mentioning that the global occupancy used here is the union of the frustums corresponding to 30 frames of each scene, which represents the region that has been explored.

4.2. Implementation Details

Local Refinement Module. Following existing works [13, 56], we use a pre-trained EfficientNet-B7 [39] to initialize the image encoder in our local module. The depth prediction network used in the depth-aware branch is a fine-tuned DepthAnything-V2 model [51] that remains frozen during the training, and the depth-aware layer is a 3-layer MLP. The resolutions of the monocular input are set to 480×640 and the number of Gaussians used to conduct the local prediction is 16200. We utilize the AdamW [26] optimizer with a weight decay of 0.01. The learning rate warms up in the first 1000 iterations to a maximum value of 2e-4 and decreases according to a cosine schedule [25]. We train our local refinement module for 10 epochs using 8 NVIDIA GeForce RTX 4090 GPUs on the Occ-ScanNet dataset and 20 epochs on the Occ-ScanNet-mini2 dataset.

EmbodiedOcc Framework. We initialize the Gaussians with a 0.16 m interval to represent a novel scene. For each update, the confidence value θ of well-updated Gaussians is set to 0 in the first two refinement layers (frozen) and 0.5 in the final refinement layer. We train our EmbodiedOcc for 5 epochs using 8 NVIDIA GeForce RTX 4090 GPUs on the EmbodiedOcc-ScanNet dataset and 20 epochs using 4 NVIDIA GeForce RTX 4090 GPUs on the EmbodiedOcc-ScanNet-mini dataset. The maximum value of the learning rate is set to 2e-4 using 8 GPUs and 1e-4 using 4 GPUs. The other settings remain the same with the training of the local refinement module.

Table 3. **Look-Back Prediction *vs* First-Time Prediction.** For $K = k$, we simply select $0, 1, ..., k-1$th frames to evaluate our EmbodiedOcc framework and the occupancy ground truth used here is the union of the k frustums. K was set to 3/5/8.

Mode	K	Frame List	IoU	mIoU
First-Time	3	[0, 1, 2]	49.39	39.32
Look-Back	3	[0, 1, 2, 1, 0]	**49.52**	**40.09**
First-Time	5	[0, ..., 4]	50.13	40.03
Look-Back	5	[0, ..., 3, 4, 3, ..., 0]	**50.64**	**40.98**
First-Time	8	[0, ..., 7]	50.94	40.86
Look-Back	8	[0, ..., 6, 7, 6, ..., 0]	**51.14**	**41.17**

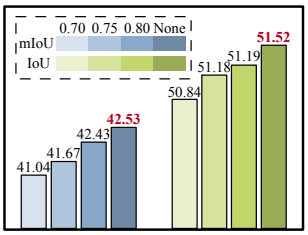

Figure 5. **Performance with different stopping ratios.**

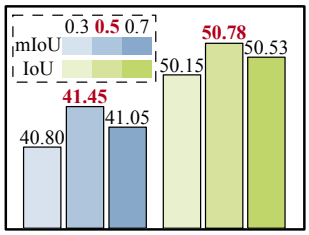

Figure 6. **Ablation study of the confidence refinement.**

4.3. Main Results

Local Occupancy Prediction. We evaluated our local refinement module on the Occ-ScanNet dataset [56]. As shown in Table 1, the results indicate that our local refinement module outperforms ISO [56]. We also implemented several state-of-the-art driving scene methods [11, 13, 46] on this benchmark and our local refinement module outperforms them by a large margin. This is because they mainly focus on the coarse layout (e.g., positions of objects) while indoor scenes require modeling of the fine-grained structure (e.g., shapes of objects).

Embodied Occupancy Prediction. We assessed the occupancy prediction for the entire scene after processing 30 frames, and the ground truth for calculating IoU and mIoU is the union of the frustums. We spliced the local occupancy obtained from our local module to serve as the main baseline (referred to as SplicingOcc), as our local module has achieved the best local performance to date. It can be observed in Table 2 that our EmbodiedOcc exhibits superior prediction of the scene, which is achieved through the integration of different views. We also compared our EmbodiedOcc with the driving scene methods mentioned before (we obtained their embodied results by voting from different local predictions). Their poor results are due to ignoring the continuity of the observations without a global memory.

4.4. Experimental Analysis

Effect of Continuous Online Updating. We expect EmbodiedOcc to have better performance when encountering parts that have been explored before, and thus, we designed a Look-Back evaluation on the EmbodiedOcc-ScanNet dataset. Specifically, after processing K frames, we direct the model to reprocess the last $K-1$ frames. By comparing this Look-Back result with the First-Time prediction, we verified that our EmbodiedOcc has met our expectations as shown in Table 3.

Table 4. **Analysis of the model design.**

Method	Gaussian	Structure	Memory	Local Prediction IoU	Local Prediction mIoU	Embodied Prediction IoU	Embodied Prediction mIoU
EmbodiedOcc-Voxel	×	✓	✓	47.50	38.12	37.53	26.99
EmbodiedOcc w/o memory	✓	✓	×	**53.55**	**45.15**	49.01	40.74
EmbodiedOcc	✓	✓	✓	**53.55**	**45.15**	**51.52**	**42.53**

Table 5. **Analysis of the depth-aware branch.**

Branch Type	Depth Estimation Module	Local Prediction IoU	Local Prediction mIoU	Embodied Prediction IoU	Embodied Prediction mIoU
Depth-aware branch	DepthAnything-V2	**53.93**	**46.20**	**50.78**	**41.45**
No-depth branch	/	48.15	40.07	37.52	30.73
Naive-depth branch	DepthAnything-V2	50.32	42.73	/	/
Depth-aware branch	IndoorDepth	51.24	43.87	46.42	37.78

Table 6. **Analysis of the Gaussian parameters.**

Gaussian Number (In local box)	Gaussian Scale Min(m)	Gaussian Scale Max(m)	Gaussian Interval(m) (In global scene)	Local Prediction IoU	Local Prediction mIoU	Embodied Prediction IoU	Embodied Prediction mIoU
16200	0.01	0.08	(0.16, 0.16, 0.16)	**53.93**	**46.20**	**50.78**	**41.45**
8100	0.01	0.08	(0.20, 0.20, 0.20)	50.47	42.82	46.24	37.99
16200	0.01	0.20	(0.16, 0.16, 0.16)	51.57	43.74	48.09	38.40

Table 7. **Runtime decomposition.**

Scene level (ms)	Scene init.	6.626	Occ head	39.635
Frame level (ms)	Load memory	0.973	Depth aware	1.816
	Img backbone	61.478	GS Encoder	14.761
	Depthanything	34.687	Update memory	0.474

Effect of the Stopping Mechanism. We use Figure 5 to show the effectiveness of our stopping mechanism. The ground truth used here for calculating IoU and mIoU is the union occupancy of the 30 frustums in a global scene. We observed that using a larger threshold results in more observations and better performance.

Analysis of the Confidence Refinement. During each update, local Gaussians are refined through three refinement layers. For Gaussians that have been updated before, we froze the first two layers and updated them in the last refinement layer when training our EmbodiedOcc. Figure 6 on the Occ-ScanNet-mini2 and the EmbodiedOcc-ScanNet-mini datasets shows the impact of different confidence values (determines the coefficient of each $\Delta\mathbf{G}$). We observe that moderate updates to those previously processed Gaussians yield the best embodied occupancy prediction.

Analysis of the Model Design. The essence of our EmbodiedOcc is the explicit Gaussian memory. We adopt object-centric Gaussians instead of grid-based voxels since Gaussians are more flexible for local-global interaction. We implemented a voxel version of our EmbodiedOcc and evaluated it on our benchmark. As shown in Table 4, the satisfactory local yet poor embodied performance of EmbodiedOcc in the voxel version verified our conclusion. Results in Table 4 were evaluated on the Occ-ScanNet and EmbodiedOcc-ScanNet datasets.

Analysis of the Depth-Aware Branch. We analyze the effect of our depth-aware branch in Table 5 using the

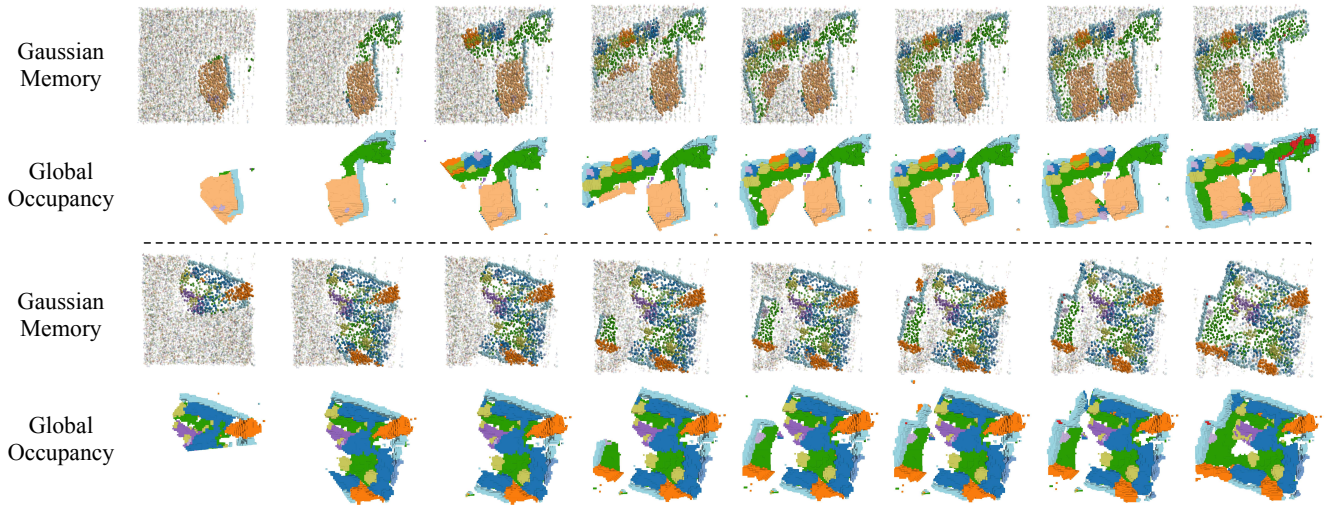

Figure 7. **Visualization of the embodied occupancy prediction.** We visualize the update of Gaussian memory and corresponding global occupancy. As the Gaussians transition from random to ordered, the occupancy of the current scene becomes more accurate and complete.

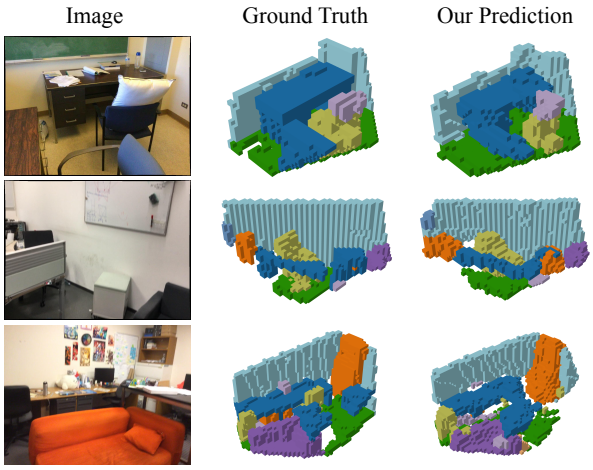

Figure 8. **Visualization of local occupancy prediction.**

Occ-ScanNet-mini2 and the EmbodiedOcc-ScanNet-mini datasets. We find that depth information will significantly benefit the local and embodied occupancy prediction. As shown in the second row, without the assistance of depth information, the performance of embodied occupancy prediction drops sharply. This indicates that the update of Gaussians within the current frustum may corrupt previous predictions without the guidance of depth information. The results in the third row suggest that the depth-aware branch we employ is more reasonable compared to the naive method of directly initializing a portion of Gaussians with the pseudo point cloud recovered from the predicted depth map, the latter also poses difficulties for the initialization of global Gaussians so we do not provide the embodied results. Besides, we replaced DepthAnything-V2 with IndoorDepth [6] in the last row to prove that our depth-aware branch does not rely on a specific depth prediction network.

Analysis of the Gaussian Parameters. We analyze the effect of different Gaussian parameters in Table 6 using the Occ-ScanNet-mini2 and the EmbodiedOcc-ScanNet-mini datasets. We see that decreasing the number or increasing the scale of the Gaussians can lead to a decrease in performance during local and embodied occupancy prediction. This is closely related to the physical properties of Gaussians. Gaussians initialized too sparse may lead to holes in occupancy prediction, while Gaussians with too large scale will overlap and influence each other which is also detrimental to the correct prediction of occupancy.

Runtime Analysis. We present in Table 7 a runtime analysis on scene 0687-00 from the EmbodiedOcc-ScanNet dataset. The runtime decomposition details show that our method is efficient while the main bottleneck is the image and depth backbones, suggesting that the overall runtime of our EmbodiedOcc can be further reduced.

Visualizations. Figure 7 and 8 visualize the global and local predictions, respectively. Our model demonstrates reasonable local perception ability and further achieves good online prediction with the Gaussian memory. Due to space limitations, we will use a more diverse set of samples to further show the visual effect of our EmbodiedOcc in the supplementary material.

5. Conclusion

In this paper, we have presented an embodied 3D occupancy prediction task and proposed a Gaussian-based EmbodiedOcc framework accordingly. Our EmbodiedOcc maintains an explicit Gaussian memory of the current scene and updates this memory during the exploration of this scene. Both quantitative and visualization results have shown that our EmbodiedOcc outperforms existing methods in terms of local occupancy prediction and accomplishes the embodied occupancy prediction task with high accuracy and strong expandability. We believe that our EmbodiedOcc paves the way for enabling active agents to conduct accurate and flexible embodied occupancy prediction.

Acknowledgements

We would like to thank Tianyu Hu for her valuable assistance with the experiments. This work was supported in part by the National Natural Science Foundation of China under Grant 62125603, Grant 62321005, and Grant 62336004, and in part by the Beijing Natural Science Foundation under Grant No. L247009.

References

[1] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *CVPR*, pages 324–333, 2021. 2

[2] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022. 2, 5, 6

[3] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, pages 5828–5839, 2017. 2

[4] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *CVPR*, pages 4578–4587, 2018. 2

[5] Tianchen Deng, Yaohui Chen, Leyan Zhang, Jianfei Yang, Shenghai Yuan, Jiuming Liu, Danwei Wang, Hesheng Wang, and Weidong Chen. Compact 3d gaussian splatting for dense visual slam. *arXiv preprint arXiv:2403.11247*, 2024. 2

[6] Chao Fan, Zhenyu Yin, Yue Li, and Feiqing Zhang. Deeper into self-supervised monocular indoor depth estimation. *arXiv preprint arXiv:2312.01283*, 2023. 8

[7] Quankai Gao, Qiangeng Xu, Zhe Cao, Ben Mildenhall, Wenchao Ma, Le Chen, Danhang Tang, and Ulrich Neumann. Gaussianflow: Splatting gaussian dynamics for 4d content creation. *arXiv preprint arXiv:2403.12365*, 2024. 2

[8] Martin Garbade, Yueh-Tung Chen, Johann Sawatzky, and Juergen Gall. Two stream 3d semantic scene completion. In *CVPRW*, pages 0–0, 2019. 2

[9] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *CVPR*, pages 9224–9232, 2018. 2

[10] Antoine Guédon and Vincent Lepetit. Gaussian frosting: Editable complex radiance fields with real-time rendering. *arXiv preprint arXiv:2403.14554*, 2024. 2

[11] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *CVPR*, pages 9223–9232, 2023. 2, 5, 6, 7

[12] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *CVPR*, pages 19946–19956, 2024.

[13] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In *ECCV*, pages 376–393, 2025. 2, 3, 4, 6, 7

[14] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. In *ICPR*, pages 4065–4071. IEEE, 2022. 1

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *TOG*, 42(4), 2023. 2

[16] Maksim Kolodiazhnyi, Anna Vorontsova, Matvey Skripkin, Danila Rukhovich, and Anton Konushin. Unidet3d: Multi-dataset indoor 3d object detection. In *AAAI*, 2025. 2

[17] Xiaohan Lei, Min Wang, Wengang Zhou, Li Li, and Houqiang Li. Instance-aware exploration-verification-exploitation for instance imagegoal navigation. In *CVPR*, pages 16329–16339, 2024. 1

[18] Jie Li, Yu Liu, Dong Gong, Qinfeng Shi, Xia Yuan, Chunxia Zhao, and Ian Reid. Rgbd based dimensional decomposition residual network for 3d semantic scene completion. In *CVPR*, pages 7693–7702, 2019. 2

[19] Jie Li, Yu Liu, Xia Yuan, Chunxia Zhao, Roland Siegwart, Ian Reid, and Cesar Cadena. Depth based semantic scene completion with position importance aware loss. *RAL*, 5(1): 219–226, 2019. 2

[20] Mingrui Li, Shuhong Liu, Heng Zhou, Guohao Zhu, Na Cheng, Tianchen Deng, and Hongyu Wang. Sgs-slam: Semantic gaussian splatting for neural dense slam. In *ECCV*, pages 163–179, 2025. 2

[21] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *CVPR*, 2023. 2

[22] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023. 2

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 5

[24] Zhiheng Liu, Hao Ouyang, Qiuyu Wang, Ka Leong Cheng, Jie Xiao, Kai Zhu, Nan Xue, Yu Liu, Yujun Shen, and Yang Cao. Infusion: Inpainting 3d gaussians via learning depth completion from diffusion prior. *arXiv preprint arXiv:2404.11613*, 2024. 2

[25] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 6

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[27] Zhicheng Lu, Xiang Guo, Le Hui, Tianrui Chen, Min Yang, Xiao Tang, Feng Zhu, and Yuchao Dai. 3d geometry-aware deformable gaussian splatting for dynamic view synthesis. In *CVPR*, pages 8900–8910, 2024. 2

[28] Ruiyuan Lyu, Tai Wang, Jingli Lin, Shuai Yang, Xiaohan Mao, Yilun Chen, Runsen Xu, Haifeng Huang, Chenming Zhu, Dahua Lin, et al. Mmscan: A multi-modal 3d scene dataset with hierarchical grounded language annotations. *arXiv preprint arXiv:2406.09401*, 2024. 2

[29] Francesco Palandra, Andrea Sanchietti, Daniele Baieri, and Emanuele Rodolà. Gsedit: Efficient text-guided editing of 3d objects via gaussian splatting. *arXiv preprint arXiv:2403.05154*, 2024. 2

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 1, 2

[31] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *ICCV*, pages 9277–9286, 2019. 2

[32] Sonia Raychaudhuri, Tommaso Campari, Unnat Jain, Manolis Savva, and Angel X Chang. Mopa: Modular object navigation with pointgoal agents. In *WACV*, pages 5763–5773, 2024. 1

[33] Christoph B Rist, David Emmerichs, Markus Enzweiler, and Dariu M Gavrila. Semantic scene completion using local deep implicit functions on lidar data. *TPAMI*, 44(10):7205–7218, 2021. 2

[34] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: Fully convolutional anchor-free 3d object detection. In *ECCV*, pages 477–493, 2022. 1

[35] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. 6

[36] Myrna C Silva, Mahtab Dahaghin, Matteo Toso, and Alessio Del Bue. Contrastive gaussian clustering: Weakly supervised 3d scene segmentation. *arXiv preprint arXiv:2404.12784*, 2024. 2

[37] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, pages 1746–1754, 2017. 2

[38] Jiakai Sun, Han Jiao, Guangyuan Li, Zhanjie Zhang, Lei Zhao, and Wei Xing. 3dgstream: On-the-fly training of 3d gaussians for efficient streaming of photo-realistic free-viewpoint videos. In *CVPR*, pages 20675–20685, 2024. 2

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114, 2019. 6

[40] 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 *ICCV*, pages 8406–8415, 2023. 2

[41] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *CVPR*, pages 2708–2717, 2022. 1, 2

[42] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Ca-group3d: Class-aware grouping for 3d object detection on point clouds. *NeurIPS*, 35:29975–29988, 2022. 1, 2

[43] Lening Wang, Wenzhao Zheng, Yilong Ren, Han Jiang, Zhiyong Cui, Haiyang Yu, and Jiwen Lu. Occsora: 4d occupancy generation models as world simulators for autonomous driving. *arXiv preprint arXiv:2405.20337*, 2024. 2

[44] Tai Wang, Xiaohan Mao, Chenming Zhu, Runsen Xu, Ruiyuan Lyu, Peisen Li, Xiao Chen, Wenwei Zhang, Kai Chen, Tianfan Xue, Xihui Liu, Cewu Lu, Dahua Lin, and Jiangmiao Pang. Embodiedscan: A holistic multi-modal 3d perception suite towards embodied ai. In *CVPR*, 2024. 2

[45] 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. *arXiv preprint arXiv:2303.03991*, 2023. 2

[46] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *ICCV*, pages 21729–21740, 2023. 2, 6, 7

[47] Shun-Cheng Wu, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scfusion: Real-time incremental scene reconstruction with semantic completion. In *3DV*, pages 801–810. IEEE, 2020. 2

[48] Yuting Xiao, Xuan Wang, Jiafei Li, Hongrui Cai, Yanbo Fan, Nan Xue, Minghui Yang, Yujun Shen, and Shenghua Gao. Bridging 3d gaussian and mesh for freeview video rendering. *arXiv preprint arXiv:2403.11453*, 2024. 2

[49] Xiuwei Xu, Chong Xia, Ziwei Wang, Linqing Zhao, Yueqi Duan, Jie Zhou, and Jiwen Lu. Memory-based adapters for online 3d scene perception. *arXiv preprint arXiv:2403.06974*, 2024. 2

[50] Chi Yan, Delin Qu, Dan Xu, Bin Zhao, Zhigang Wang, Dong Wang, and Xuelong Li. Gs-slam: Dense visual slam with 3d gaussian splatting. In *CVPR*, pages 19595–19604, 2024. 2

[51] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *arXiv preprint arXiv:2406.09414*, 2024. 6

[52] Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. *arXiv preprint arXiv:2408.14197*, 2024. 2

[53] Zeyu Yang, Zijie Pan, Xiatian Zhu, Li Zhang, Yu-Gang Jiang, and Philip HS Torr. 4d gaussian splatting: Modeling dynamic scenes with native 4d primitives. *arXiv preprint arXiv:2412.20720*, 2024. 2

[54] Jiawei Yao, Chuming Li, Keqiang Sun, Yingjie Cai, Hao Li, Wanli Ouyang, and Hongsheng Li. Ndc-scene: Boost monocular 3d semantic scene completion in normalized device coordinates space. In *ICCV*, pages 9455–9465, 2023. 2

[55] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *CVPR*, pages 3947–3956, 2019. 2

[56] Hongxiao Yu, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Monocular occupancy prediction for scalable indoor scenes. *arXiv preprint arXiv:2407.11730*, 2024. 2, 5, 6, 7

[57] Vladimir Yugay, Yue Li, Theo Gevers, and Martin R Oswald. Gaussian-slam: Photo-realistic dense slam with gaussian splatting. *arXiv preprint arXiv:2312.10070*, 2023. 2

[58] Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *ECCV*, 2024. 2

[59] Sicheng Zuo, Wenzhao Zheng, Yuanhui Huang, Jie Zhou, and Jiwen Lu. Pointocc: Cylindrical tri-perspective view for point-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2308.16896*, 2023. 2