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
https://ykiwu.github.io/EmbodiedOcc/

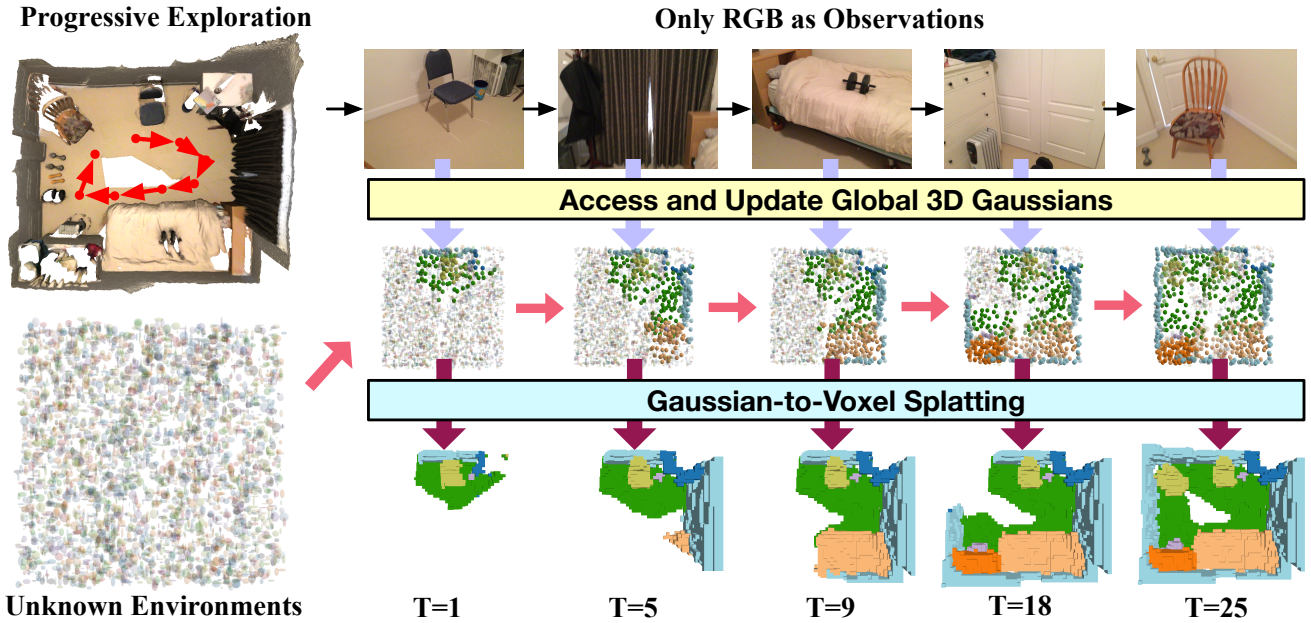

Figure 1. Accepted real-time monocular RGB inputs, our EmbodiedOcc can conduct embodied occupancy prediction in indoor scenes. We initialize the scene to be explored with uniform 3D semantic Gaussians and update the maintained Gaussian memory online based on real-time observations. As exploration progresses, the occupancy prediction for the global scene given by our EmbodiedOcc continually improves, which is exactly the capability a promising embodied agent should possess.

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 which demands 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. Experiments demonstrate that our EmbodiedOcc outperforms existing local prediction methods and accomplishes the embodied occupancy prediction with high accuracy and strong expandability. Code: https://github.com/YkiWu/EmbodiedOcc.

1. Introduction

With the rapid development of embodied intelligence and active agents, 3D scene perception [19, 21, 26, 27] has become a crucial task in computer vision. Intelligent agents

*Equal contribution. †Project leader.

exploring indoor scenarios make decisions and execute downstream tasks by perceiving and comprehending their surrounding environments. 3D perception capabilities required by these agents are diverse, among which 3D occupancy prediction [1, 7, 9, 31, 40] is gaining increasing popularity due to its efficiency, uniformity, and scalability.

Whereas 3D occupancy prediction based on visual information in outdoor driving scenarios [7, 9, 13, 25, 28, 30, 31, 37, 42, 43] has made significant progress, indoor research with the same settings is still at the preliminary stage of exploration due to the diversity and complexity of indoor scenes. Most existing methods [1, 38, 40] derive the local offline 3D local occupancy prediction by integrating semantic and depth information extracted from the visual inputs. However, these works are inherently inconsistent with the core requirements of embodied agents. A more promising active agent should be capable of progressively exploring and updating the global occupancy of a 3D scene with the change of its position and perspective.

To bridge the gap between existing research and practical scenarios, we formulate an embodied 3D occupancy prediction task to evaluate the ability of a perception model to explore an unknown scene progressively. We propose a Gaussian-based EmbodiedOcc framework to accomplish this new task. We initialize the global scene with uniform 3D semantic Gaussians and progressively update them located within the field of view observed by the agent. Throughout the whole exploration process, we maintain an explicit global memory of 3D Gaussians, accompanied by a Gaussian-to-voxel splatting module [9] which derives the current 3D occupancy. Specifically, we propose a local prediction module for each update, extracting semantic features from the observed monocular image and incorporating them via deformable cross-attention. These well-integrated features are used to update the Gaussians within the frustum. Our local prediction module employs a simple yet effective depth-aware branch to introduce explicit hint information for each Gaussian, ensuring the update of these Gaussians to better align with the local structures. During the continuous exploration of the same scene by the agent, the Gaussians within the current frustum are taken from the memory, and those updated before can provide information of high confidence for the update of this frame. This ensures the consistency of the 3D representation during the fusion and update process, which actually benefits from the physical meaning and structural information of Gaussians.

We reorganize an EmbodiedOcc-Scannet benchmark for the embodied 3D occupancy prediction task based on the locally annotated Occ-Scannet dataset [2, 40]. Experiments demonstrate that our EmbodeidOcc outperforms existing methods in terms of local occupancy prediction and accomplishes the embodied occupancy prediction of indoor scenes with high accuracy and strong expandability.

2. Related Work

3D Occupancy Prediction. Benefiting from its compactness and versatility, 3D occupancy prediction has gained great popularity in both indoor and outdoor scenes over the last few years. Methods based on multi-view images or additional 3D information[7–9, 12, 25, 31] have made significant advancements in many scenarios. MonoScene[1] was the first to derive 3D occupancy prediction from a single image, and subsequent works[38, 40] further focused on addressing the depth ambiguity in this monocular setting, collectively propelling this field into a more challenging stage. However, the majority of these efforts were confined to local and offline prediction. EmbodiedScan[17, 29] introduced a comprehensive framework capable of continuous occupancy prediction from multi-modal sequential inputs. Despite this, embodied online 3D occupancy prediction based on real-time monocular visual input is more aligned with the requirements of embodied agents.

Online 3D Scene Perception. Accurate comprehension of 3D scenes is an indispensable capability for embodied agents. Many tasks, such as 3D occupancy prediction[1, 40] and object detection[20, 27], are direct manifestations of this capability. Currently, most works on 3D scene perception[5, 19, 39] were conducted offline, taking preacquired and reconstructed 3D data to obtain a relatively lagging perception. Based on this situation, Online3D[34] introduced an adapter-based model that equips mainstream offline frameworks with the competence to perform online scene perception, which means they can process real-time RGB-D sequences. However, this framework still fails to overcome the intrinsic limitation of conventional point cloud modality. In a more general embodied scenario, real-time monocular visual input for scene perception can further advance the research on embodied agents.

3D Gaussian Splatting. 3D Gaussian Splatting[10] uses anisotropic 3D Gaussians to model a 3D scene, renowned for its fast speed and high quality in the field of radiance field rendering. The explicit physical characteristics of 3D Gaussians and the splat-based rasterization employed during rendering have also motivated rapid advancements in research fields such as scene editing[6, 18, 23], dynamic scenarios[4, 16, 33], and SLAM[3, 11, 35, 41]. GaussianFormer[9] pioneers the application of 3D Gaussians in outdoor 3D semantic occupancy prediction, updating Gaussians through comprehensive features extracted from multi-view images. These Gaussians are ultimately converted into local 3D occupancy prediction through an elaborately designed Gaussian-to-voxel splatting module. Compared to conventional voxel-based methods, using 3D Gaussian representation constitutes a more flexible approach. In this paper, we will leverage this significant attribute to accomplish embodied occupancy prediction in indoor scenarios.

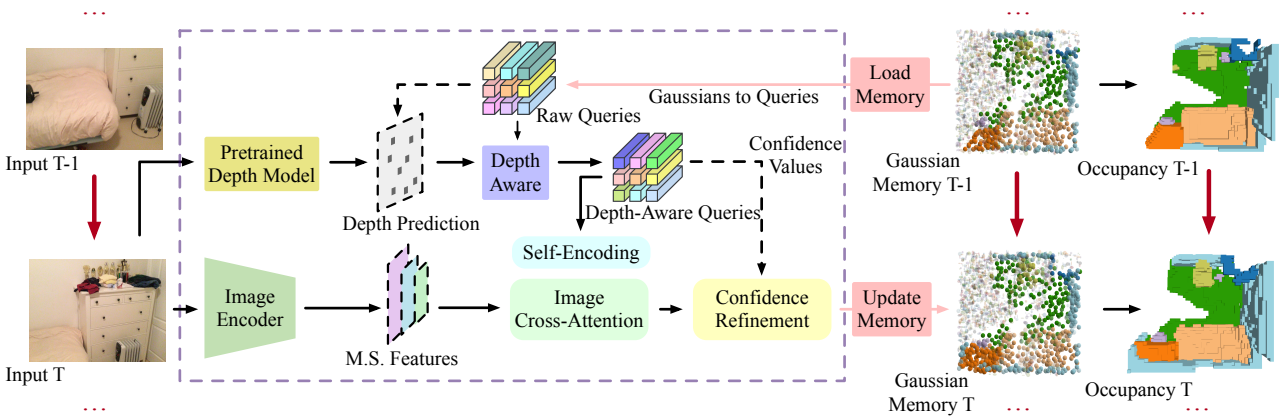

Figure 2. **Framework of our EmbodiedOcc for embodied 3D occupancy prediction.** We maintain an explicit global memory of 3D Gaussians during the exploration of the current scene. For each update, the Gaussians within the current frustum are taken from the memory and updated using semantic and structural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

3. Proposed Approach

3.1. Embodied 3D Occupancy Prediction

Conventional works in indoor scenarios for occupancy prediction accepted RGB-Ds or depth inputs to predict the semantic occupancy of a 3D scene. This setting provides the model with ample information for inference. However, it undoubtedly diminishes the comprehension capability of the model in practice. We humans are capable of effortlessly processing the visual information gathered by binoculus to obtain an initial 3D perception of their surroundings. Many recent approaches have focused on endowing models with the same competence, which means to accept monocular RGB image as input and derive a 3D occupancy prediction within the current frustum. We have:

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

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 application 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 real-time visual inputs gathered egocentrically to update the 3D occupancy prediction of the current scene. This capability facilitates the execution of downstream tasks based on real-time perception.

To this end, 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 this embodied occupancy prediction task, 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:

$$\begin{aligned} \mathbf{Y}_1 &= \mathcal{F}_{embodied}(x_1), \\ \mathbf{Y}_t &= \mathcal{F}_{embodied}(\mathbf{Y}_{t-1}, x_t), \end{aligned} \tag{2}$$

where $\mathcal{F}_{embodied}$ is the embodied occupancy 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. X_{room}, Y_{room}, Z_{room} represent the dimensions of the whole scene, which differ in value from the monocular setting but share the same world coordinate system.

3.2. Local Occupancy Prediction Module

Differing from conventional works that conducted feature integration in a voxelized space, GaussianFormer[9] first proposed an object-centric 3D representation to complete the 3D occupancy prediction task. Each semantic Gaussian can describe a local region and we can calculate the summation of the contribution of surrounding Gaussians to get the occupancy prediction result at a specific point. Motivated by this, we design our local and embodied occupancy prediction module based on this representation. Elaborate designs tailored for indoor perception characteristics will

fully leverage the flexibility and scalability inherent in this representation. We will first explain our local occupancy prediction module in this subsection.

Local Prediction in Camera Coordinate System. We use a set of 3D semantic Gaussians to represent an indoor scene and update the Gaussian-based representation according to semantic and structural features extracted from the input image. The extrinsics in indoor scenarios are constantly changing, which will pose additional difficulties for our local occupancy prediction module. Therefore, for each prediction, we initialize a set of 3D semantic Gaussians in the camera coordinate system. Each Gaussian is represented by a vector which is composed of 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 $\mathbf{c} \in \mathbb{R}^C$. The interactions between image features and Gaussians, as well as the interactions among Gaussians, will all take place in the camera coordinate system.

The update for each Gaussian $\mathbf{G} = (\mathbf{m}, \mathbf{s}, \mathbf{r}, \mathbf{o}, \mathbf{c})$ is achieved by directly updating its corresponding high-dimensional feature vector $\mathbf{Q} \in \mathbb{R}^m$. Following GaussianFormer, we use a self-encoding module and an image cross-attention module to facilitate effective interaction among these feature vectors and the image features extracted by an image backbone. Ultimately, these high-dimensional feature vectors with aggregated information will be used to obtain the update amounts $\Delta\mathbf{G} = (\Delta\mathbf{m}, \Delta\mathbf{s}, \Delta\mathbf{r}, \Delta\mathbf{o}, \Delta\mathbf{c})$ for the corresponding Gaussian properties:

$$\mathbf{G}_{new} = (\Delta\mathbf{m}+\mathbf{m}, \Delta\mathbf{s}+\mathbf{s}, \Delta\mathbf{r}\otimes\mathbf{r}, \Delta\mathbf{o}+\mathbf{o}, \Delta\mathbf{c}+\mathbf{c}), \quad (3)$$

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

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 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 refinement of 3D semantic Gaussians in our local prediction module.

We first use a fine-tuned 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 a portion of Gaussians. Although providing direct hints for the means of some Gaussians, this cannot exploit the deeper potential of the depth information. We design a simple yet effective depth-aware layer $\mathcal{M}_{depthaware}$ to awaken this potential. We still uniformly initialize a certain number of Gaussians within the current frustum. For each Gaussian, we project its mean $\mathbf{m}$ into the pixel coordinate system through the intrinsics

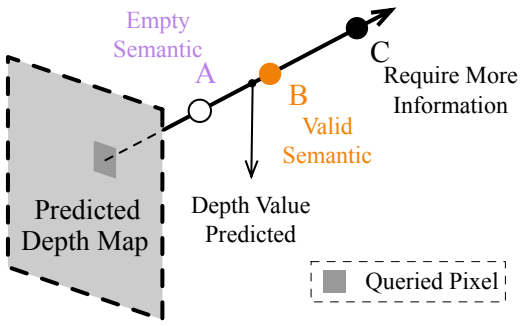

Figure 3. **Motivation of the depth-aware branch.** We use a depth-aware branch to provide local structural information for the update of each Gaussian. Along a specific ray, Gaussians distributed in front of the true depth point are likely to model the empty semantic (as Gaussian A). Gaussians distributed behind the true depth point closely are likely to model valid semantics (as Gaussian B). Those Gaussians that are distributed behind the true depth point but are too far away require more information to guide their updates (as Gaussian C).

$K_{mono} \in \mathbb{R}^{3\times3}$ of the camera and obtain the corresponding depth value d via the pixel coordinates. The sampled depth values $\mathbf{d}$, along with the z-components of the Gaussian means $\mathbf{z}$ in the camera coordinate system, are fed into the depth-aware layer. The depth-aware layer is a multi-layer perceptron(MLP) that outputs depth-aware features $\mathcal{Q}_{depth}$ for these Gaussians. Then we add these depth-aware features $\mathcal{Q}_{depth}$ to the feature vectors $\mathcal{Q}$ of these Gaussians, 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 refinement of other dimensions:

$$\begin{aligned} \mathbf{Q}_{depth} &= \mathcal{M}_{depthaware}(z, \mathcal{S}(D_{metric}, u, v)), \\ \mathcal{Q}_{mono} &= \{\hat{\mathbf{Q}}_i, i = 1, 2, ..., N | \hat{\mathbf{Q}}_i = \mathbf{Q}_i + \mathbf{Q}_{depth}\}, \end{aligned} \quad (4)$$

where $\mathcal{S}$ is the sample function to get the depth values, u, v refer to two pixel coordinates corresponding to each Gaussian and N is the total number of the Gaussians. We illustrate the guiding role of the depth-aware branch in Figure 3.

During the local occupancy prediction module, we conduct the refinement of Gaussians three times using the integrated features. After the final refinement, we use a Gaussian-to-Voxel Splatting proposed in GaussianFormer [9] to obtain the final occupancy within the frustum.

3.3. Gaussian Memory Updated Online

Suppose we are in a novel environment, we will first wander through it to explore the surroundings. During this process, the objects within the scene and their relationships are continuously updated in our minds, indicating the formation of a memory regarding this scene. Upon returning to the scene next time or revisiting it for further exploration, the visual information received this time merely serves to refine this memory. Indeed, the embodied occupancy predic-

tion framework we propose in this paper operates similarly. In this subsection, we will elaborate on how we maintain and update the Gaussian memory used in the final embodied occupancy prediction framework.

Online Prediction in World Coordinate System. Although our local occupancy prediction module initializes and updates Gaussians in the camera coordinate system to conduct offline occupancy prediction based on monocular input, during the transition from offline to online prediction, it is necessary to initialize the entire scene with uniform Gaussians in the world coordinate system. For a novel scene which is to be explored, we have: $\mathcal{G}_{room} = \{\mathbf{G}_i, i = 1, 2, ..., N | \mathbf{G}_i = (\mathbf{m}_i, \mathbf{s}_i, \mathbf{r}_i, \mathbf{o}_i, \mathbf{c}_i)\}$, 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).

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 Gaussian memory. These Gaussians will interact and be refined following the pipeline in 3.2. Then we detach these Gaussians and put them back into the Gaussian memory.

Confidence Refinement. Apart from the initial update for each scene which is akin to the local prediction, subsequent 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 more accurate physical properties and 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 this end, 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. For the Gaussians taken from the memory, we generate a set of confidence values Θ based on their tags Γ. For those Gaussians within the frustum and marked as having been previously updated($\Gamma = \mathbf{1}$), we set their confidence values to a certain value between 0 and 1, and they are slightly updated in the current update. For those Gaussians that have never been updated, we set their confidence values to 0, indicating that they will be the focus of the current update. During the refinements, we have:

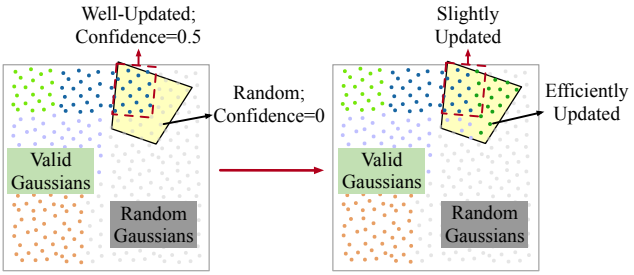

Figure 4. **Illustration of our Gaussian memory and confidence refinement.** During each update, the Gaussians within the current frustum are taken from the memory and updated according to their tags Γ. Confidence values of those well-updated Gaussians are set to a certain value between 0 and 1, while others are set to 0. The former will be updated slightly and the latter efficiently.

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

where we use $\oplus$ to represent the composition of rotation quaternions and the add operation of other parts. We use Figure 4 to illustrate how we maintain the Gaussian memory and refine Guassians according to their confidence values.

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 occupancy prediction 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[14], TPVFormer[7] and MonoScene[1]. We use monocular occupancy within the frustum $\mathbf{Y}^{fov}_{mono}$ and the corresponding ground truth $\mathbf{Y}^{fov}_{gt}$ to compute the loss. So the final expression of the loss is:

$$\begin{aligned} \mathcal{L} = \lambda_1 \mathcal{L}_{focal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) + \mathcal{L}_{lov}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) \\ + \mathcal{L}^{geo}_{scal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}) + \mathcal{L}^{sem}_{scal}(\mathbf{Y}^{fov}_{mono}, \mathbf{Y}^{fov}_{gt}), \end{aligned} \tag{6}$$

where λ_1 is a balance factor.

Then we use the trained local occupancy prediction module in the monocular setting to train our EmbodiedOcc model. For efficient training, we initialize the Gaussian memory of a scene before the first update and compute the current loss following equation 6 after each update. To ensure consistency, the local occupancy ground truth used for every loss calculation is obtained correspondingly from the occupancy of the whole scene. After a certain number of updates, we reinitialize the Gaussian memory and come to the prediction of the next scene. Trained with such a pipeline, our EmbodiedOcc is capable of effectively

performing 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

In this paper, we propose an EmbodiedOcc-ScanNet benchmark based on the locally annotated Occ-Scannet dataset. We explain our benchmark in detail in three parts: task descriptions, datasets, and evaluation metrics we use.

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 images as input and obtain the occupancy prediction within the frustum of the corresponding camera. Embodied occupancy prediction accepts real-time visual inputs continuously and updates the occupancy prediction of the current scene online. The visual input at a certain step t during embodied occupancy prediction is still monocular, which is a relatively challenging setting compared with multi-view input or input with 3D information.

Datasets. In the local occupancy prediction task, we used the Occ-ScanNet dataset [40] 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. We trained and evaluated our local occupancy prediction module on this dataset.

Based on this dataset, we reorganized an EmbodiedOcc-ScanNet dataset to train and evaluate our EmbodiedOcc framework [22, 40]. During the training and evaluation of our EmbodiedOcc framework, we have to ensure that scenes in the training set are different from those in the evaluating set. So we split the scenes again and obtained our final EmbodiedOcc-ScanNet dataset, which comprises 537/137 scenes in the train/val splits. Each scene in the EmbodiedOcc-ScanNet dataset consists of 30 posed frames with their corresponding occupancy. The resolutions of global occupancy of each scene are calculated by $(l_x \times l_y \times l_z)/0.08m$, where $(l_x \times l_y \times l_z)$ is the range of this scene in the world coordinate system. In addition, we maintain a global mask of the visible range in corresponding global occupancy for each 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.

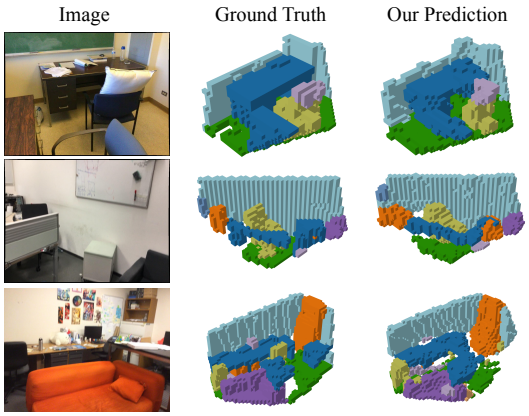

Figure 5. **Qualitative visualization of our local occupancy prediction.** The input image is displayed on the left and our prediction on the right, while the ground truth is shown in the middle.

To summarize, we conducted the local occupancy prediction task on the Occ-ScanNet and Occ-ScanNet-mini2 datasets and conducted the embodied occupancy prediction task on the EmbodiedOcc-ScanNet and EmbodiedOcc-ScanNet-mini datasets.

Evaluation Metrics. We use mIoU and IoU as the evaluation metric. For local occupancy prediction, we calculate the mIoU and IoU using the occupancy within the box(same with the evaluation in ISO[40]). 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 in the current scene.

4.2. Implementation Details

Local Occupancy Prediction Module. Following existing works[9, 40], we use a pre-trained EfficientNet-B7 [24] to initialize the image encoder in our local occupancy prediction module. The depth prediction network used in the depth-aware branch is a fine-tuned DepthAnything-V2 model [36], remaining frozen during the training. The depth-aware layer is a 3-layer MLP and the other parts of our local occupancy prediction module follow the GaussianFormer[9]. 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[15] 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. We train our local occupancy prediction mod-

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
MonoScene [1]	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 [40]	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
Ours	x^{rgb}	**53.95**	**40.90**	**50.80**	**41.90**	**33.00**	**41.20**	**55.20**	**61.90**	**43.80**	**35.40**	**53.50**	**42.90**	**45.48**

Table 2. **Embodied Prediction Performance on the EmbodiedOcc-ScanNet dataset.** SplicingOcc refers to the splicing results of all the local prediction results in the same scene. We use the results from our local occupancy prediction module to build this baseline as our method has achieved the best local performance to date.

Method	Dataset	IoU	ceiling	floor	wall	window	chair	bed	sofa	table	tvs	furniture	objects	mIoU
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**
SplicingOcc	EmbodiedOcc-mini	48.75	**29.00**	37.60	37.30	**26.80**	44.50	**65.90**	52.70	40.80	**36.60**	54.50	**27.90**	41.24
EmbodiedOcc	EmbodiedOcc-mini	**50.78**	22.10	**43.70**	**39.00**	26.60	**45.00**	63.70	**54.40**	**43.90**	34.70	**55.30**	27.60	**41.45**

Table 3. **Look-Back Prediction *vs* First-Time Prediction on the EmbodiedOcc-ScanNet dataset.** For $\mathrm{K} = k$, we simply select $0, 1, ..., \mathrm{k}-1$th frames to evaluate our EmbodiedOcc and the occupancy ground truth used here is the union of the frustums corresponding to the k frames. 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, 0, 1, 2]	**50.00**	**39.93**
First-Time	5	[0, 1, ..., 4]	50.13	40.03
Look-Back	5	[0, 1, ..., 4, 0, 1, ..., 4]	**50.64**	**40.42**
First-Time	8	[0, 1, ..., 7]	50.94	40.86
Look-Back	8	[0, 1, ..., 7, 0, 1, ..., 7]	**51.20**	**41.03**

ule 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 $0.16m$ 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 occupancy prediction module.

4.3. Results and Analysis

Local Occupancy Prediction. We evaluated our local occupancy prediction module on the Occ-ScanNet dataset. As shown in Table 1, the results indicate that our local occupancy prediction module outperforms ISO[40]. We also conducted visualization to demonstrate the performance of our local occupancy prediction module in Figure 5.

Embodied Occupancy Prediction. We spliced the local occupancy obtained from our local occupancy prediction module to serve as the baseline, on which we evaluated the performance of our EmbodiedOcc. Firstly, we assessed the occupancy prediction for the entire scene after processing all frames (30 frames), and the ground truth for calculating IoU and mIoU is the union of the frustums. Table 2 presents a performance comparison between our EmbodiedOcc and the baseline. It can be observed that our EmbodiedOcc exhibits superior prediction of the scene, which is achieved through the integration of different views. We conducted qualitative visualization to demonstrate the performance of our EmbodiedOcc in Figure 6.

Secondly, we expect EmbodiedOcc to have improved performance when encountering parts that have been explored before and thus conducted a Look-Back evaluation. Specifically, after processing K frames, we direct the model to come back to the first frame and reprocess these K 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.

Analysis of the Gaussian Parameters and the Depth-Aware Branch. We analyze the effect of different Gaussian parameters and the depth-aware branch in Table 4. We see that decreasing the number or increasing the scale of the Gaussians can lead to a decrease in performance during both 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 pre-

Table 4. **Effect of the Gaussian parameters and the depth-aware branch.** We conducted the local occupancy prediction task on the Occ-ScanNet-mini2 dataset and the embodied occupancy prediction task on the EmbodiedOcc-ScanNet-mini dataset.

Depth Type	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
Depth-aware branch	16200	0.01	0.08	(0.16, 0.16, 0.16)	**53.93**	**46.20**	**50.78**	**41.45**
Naive-depth branch	16200	0.01	0.08	(0.16, 0.16, 0.16)	50.32	42.73	/	/
No-depth branch	16200	0.01	0.08	(0.16, 0.16, 0.16)	48.15	40.07	37.52	30.73
Depth-aware branch	8100	0.01	0.08	(0.20, 0.20, 0.20)	50.47	42.82	46.24	37.99
Depth-aware branch	16200	0.01	0.20	(0.16, 0.16, 0.16)	51.57	43.74	48.09	38.40

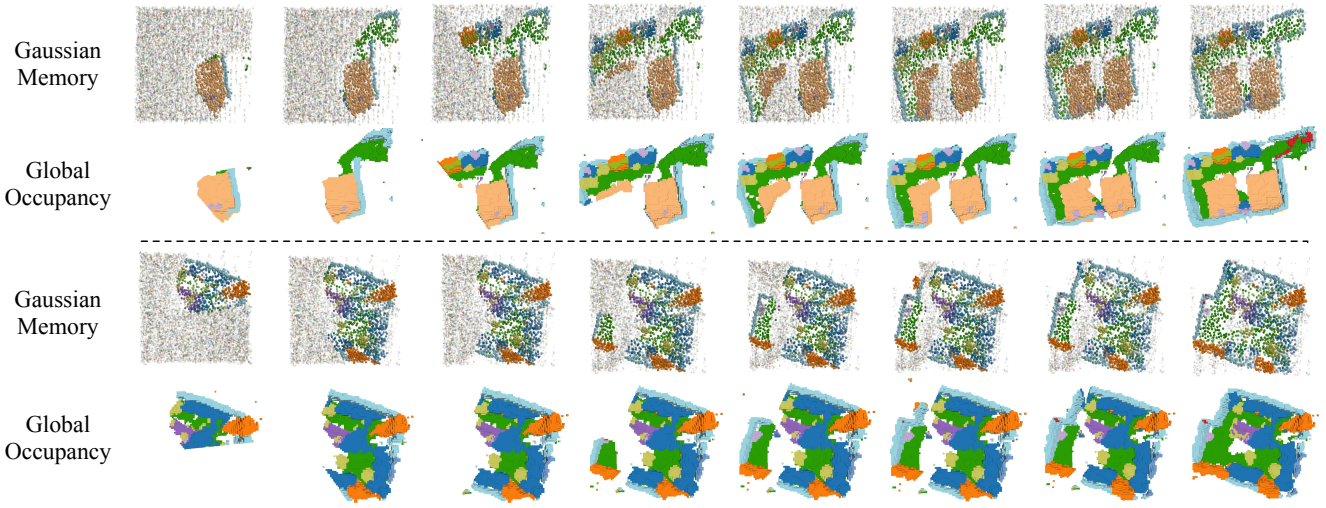

Figure 6. **Visualization of the embodied occupancy prediction.** We select two scenes to show the update of Gaussian memory and corresponding global occupancy with continuous exploration. As the Gaussians transition from random to increasingly ordered, the occupancy prediction of the current scene becomes more accurate and complete.

diction, while Gaussians with too large scale will overlap and influence each other which is also detrimental to the correct prediction of occupancy. We also find that depth information will significantly benefit the local and embodied occupancy prediction. As shown in the third row of Table 4, 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. Results in Table 4 also 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.

Analysis of the Confidence Refinement. During each update, current Gaussians are refined through three refinement layers. We froze the first two refine layers and equally updated all Gaussians in the last refine layer when training our EmbodiedOcc. Table 5 presents the impact of varying numbers of frozen refinement layers and different confidence values (determines the coefficient of each $\Delta \mathbf{G}$) on the embodied occupancy prediction task. We can observe that moderate updates to those previously processed Gaussians yield the best embodied occupancy prediction.

Table 5. **Ablation study of the confidence refinement.**

Frozen Layers	Confidence Layers	Coefficient θ	Embodied Prediction IoU	Embodied Prediction mIoU
2	1	0.5	**50.78**	**41.45**
3	0	0.5	48.33	39.44
1	2	0.5	50.36	40.99
0	3	0.5	50.18	40.28
2	1	0.7	50.53	41.05
2	1	0.3	50.15	40.80

5. Conclusion

In this paper, we have formulated 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 has paved the way for enabling active agents to conduct accurate and flexible embodied occupancy prediction.

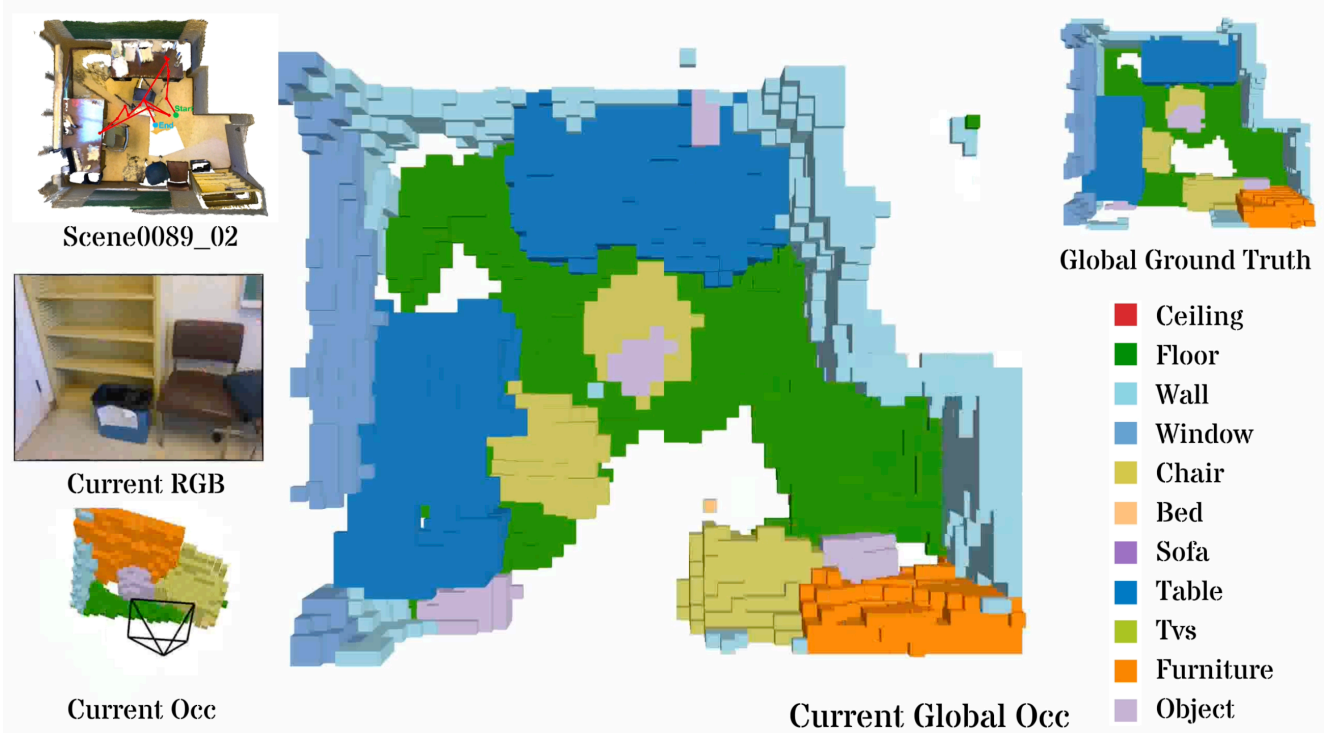

Figure 7. **Visualizations of the proposed EmbodiedOcc for Embodied 3D Occupancy Prediction on the EmbodiedOcc-ScanNet.** We visualize the current monocular RGB input and local occupancy prediction given by our EmbodiedOcc in the bottom left corner, and the global occupancy ground truth of the current scene in the top right corner. The global occupancy for the current scene given by our EmbodiedOcc is right in the center.

A. EmbodiedOcc-ScanNet Dataset Details

We reorganize our EmbodiedOcc-ScanNet dataset following the data formulation used in NYUv2 [22] and Occ-ScanNet [40]. We noted that the Occ-ScanNet dataset consists of frames sampled from the original ScanNet [2] dataset randomly, which means that different frames may come from the same indoor scene. For all scenes in the Occ-ScanNet dataset, we selected 537 scenes to constitute the training set for EmbodiedOcc-ScanNet, and 137 scenes to form the evaluation set. We split these scenes in this way to ensure that scenes in the training set are different from those in the evaluation set.

For each scene, we first obtain a global occupancy of it from the voxel labels in the CompleteScanNet [32] dataset using the K-Nearest Neighbors algorithm. Next, we count and resample the frames of this scene in the Occ-ScanNet dataset using a certain interval to obtain 30 posed images. For each frame, we select a specific area in front of the camera as the range of local occupancy. The selection of the local voxel origin is consistent with the Occ-ScanNet [40]. Then, we obtain the current local occupancy from the global occupancy using the K-Nearest Neighbors algorithm. In addition to this, we maintain a mask in global resolutions for each frame, which marks the intersection of the current local voxel and frustum. This allows us to obtain the occupancy ground truth of the explored area by splicing together the masks of processed frames, enhancing the flexibility of our EmbodiedOcc-ScanNet. The pipeline to generate one scene in our EmbodiedOcc-ScanNet is shown in Figure 8.

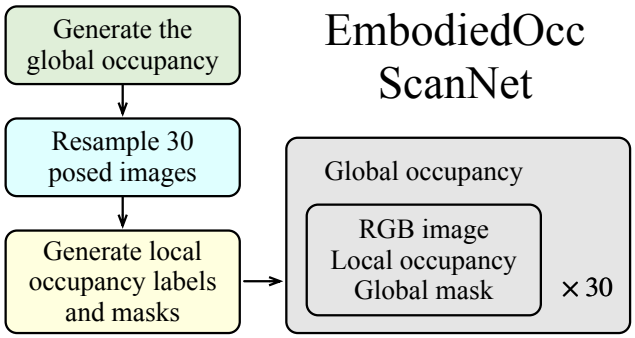

Figure 8. **Pipeline of our EmbodiedOcc-ScanNet.**

B. Additional Visualizations

Due to space limitations, we only selected a few frames in the main text to demonstrate the performance of our local occupancy prediction module. In Figure 9, we use a more diverse set of monocular samples to further showcase the visual effects of the local occupancy obtained by our local occupancy prediction module.

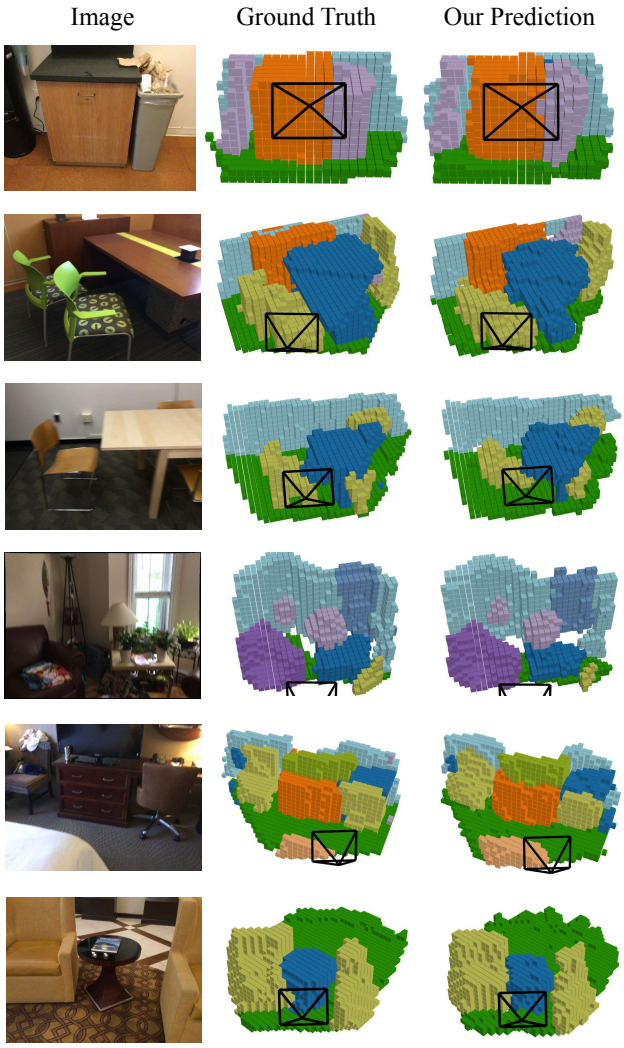

Figure 9. **Additional visualizations of our local occupancy prediction module.**

To fully demonstrate the working process of our EmbodiedOcc, we use a video demo to showcase the performance of EmbodiedOcc when exploring indoor scenes. Figure 7 shows a sampled image from the video demo for embodied 3D occupancy prediction on the EmbodiedOcc-ScanNet. The video demo and our implementation code are both included in the zip folder.

References

[1] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *CVPR*, pages 3991–4001, 2022. 2, 5, 7

[2] 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, 9

[3] 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

[4] 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

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

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

[7] 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

[8] 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.

[9] 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

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

[11] 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

[12] 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

[13] 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

[14] 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

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

[16] 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

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

[18] 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

[19] 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

[20] 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

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

[22] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760, 2012. 6, 9

[23] 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

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

[25] 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

[26] 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

[27] Haiyang Wang, Lihe Ding, Shaocong Dong, Shaoshuai Shi, Aoxue Li, Jianan Li, Zhenguo Li, and Liwei Wang. Cagroup3d: Class-aware grouping for 3d object detection on point clouds. *NeurIPS*, 35:29975–29988, 2022. 1, 2

[28] 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

[29] 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

[30] 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

[31] 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

[32] Shun-Cheng Wu, Kesuke Tateno, Nassir Navab, and Federico Tombari. Scfusion: Real-time incremental scene reconstruction with semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 801–810, 2020. 9

[33] 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

[34] 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

[35] 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

[36] 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

[37] 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

[38] 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

[39] 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

[40] Hongxiao Yu, Yuqi Wang, Yuntao Chen, and Zhaoxiang Zhang. Monocular occupancy prediction for scalable indoor scenes. *arXiv preprint arXiv:2407.11730*, 2024. 2, 6, 7, 9

[41] 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

[42] 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

[43] 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