FLM-Occ: Feed-forward Likelihood Maximization for Efficient Indoor Occupancy Prediction

Guangcheng Chen[1,3], Lihuang Fang[1,3], Huaqi Tao[1,3], Yicheng He[1,3], Li He*[1,2], and Hong Zhang*[1,3]

[1] Southern University of Science and Technology
[2] Guangdong Laboratory of Artificial Intelligence and Digital Economy (Shenzhen)
[3] Shenzhen Key Laboratory of Robotics and Computer Vision
https://gcchen97.github.io/flm-occ/

Abstract. Recent indoor occupancy prediction methods adopt Gaussian primitives as a sparse 3D representation for computational efficiency. However, their training relies on voxel classification, which imposes only local constraints and lacks global supervision on the distribution of the primitives. Therefore, they inevitably predict spurious primitives in empty regions, undermining both representational and computational efficiency. To address this, we propose Feed-forward Likelihood Maximization (FLM), a novel framework that reformulates occupancy prediction as voxel distribution estimation. In FLM, a network is trained to predict a mixture model that maximizes the likelihood over ground-truth occupied voxels in a feed-forward manner. To enable end-to-end training of networks and voxelization of a standard mixture model, we define mixture weights as normalized primitive volumes to implicitly enforce simplex constraints and derive novel voxelization formulas. Based on FLM, our FLM-Occ, a novel method that is capable of relocating randomly initialized primitives over long distances to model a scene. On Occ-ScanNet, FLM-Occ achieves superior accuracy using only 32 superquadrics, 2.7% of the prior SoTA, while running 3.7× faster.

1 Introduction

Monocular occupancy prediction aims to reconstruct a geometric and semantic 3D layout from a single RGB image in the form of a voxel grid [1,43]. It has been widely-studied in autonomous driving [34,39,50] and embodied AI [18,30,41,43]. Yet dense occupancy prediction suffers from cubic computational complexity, making it prohibitively expensive for real-time applications.

Recent methods have explored various 3D representations for efficiency. Among them, mixture model-based methods [9, 12, 18, 28, 41, 49] have drawn increasing attention. They require far fewer geometric primitives than sparse voxel-based methods [13, 16, 19, 22, 34, 39, 46] because each primitive can represent multiple voxels. They also provide continuous spatial representation as plane-based methods [7, 11, 17, 45], allowing self-supervision via image rendering [10]. During

* Li He and Hong Zhang are corresponding authors.

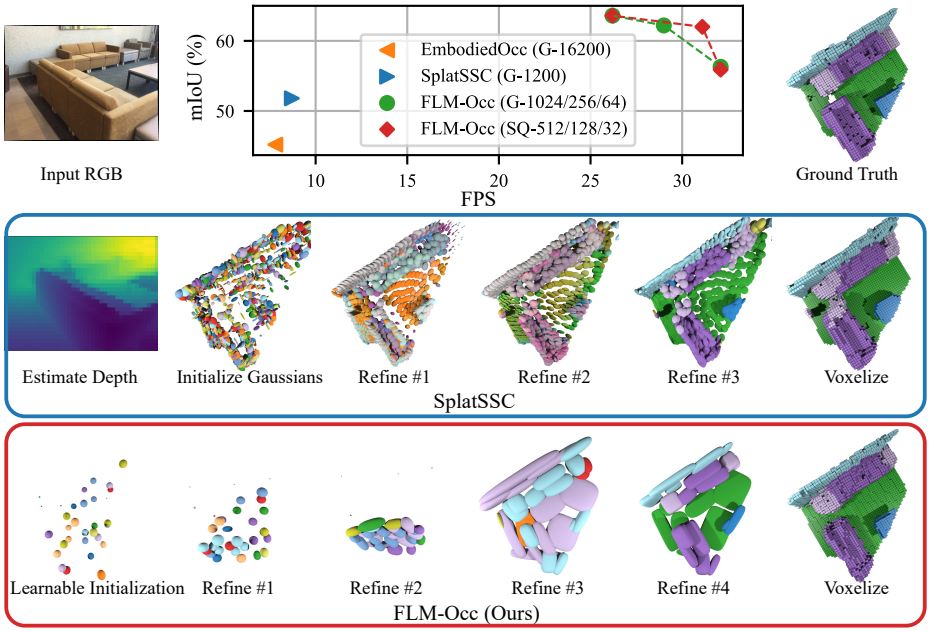

Fig. 1: Comparison (top-left) on Occ-ScanNet [44]. The numbers following each method name indicate primitive count. Our FLM-Occ achieves significantly better accuracy–efficiency trade-offs than EmbodiedOcc [41] and SplatSSC [28]. The second and third rows show the prediction process of SplatSSC and FLM-Occ.

inference, they initialize a fixed number of primitives and refine them to model the semantic distribution of the scene in the input image. However, as illustrated in Fig. 2, they suffer from a critical limitation: spurious primitives persist in empty regions, wasting representation capacity and increasing computation.

In this paper, we identify the root cause of this problem as the training objective—voxel classification. When these methods voxelize the predicted primitives, a voxel-wise cross-entropy loss is used for training. However, this loss cannot optimize the global distribution of the primitives and fails to induce distant primitive relocations. Our gradient analysis shows that this is because the training gradients of this loss with respect to primitive positions vanish when the primitives are far from ground-truth occupied voxels. Consequently, their networks are only able to make local adjustments of the primitives.

To resolve the above problem, we propose Feed-forward Likelihood Maximization (**FLM**), a novel framework that reformulates occupancy prediction as voxel distribution estimation, parameterized by a mixture model. This framework is based on maximum likelihood estimation (MLE) [25], which is a widely used method for estimating the parameters that best explain the observed data. Specifically, our FLM objective maximizes the joint likelihood of all ground truth

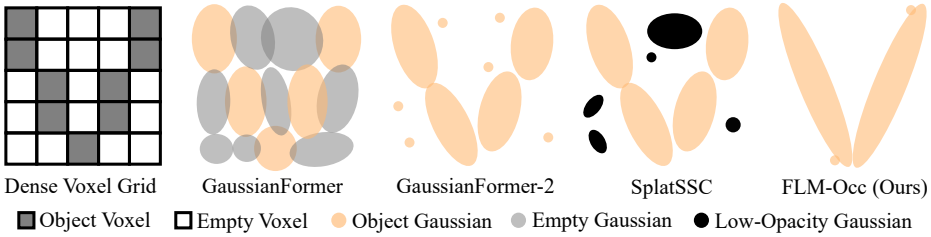

Fig. 2: Four Gaussian-based methods for occupancy prediction. GaussianFormer [12], GaussianFormer-2 [9], and SplatSSC [28] leave redundant primitives in empty regions because they are incapable of substantially relocating the Gaussians, undermining their representational efficiency. With MLE, FLM-Occ learns to relocate Gaussians toward occupied regions.

occupied voxels, and the trained network learns to predict a mixture model that maximizes the likelihood in a feed-forward manner.

As it turns out, enabling end-to-end training in FLM is non-trivial since simplex constraints are required for mixture weights and a new voxelization method is necessary for a standard mixture model. To tackle this issue, we define *the weights of the primitives as normalized primitive volumes*. This implicitly enforces the simplex constraints, enabling end-to-end training. In addition, it aligns each mixture weight with the spatial extent of the primitive, preventing degenerate cases where excessively large primitives receive vanishing weights and thereby preserving representational efficiency.

Finally, we present **FLM-Occ**, a lightweight network for monocular occupancy prediction. Unlike prior work that relies on depth maps to place the primitives, FLM-Occ directly refines them with the guidance of image features to match scene structures. FLM-Occ is able to substantially relocate randomly initialized primitives to align with objects. As shown in Fig. 1, FLM-Occ achieves compact representation, efficient inference, and superior performance on the OccScanNet dataset [44].

2 Related Work

Mixture Model-based Occupancy Prediction. Mixture model-based methods train neural networks to refine a set of primitives, such as Gaussians [12] or superquadrics [49], to model a scene. The pioneering work GaussianFormer [12] uniformly distributes Gaussians in 3D space and then deforms and classifies them to model the semantic distribution. The occupancy grid for calculating training loss is then extracted from the Gaussians. GaussianFormer [12] does not take advantage of the sparsity of 3D scenes for optimizing its efficiency, as it wastefully models empty regions with Gaussians. Later work exploits scene sparsity by using object-centric representations with fewer primitives. GaussianFormer-2 [9] predicts the distributions of objects along camera rays to initialize Gaussians accordingly. SplatSSC [28] uses a depth estimation model to predict depth maps

and places initial Gaussians at the estimated depths. Nevertheless, their primitives are adjusted locally, and thus some of them persist in empty regions. We address this problem by leveraging MLE for training, enabling distant primitive relocations and thus eliminating the need for initializing primitives near objects.

Training Objectives of Occupancy Prediction. Occupancy prediction has long been formulated as a voxel classification problem, regardless of the underlying 3D representation. OPUS [38] alternatively adopts a point regression paradigm, yet it still relies on voxel-wise regression and operations that remain computationally expensive. Other methods [6, 10, 14, 40, 48] adopt image reconstruction losses [15, 23, 24] for self-supervision, but their performance remains limited. In contrast, we reformulate occupancy prediction as voxel distribution estimation and train a network via MLE, enabling global supervision on the mixture models and encouraging large relocations of the primitives. This formulation allows us to fully exploit the representation capacity of a mixture model and thereby minimizes the number of primitives and accelerates inference.

Feed-forward 3D Gaussian Splatting (F3DGS). Since the pioneering work 3D Gaussian Splatting (3DGS) [15], mixture models have emerged as a transformative alternative representation in 3D reconstruction. Rather than estimating the parameters of a mixture model through iterative optimization, F3DGS methods [2, 3, 8, 32, 47] train networks to estimate the parameters in a feed-forward manner. However, these methods are unable to substantially relocate the Gaussians. In 3DGS [15], Gaussians are indirectly relocated through the non-differentiable densification and pruning [2, 15] process. As for F3DGS methods, they avoid this problem by representing scenes with pixel-aligned Gaussian maps [32] that align well with their pixel-wise training losses and regularize Gaussian means with depth estimation [5, 29, 36, 42]. In our work, we instead leverage MLE to learn global relocation of unordered primitives.

3 Methodology

We present our FLM and its application in monocular occupancy prediction for indoor scenes. We begin by reviewing the formulations of prior mixture model-based methods and their training objective in Section 3.1. We then introduce FLM in Sections 3.2, followed by the architecture of FLM-Occ in Section 3.3.

3.1 Preliminaries

Mixture model-based methods [9,12,28,41,49] refine a fixed number of primitives, denoted by $\{\mathbf{g}_j\}_{j=1}^{M}$, to model 3D scenes according to the input images. The primitive positions can be initialized either by uniform sampling in 3D space [12] or by back-projecting samples from a depth map [9,28]. The number of primitives is fixed during training and inference.

These methods can be categorized into two types, depending on whether they use object-centric representations. The pioneering work GaussianFormer [12] uses Gaussians to model scenes. Each Gaussian $\mathbf{g}_j$ consists of a position $\boldsymbol{\mu}_j \in \mathbb{R}^3$,

Table 1: Summary of mixture model-based methods. For notational simplicity, the density of voxel $\mathbf{x}$ from the j-th primitive $\mathcal{K}(\mathbf{x}|\mathbf{g}_j)$ is denoted by $\mathcal{K}_j$. Type 1 treats "empty" as one of the semantic classes, while Type 2 and ours model geometry and semantics separately.

Method	Type	Voxelization Formula		Training Objective
		Geometry	Semantics	
GaussianFormer [12]	1	$\sum_{j\in\mathcal{N}} \mathcal{K}_j \mathbf{f}_j$		Voxel Classifcation
GaussianFormer-2 [9]	2	$1-\prod_{j\in\mathcal{N}}(1-\mathcal{K}_j)$	$\sum_{j\in\mathcal{N}} \frac{a_j}{v_j}\mathcal{K}_j \mathbf{f}'_j / \sum_{j\in\mathcal{N}} \frac{a_j}{v_j}\mathcal{K}_j$	
QuadricFormer [49]		$1-\prod_{j\in\mathcal{N}}(1-\mathcal{K}_j)$	$\sum_{j\in\mathcal{N}} a_j\mathcal{K}_j \mathbf{f}'_j / \sum_{j\in\mathcal{N}} a_j\mathcal{K}_j$	
SplatSSC [28]		$1-\prod_{j\in\mathcal{N}}(1-a_j\mathcal{K}_j)$	$\sum_{j\in\mathcal{N}} \frac{1}{v_j}\mathcal{K}_j \mathbf{f}'_j / \sum_{j\in\mathcal{N}} \frac{1}{v_j}\mathcal{K}_j$	
FLM-Occ (ours)	New	$\sum_{j=1}^{M} \mathcal{K}_j$	$\sum_{j=1}^{M} \mathcal{K}_j \mathbf{f}_j / \sum_{j=1}^{M} \mathcal{K}_j$	MLE

scale $\mathbf{s}_j \in \mathbb{R}^3$, a rotation $\mathbf{R}_j \in \mathbb{R}^{3\times3}$, an opacity a_j and semantic logits $\mathbf{f}_j \in \mathbb{R}^n$. The semantic logits $\mathbf{c}$ of voxel $\mathbf{x}$ are aggregated from neighboring Gaussians:

$$\mathbf{c} = \sum_{j\in\mathcal{N}} \mathcal{K}(\mathbf{x}|\mathbf{g}_j)\mathbf{f}_j, \tag{1}$$

$$\mathcal{N} = \big\{ j \in \{1,2,\ldots,M\} \mid \|\mathbf{x}-\boldsymbol{\mu}_j\|_\infty < 3\|\mathbf{s}_j\|_\infty \big\}, \tag{2}$$

where $\mathcal{K}$ is the primitive function and $\mathcal{N}$ is the index set of neighboring Gaussians. $\mathcal{N}$ is used for reducing computation. This representation is more efficient than voxels because each Gaussian can present several voxels, but it still models empty regions with Gaussians.

GaussianFormer-2 [9] improves its predecessor by using an object-centric representation that separately models geometry and semantics, aiming to restrict Gaussians in occupied regions for efficiency. The probability of a voxel being occupied p and the semantics $\mathbf{c}$ are calculate as

$$p = 1 - \prod_{j\in\mathcal{N}} \big(1-\mathcal{K}(\mathbf{x}|\mathbf{g}_j)\big),\ \mathbf{c} = \frac{\sum_{j\in\mathcal{N}} \frac{a_j}{v_j}\mathcal{K}(\mathbf{x}|\mathbf{g}_j)\mathbf{f}'_j}{\sum_{j\in\mathcal{N}} \frac{a_j}{v_j}\mathcal{K}(\mathbf{x}|\mathbf{g}_j)}, \tag{3}$$

where $a_j \in [0,1]$, is opacity, $\mathbf{f}'_j$ is the softmax-normalized $\mathbf{f}_j$, and v_j is the volume of primitive $\mathbf{g}_j$ calculated as:

$$v_j = \int_{\mathbb{R}^3} \mathcal{K}(\mathbf{x}|\mathbf{g}_j) d\mathbf{x}. \tag{4}$$

Later work improves Eq. (3) by utilizing superquadrics [49] or devising more effective primitive weighting [28]. Tab. 1 summarizes the different methods.

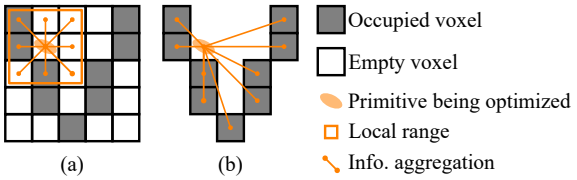

Fig. 3: Information graphs of the training losses for a single primitive via voxel classification (a) and MLE (b). Unlike voxel classification objectives, which supervise both empty and occupied voxels, our MLE-based FLM loss only models the distribution of occupied voxels.

Regardless of the voxelization formulas, both types of methods calculate the same voxel classification loss $\mathcal{L}_{ce}$ for "empty" and semantic classes with the cross-entropy function f_{ce} on the output $H \times W \times D$ voxel grid :

$$\mathcal{L}_{ce} = \frac{1}{HWD} \sum_{i}^{HWD} f_{ce}(\mathbf{c}'_i, \mathbf{c}_i^{(gt)}). \tag{5}$$

The only difference lies in the definition of $\mathbf{c}'_i$: type 1 uses $\mathbf{c}'_i = \mathbf{c}_i$ while type 2 uses $\mathbf{c}'_i = [1 - p_i; p_i \cdot \mathbf{c}_i]$.

A critical limitation of $\mathcal{L}_{ce}$ is its inability to relocate primitives to distant locations. This limitation can be shown by analyzing its gradient. We consider the case of binary classification without loss of generality and derive the gradient of $\mathcal{L}_{ce}$ for a ground-truth occupied voxel $\mathbf{x}_i$ w.r.t. the position of $\mathbf{g}_j$:

$$\frac{\partial \mathcal{L}_{ce}^{(i)}}{\partial \boldsymbol{\mu}_j} = -\frac{\prod_{r \neq j, r \in \mathcal{N}} \left(1 - \mathcal{K}(\mathbf{x}_i | \mathbf{g}_r)\right)}{1 - \prod_{r \in \mathcal{N}} \left(1 - \mathcal{K}(\mathbf{x}_i | \mathbf{g}_r)\right)} \cdot \frac{\partial \mathcal{K}(\mathbf{x}_i | \mathbf{g}_j)}{\partial \boldsymbol{\mu}_j}, \tag{6}$$

with $\mathcal{K}_r = \mathcal{K}(\mathbf{x}_i | \mathbf{g}_r)$ and $\mathcal{K}_j = \mathcal{K}(\mathbf{x}_i | \mathbf{g}_j)$. This suggest that the gradient $\partial \mathcal{L}_{ce}^{(i)} / \partial \boldsymbol{\mu}_j$ vanishes when any $\mathcal{K}_r$ is close to 1. In other words, if an occupied voxel is already explained by neighboring primitives, as shown in Fig. 3, this voxel cannot impact other primitives, leaving distant primitives stranded in empty regions with negligible gradients.

3.2 Feed-forward Likelihood Maximization

To overcome the limitation of voxel-wise classification loss, we revisit MLE, a widely used method for estimating probabilistic models. The goal of MLE is to find the parameters that maximize the joint probability for the observed data. This aligns well with the goal of occupancy prediction, i.e., to predict a mixture model that accurately fits the distribution of the ground-truth occupied voxels:

$$\{\hat{\mathbf{g}}_j\}_{j=1}^{M} = \arg\max_{\{\mathbf{g}_j\}_{j=1}^{M}} \prod_{i=1}^{P} \sum_{j=1}^{M} \frac{w_j}{v_j} \mathcal{K}(\mathbf{x}_i|\mathbf{g}_j) \tag{7}$$

$$\text{subject to} \quad \sum_{j=1}^{M} w_j = 1 \ . \tag{8}$$

where $w_j \in [0,1]$ is the mixture weight and $\{\mathbf{x}_i\}_{i=1}^{P}$ is the positions of the occupied voxels. $\mathcal{K}$ could be any unnormalized probabilistic kernel function.

Eq. (7) forms the basis of our Feed-forward Likelihood Maximization (FLM). Within FLM, a neural network $\mathcal{F}_\theta$ is trained to refine an initial mixture model $\{\mathbf{g}_j^{(0)}\}_{j=1}^{M}$, with the guidance of image features $\mathcal{F}_{enc}(\mathcal{I})$, such that the resulting model maximizes the scene likelihood:

$$\{\hat{\mathbf{g}}_j\}_{j=1}^{M} = \mathcal{F}_\theta\Big(\{\mathbf{g}_j^{(0)}\}_{j=1}^{M},\, \mathcal{F}_{enc}(\mathcal{I})\ \Big) \ . \tag{9}$$

However, the end-to-end training of $\mathcal{F}_\theta$ is non-trivial since directly utilizing Eq. (7) for training raises two issues: how to (1) handle the simplex constraint Eq. (8) and (2) voxelize a standard mixture model. These issues do not arise in previous methods because they replace mixture weights with unconstrained opacities and use different voxelization formulas designed for classification loss. To address them, we reparameterize mixture weights as normalized volumes:

$$w_j = \frac{v_j}{\sum_{j=1}^{M} v_j} \ . \tag{10}$$

This reparameterization implicitly enforces simplex constraints on the weights. It also ensures that each primitive's weight is proportional to its spatial extent so that the total volume $\sum_{j=1}^{M} v_j$ directly reflects the number of occupied voxels. Consequently, large primitives account for big structures, while smaller ones specialize in local details. With this reparameterization, the joint likelihood in Eq. (7) can be rewritten as:

$$\left(\prod_{i=1}^{P} \frac{1}{\sum_{j=1}^{M} v_j}\right) \cdot \left(\prod_{i=1}^{P} \sum_{j=1}^{M} \mathcal{K}(\mathbf{x}_i|\mathbf{g}_j)\right). \tag{11}$$

FLM Loss. To construct a loss for numerically stable training, we take the negative logarithm of the likelihood in Eq. (11) and average it over occupied voxels, resulting in the following FLM loss:

$$\mathcal{L}_{flm} = \underbrace{\log \sum_{j=1}^{M} v_j}_{\text{Volume sum}} \underbrace{- \frac{1}{P} \sum_{i=1}^{P} \log \sum_{j=1}^{M} \mathcal{K}(\mathbf{x}_i|\mathbf{g}_j)}_{\text{Density sum}} . \tag{12}$$

Interestingly, minimizing the volume term of FLM loss will compress the total volume, thereby reducing densities in empty regions, and increasing the density

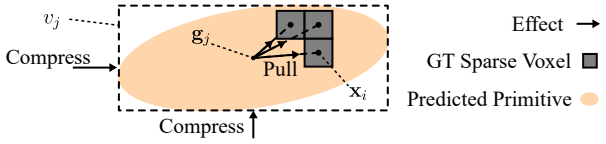

Fig. 4: 2D illustrations of the roles of the two terms in the FLM loss. The volume term compresses the primitives to match the voxel shape, and the density term pulls them toward voxel locations.

term will pull the primitives toward occupied voxels. In this sense, it explicitly drives all primitives toward objects and fully exploits their representational capacity. Therefore, using FLM loss allows us to use significantly fewer primitives than using $\mathcal{L}_{ce}$ (Eq. (5)). Fig. 4 illustrates the physical interpretation of the two terms of FLM loss. To understand why $\mathcal{L}_{flm}$ is able to overcome the limitation of $\mathcal{L}_{ce}$, we derive the gradients of FLM loss of $\mathbf{x}_i$ w.r.t. the position of $\mathbf{g}_j$:

$$\frac{\partial \mathcal{L}_{flm}^{(i)}}{\partial \boldsymbol{\mu}_j} = -\frac{1}{\sum_{j=1}^{M} \mathcal{K}(\mathbf{x}_i|\mathbf{g}_j)} \cdot \frac{\partial \mathcal{K}(\mathbf{x}_i|\mathbf{g}_j)}{\partial \boldsymbol{\mu}_j}. \tag{13}$$

Compared to $\partial \mathcal{L}_{ce}^{(i)}/\partial \boldsymbol{\mu}_j$ (Eq. (6)), $\partial \mathcal{L}_{flm}^{(i)}/\partial \boldsymbol{\mu}_j$ does not vanish when $\mathcal{K}(\mathbf{x}_i|\mathbf{g}_j) \to 1$, and therefore every $\mathbf{x}_i$ can provide relocation signals to all primitives. So, $\mathcal{L}_{flm}$ can supervise the global distribution of the primitives and enable a network to generate primitive relocation when necessary. Moreover, the computational cost of $\mathcal{L}_{flm}$ is also much less than that of $\mathcal{L}_{ce}$ because it only considers occupied regions, especially in Occ-ScanNet [44] where only 3% of voxels are occupied.

Given the formulation of superquadrics

$$\mathcal{K}_q(\mathbf{x}, \boldsymbol{\epsilon}, \mathbf{s}) = \exp\big(-\frac{1}{2} f(\mathbf{x}, \boldsymbol{\epsilon}, \mathbf{s})\big), \tag{14}$$

$$f(\mathbf{x}, \boldsymbol{\epsilon}, \mathbf{s}) = \Big(\big(\frac{x}{s_x}\big)^{\frac{2}{\epsilon_2}} + \big(\frac{y}{s_y}\big)^{\frac{2}{\epsilon_2}}\Big)^{\frac{\epsilon_2}{\epsilon_1}} + \big(\frac{z}{s_z}\big)^{\frac{2}{\epsilon_1}}, \tag{15}$$

we can further unify the closed forms of v_j for Gaussian and superquadric:

$$v_j = 2^{\frac{3\epsilon_1}{2}}\, \epsilon_1^2\, \epsilon_2\, \frac{\Gamma\big(\frac{\epsilon_2}{2}\big)^2}{\Gamma(\epsilon_2)}\, \Gamma(\epsilon_1)\, \Gamma\big(\frac{\epsilon_1}{2}\big) s_x s_y s_z, \tag{16}$$

where $\boldsymbol{\epsilon}(\epsilon_1, \epsilon_2)^\top$ and $\mathbf{s} = (s_x, s_y, s_z)^\top$ are shape parameters and scales, respectively, and $\Gamma(z) = \int_0^\infty e^{-t} t^{z-1} dt$ is the gamma function [26]. When $\epsilon_1 = \epsilon_2 = 1$, $v_j = (2\pi)^{\frac{3}{2}} s_x s_y s_z$ is the volume of the Gaussian. The derivation is provided in the supplementary material.

Voxelization. The density term in Eq. (12) naturally leads to effective and object-centric voxelization formulas for geometry and semantics:

$$p = \sum_{j=1}^{M} \mathcal{K}(\mathbf{x}|\mathbf{g}_j),\ \mathbf{c} = \frac{\sum_{j=1}^{M} \mathcal{K}(\mathbf{x}|\mathbf{g}_j)\mathbf{f}_j}{\sum_{j=1}^{M} \mathcal{K}(\mathbf{x}|\mathbf{g}_j)}. \tag{17}$$

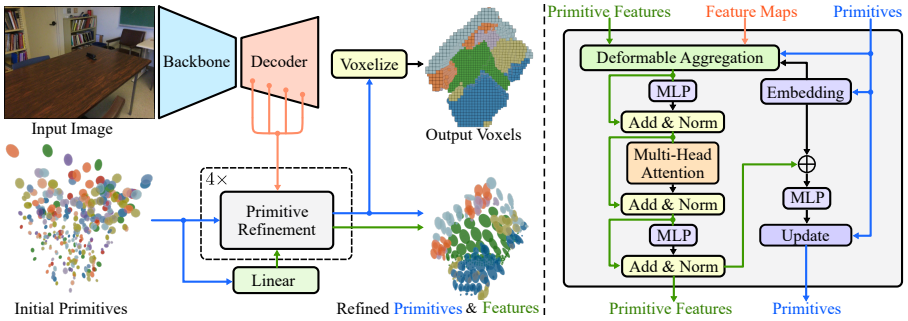

Fig. 5: Overview of our FLM-Occ (left) and the structure of the refinement block (right). FLM-Occ refines primitives to model semantic occupancy grids.

Notably, $\mathbf{c}$ is the sum of the contributions that reflect how much each primitive contributes to the semantics of $\mathbf{x}$. Although Eq. (17) seems similar to Eq. (1), the calculation of $\mathbf{c}$ in Eq. (17) only considers occupied regions while the one in Eq. (1) considers both empty and occupied regions. To achieve this, we need to use $\mathcal{L}_{flm}$ instead of $\mathcal{L}_{ce}$.

Regularization. During voxelization, a point is classified as occupied if its density exceeds a predefined threshold. Empirically, we find that most density values lie in $[0, 1]$ and a global threshold of 0.5 yields better results (see Fig. 7). Therefore, we further add a density regularization $\mathcal{L}_{reg}$ to encourage the densities to approach 1 so that we can directly set the global threshold to 0.5:

$$\mathcal{L}_{reg} = \frac{1}{P}\sum_{i=1}^{P}\Big(\sum_{j=1}^{M}\mathcal{K}(\mathbf{x}_i|\mathbf{g}_j) - 1\Big)^2. \tag{18}$$

This loss accelerates training and reduces overlap between primitives by suppressing densities that exceed one, as it facilitates the volume term in $\mathcal{L}_{flm}$.

3.3 FLM-Occ

Overview. Given an RGB image, camera intrinsics, and a fixed number of uniformly distributed primitives, FLM-Occ gradually relocates and deforms the primitives to match scene structures with the guidance of image features and outputs the corresponding occupancy grid. As shown in Fig. 5, it consists of an image encoder, a linear layer, a set of learnable primitives, four refinement blocks and a voxelization module. FLM-Occ features a significantly more streamlined and scalable architecture compared to existing methods [28, 37, 41, 44].

Image encoder. Unlike existing methods, which use Depth Anything v2 (DAv2) [42] and EfficientNet [33] to obtain geometric and semantic priors, respectively, FLM-Occ relies solely on DAv2 to extract both. Our experimental results show that this single-encoder design is sufficient.

Initial primitives. Each primitive is parameterized by a position $\boldsymbol{\mu} \in \mathbb{R}^3$, scales $\mathbf{s} \in \mathbb{R}^3$, a rotation $\mathbf{R} \in \mathbb{R}^{3\times3}$, shape parameters $\boldsymbol{\epsilon} \in \mathbb{R}^2$ and semantic logits

$\mathbf{f} \in \mathbb{R}^n$. These parameters are learnable but randomly initialized before training. Notably, this formulation subsumes standard Gaussians as a special case when $\boldsymbol{\epsilon} = (1, 1)^\top$. $[\boldsymbol{\mu}]_{xy}$ is represented in $[0, 1]^2$, $[\boldsymbol{\mu}]_z$ and $\mathbf{s}$ are represented in log space. They are transformed into Euclidean space with camera intrinsics during voxelization. Primitive features are initialized by applying a linear projection to the primitive parameters.

Refinement block. Existing methods typically constrain primitive updates to a predefined small range, necessitating careful tuning to balance training stability with the computational overhead of dense voxel grid supervision. In contrast, leveraging our proposed FLM loss, FLM-Occ eliminates these constraints, enabling unconstrained updates of raw primitive parameters $\tilde{\mathbf{g}}$ and $\tilde{\mathbf{f}}$:

$$\begin{aligned} \tilde{\mathbf{g}}^{(l+1)} &= \mathbf{d}_g^{(l+1)} + \mathbf{g}_g^{(l)}, \\ \tilde{\mathbf{f}}^{(l+1)} &= \mathbf{d}_f^{(l+1)}, \end{aligned} \tag{19}$$

where $[\mathbf{d}_g^{(l+1)}; \mathbf{d}_f^{(l+1)}]$ is predicted by a multi-layer perceptron (MLP) in the refinement block. The final primitive parameters are obtained by applying simple activation functions to $\tilde{\mathbf{g}}^{(l+1)}$ and $\tilde{\mathbf{f}}^{(l+1)}$, including sigmoid for bounded parameters and normalization for quaternions. Full formulations are provided in the supplementary material. We use deformable aggregation [20], which is commonly used for Gaussian-based occupancy prediction, to sample features for refinement. We use rotary positional encoding [31] in the multi-head self-attention [35] layers.

Voxelization. Voxels with densities larger than 0.5 will be semantically classified with softmax. We also use neighboring voxelization (Eq. (2)) for acceleration. Notably, calculating training loss in FLM does not require voxelization, and only occupied voxels are used in loss calculation.

Training objective. FLM-Occ is trained by minimizing the sum of $\mathcal{L}_{flm}$ $\big($Eq. (12)$\big)$, $\mathcal{L}_{reg}$ $\big($Eq. (18)$\big)$ and a cross entropy loss $\mathcal{L}_{ce}$ for semantics:

$$\min \mathcal{L}_{flm} + \lambda_1 \mathcal{L}_{reg} + \lambda_2 \mathcal{L}_{ce}. \tag{20}$$

4 Experiments

4.1 Dataset and Metrics

Following prior work, we evaluate our method on indoor datasets Occ-ScanNet [44] and Occ-ScanNet-mini2 [41]. Their train and test splits are 45610/19705 and 5504/2376, respectively. Each sample includes an RGB image, camera intrinsics, and a $60 \times 60 \times 36$ ground-truth voxel grid with 11 semantic classes and a voxel size of 0.08 m. We report the intersection-over-union (IoU) for geometry (occupied vs. empty) and for each semantic class, as well as the mean IoU (mIoU) across all classes, all in percentages (%).

4.2 Training Details

The loss weights λ_1 and λ_2 are set to 0.1 and 1.0, respectively. The model is optimized using AdamW [21] with $\beta_1 = 0.85$ and $\beta_2 = 0.95$. The weight decay is

Table 2: Results on Occ-ScanNet [44]. Best, second, and third results are highlighted in light red, orange, and yellow, respectively. The "Type" follows the definitions in Tab. 1. "Vox", "Que", "Gau", and "SQ" denote voxel, query, Gaussian, and superquadric, respectively.

Dataset	Method	Num.	Rep.	Type	IoU	Ceiling	Floor	Wall	Window	Chair	Bed	Sofa	Table	TVs	Furniture	Objects	mIoU
Occ-ScanNet	MonoScene [1]	129.6k	Vox	—	41.6	15.2	44.7	22.4	12.6	26.1	27.0	35.9	28.3	6.6	32.2	19.8	24.6
	ISO [44]	129.6k	Vox	—	42.2	19.9	41.9	22.4	17.0	29.0	42.4	42.0	29.6	10.6	36.4	24.6	28.7
	DiScene [18]	960	Que	—	47.2	45.3	50.6	40.4	36.7	42.3	59.7	62.0	45.6	41.2	52.4	42.7	47.2
	E.Occ [41]	16200	Gau	1	53.6	39.6	50.4	41.4	31.7	40.9	55.0	61.4	44.0	36.1	53.9	42.2	45.2
	E.Occ++ [37]	16200	Gau	1	54.9	36.4	53.1	41.8	34.4	42.9	57.3	64.1	45.2	34.8	54.2	44.1	46.2
	SplatSSC [28]	1200	Gau	2	62.8	49.1	59.0	48.3	38.8	47.4	62.4	67.0	49.5	42.6	60.7	45.4	51.8
	FLM-Occ	64	Gau	New	64.7	55.0	64.0	52.0	44.7	48.6	67.3	70.4	54.3	46.7	64.3	52.4	56.3
		32	SQ		65.3	55.1	64.5	52.2	43.3	48.3	66.5	70.6	53.5	45.6	64.4	50.7	55.9
		1024	Gau		71.2	61.9	70.0	57.5	52.6	58.1	73.5	77.5	62.7	**55.5**	70.3	59.8	63.6
		1024	SQ		**71.7**	**62.5**	**70.6**	**58.0**	**53.1**	**58.8**	**74.0**	**77.9**	**63.5**	55.2	**70.8**	**60.3**	**64.0**
Occ-ScanNet-mini2	MonoScene [1]	129.6k	Vox	—	41.9	17.0	46.2	23.9	12.7	27.0	29.1	34.8	29.1	9.7	34.5	20.4	25.9
	ISO [44]	129.6k	Vox	—	42.9	21.1	42.7	24.6	15.1	30.8	41.0	43.3	32.2	12.1	35.9	25.1	29.4
	E.Occ [41]	16200	Gau	1	55.1	29.5	49.4	41.7	36.3	41.9	60.4	59.6	46.3	34.5	58.0	43.5	45.6
	E.Occ++ [37]	16200	Gau	1	55.7	23.3	51.0	42.8	39.3	43.5	65.6	64.0	50.7	40.7	60.3	48.9	48.2
	SplatSSC [28]	1200	Gau	2	61.5	36.6	55.7	46.5	40.1	45.6	64.5	62.4	48.6	30.6	61.2	45.4	48.9
	FLM-Occ	64	Gau	New	62.1	42.4	62.1	49.3	45.5	46.8	67.4	65.9	55.3	38.4	66.3	50.9	53.6
		32	SQ		65.3	45.8	64.8	51.7	47.8	49.0	67.8	67.8	57.6	38.9	66.7	52.0	55.4
		1024	Gau		72.6	56.9	72.7	56.9	57.5	62.3	76.5	77.7	**70.4**	49.3	75.0	62.6	65.3
		1024	SQ		**72.9**	**57.9**	**72.8**	**57.4**	**58.0**	**63.1**	**76.7**	**77.8**	**70.4**	**51.0**	**75.3**	**62.8**	**65.7**

set to 0.01. The learning rate is initialized to 5×10^{-4} with a 1,000-iteration linear warmup and cosine annealing schedule, whereas the ViT of DAv2 [42] is fully fine-tuned with a learning rate of 5×10^{-5}. We train FLM-Occ with `BFloat16` for 30,000 iterations on Occ-ScanNet [44] and 15,000 iterations on Occ-ScanNet-mini2 [41], respectively, on four RTX A6000 GPUs with a batch size of 128. Random horizontal flipping and color jittering are used for data augmentation.

4.3 Quantitative Comparison

We evaluate FLM-Occ and compare it with recent methods [1, 28, 37, 41, 44]. As reported in Tab. 2, FLM-Occ achieves state-of-the-art performance, significantly outperforming existing voxel-based and Gaussian-based methods. Specifically, with only 1024 superquadrics, FLM-Occ reaches an mIoU of 64.0% on Occ-ScanNet, representing a substantial improvement of +12.2% over the previous best-performing method, SplatSSC (51.8%), despite using fewer primitives. Morever, when the primitive count is reduced to a mere 32 and 64, FLM-Occ still achieves 55.9% and 56.3% mIoU, already surpassing SplatSSC (1200 Gaussians) [28] and EmbodiedOcc (16200 Gaussians) [41]. These results suggest that MLE is naturally well aligned with mixture-model-based occupancy prediction.

In terms of implementation, FLM-Occ offers greater simplicity and flexibility compared to existing methods. While SplatSSC and EmbodiedOcc rely on

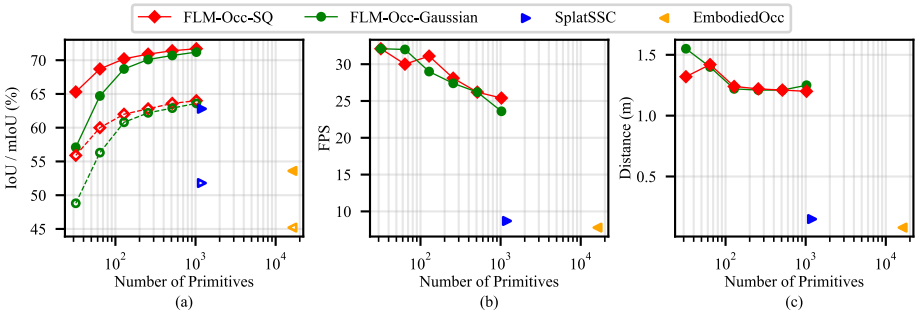

Fig. 6: Performance vs. Representation Size. Filled and hollow markers in (a) indicate IoU and mIoU, respectively. FPS is measured on a single RTX 3090 GPU.

a two-stage pipeline to fine-tune a depth estimation branch for Gaussian initialization, FLM-Occ learns the initial primitives in a single end-to-end process. Furthermore, unlike prior works that utilize EfficientNet [33] and DAv2 [42] to obtain semantic and geometric priors, respectively for guiding the refinement, we employ features from a single DAv2 [42] and obtain more accurate results. This observation is consistent with the findings in DAv2, suggesting that representations learned for depth estimation can generalize effectively to semantic occupancy tasks with fine-tuning.

4.4 Ablation on Primitives

We train FLM-Occ with different primitives (Gaussian vs. superquadric) and different numbers of primitives (32,64,128,256,512,1024) to evaluate how primitives affect the results on Occ-ScanNet [44]. We also calculate the average distance between the initialized and final primitives. The results are shown in Fig. 6.

Despite using far fewer primitives, FLM-Occ achieves higher accuracy and efficiency than prior SoTA. SplatSSC [28] uses 1,200 Gaussians and achieves 62.8% IoU and 51.8% mIoU at 8.7 FPS, whereas our model with only 512 superquadrics reaches 71.4% IoU and 63.6% mIoU at 26.2 FPS. EmbodiedOcc [41] relies on 16,200 Gaussians but achieves only 53.6% IoU and 45.2% mIoU at 7.8 FPS. These results demonstrate the efficiency of our FLM formulation.

Increasing the number of primitives improves accuracy while reducing inference speed. However, the improvement gradually saturates after 256 primitives, suggesting that a few hundred primitives are sufficient to represent the scene. Superquadrics consistently outperform Gaussians across all primitive counts, especially when less primitives are used. As the primitive count increases, the performance gap between the Gaussian and superquadric gradually narrows.

The distances between initial and final positions of the primitives generally decrease as the primitive count increases, indicating smaller adjustments during optimization. The distances (around 1.2 m) of FLM-Occ are significantly larger than those of SplatSSC [28] (0.15 m) and EmbodiedOcc [41] (0.08 m), since

Table 3: Performance with different network configurations.

	SQ Count	Refine	DAv2 [42]	ViT [27]	Fintune ViT	$\mathcal{L}_{reg}$	IoU	mIoU
1)	128	4	Rel	Base			52.4	43.9
2)		4	Metric	Base			57.3	48.9
3)		4	Rel	Base	✓		67.2	59.2
4)		4	Metric	Base	✓		67.5	59.6
5)		2	Metric	Base	✓	✓	69.2	60.7
6)		3	Metric	Base	✓	✓	69.8	61.5
7)		4	Metric	Base	✓	✓	70.2	62.0
8)		5	Metric	Base	✓	✓	70.0	61.8
9)		**4**	**Rel**	**Large**	✓	✓	**72.7**	**65.0**
10)	1024	4	Metric	Base	✓	✓	71.7	64.0

those methods initialize Gaussians using depth maps or use a large number of primitives, while our primitives are uniformly sampled in the space. This demonstrates that FLM-Occ learns to substantially relocate the primitives.

4.5 Ablation Study on Network Configuration

We utilize FLM-Occ with 128 superquadrics trained on Occ-ScanNet [44], as this configuration represents the elbow point of the performance-primitive curve in Fig. 6, suggesting it is more sensitive to parameter variations.

In Tab. 3, the results show that metric depth features improve the performance. Both IoU and mIoU increases 5 points by replacing the relative DAv2 [42] with the metric one (row 1 vs. 2). This explains why previous methods [28,37,41] choose to further finetune the metric DAv2 [27] on the ScanNet dataset [4]. With the ViT [27] finetuned, the performance significantly improves, especially when using relative DAv2 features (row 1 vs. 3), where the mIoU increases from 43.9% to 59.2% (+15.3%). This compares to a gain of 10.7% for metric features (row 2 vs. 4). These results suggest that while metric-scale features provide a stronger zero-shot prior for 3D occupancy, task-specific finetuning effectively mitigates the scale ambiguity inherent in relative features, allowing them to achieve comparable performance (row 3 vs. 4). The performance improves with more refinement blocks but saturates at four blocks (rows 5–8). Furthermore, scaling the encoder to DAv2-ViT-L yields additional gains, achieving higher results of 72.7% IoU and 65.0% mIoU (row 8 vs. 9). The density regularization loss $\mathcal{L}_{reg}$ (Eq. (18)) improves the performance (row 4 vs. 7) as we assume that densities lie within $[0, 1]$ and set a threshold of 0.5 for occupancy.

4.6 Influence of Density Threshold for Occupancy

Although the density regularization loss encourages densities of occupied voxels to approach 1, some still exceed 1. Therefore, we analyze the empirical distribution of predicted densities by FLM-Occ with 128 superquadrics trained on Occ-ScanNet [4] and evaluate how different occupancy thresholds influence IoU and mIoU. As shown in Fig. 7, setting the threshold to 0.5 yields better results.

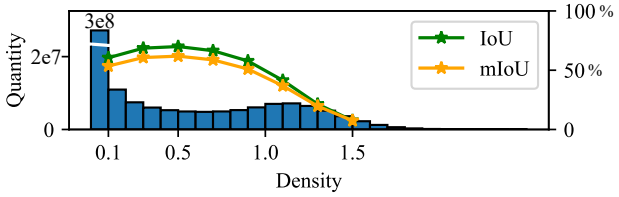

Fig. 7: Distribution of density and IoUs with different density thresholds.

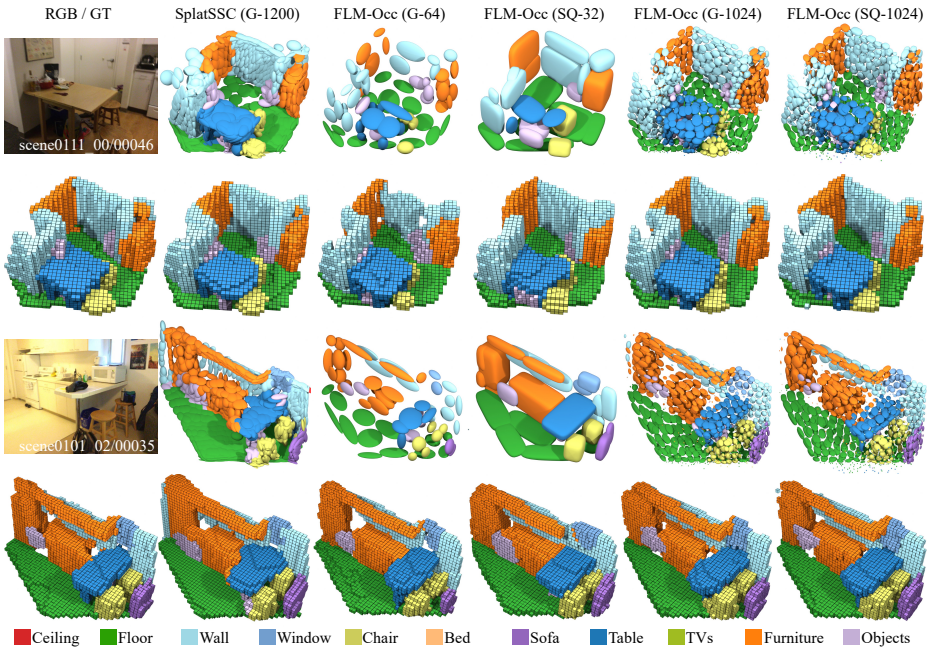

Fig. 8: Comparisons on Occ-ScanNet dataset [44]

4.7 Qualitative Results

Fig. 8 illustrates the qualitative results of SplatSSC (1200 Gaussians) [28] alongside FLM-Occ under four primitive configurations (64/1024 Gaussians and 32/1024 superquadrics). Due to the relatively coarse annotation granularity (0.08m), all methods effectively capture the scene structure. However, regarding primitive distribution, SplatSSC exhibits significant inter-primitive overlaps, as its geometric aggregation formula incentivizes such redundancy. In contrast, FLM-Occ produces sparser representations, a direct result of our MLE-based formulation and density regularization, which encourages a more efficient spatial distribution.

5 Conclusion

We propose FLM-Occ—Feed-forward Likelihood Maximization for efficient indoor occupancy prediction—a novel method that reformulates occupancy prediction not as voxel classification but as voxel distribution estimation, parameterized by a mixture model. To leverage MLE for end-to-end training, we propose FLM loss, a novel loss that explicitly drives the predicted primitives to match objects. We also propose probabilistic voxelization formulas to convert a mixture model to an occupancy grid. Built upon FLM, FLM-Occ is able to substantially relocate and deform geometric primitives to represent a scene. Compared to the prior state-of-the-art method, FLM-Occ leverages 20× fewer primitives to achieve higher accuracy and over 3× speed up on Occ-ScanNet datasets.

6 Limitations

The computational cost of the FLM loss scales with both the number of occupied voxels and primitives, which may limit its applicability to large-scale scenes. In addition, similar to EmbodiedOcc [41] and SplatSSC [28], FLM-Occ relies on a fixed number of primitives, so changing the primitive count requires retraining the network. Incorporating learnable pruning and splitting mechanisms, as in sparse voxel-based methods [13, 22], is a promising direction. Finally, since the FLM formulation may lead to less distinct occupied-region boundaries, we use a global density threshold to infer occupancy. Learned or adaptive thresholding could further improve the modeling of occupancy decision boundaries.

Acknowledgements

This work was supported in part by Shenzhen Science and Technology Program (No. SGDX20240115111759002), in part by Meituan Academy of Robotics Shenzhen, in part by the Shenzhen Association for Science and Technology (No. XHXS2025-003), in part by high level of special funds (G03034K003) from Southern University of Science and Technology, Shenzhen, China, and in part by the Open Research Fund (GML-KF-24-15) from Guangdong Laboratory of Artificial Intelligence and Digital Economy.

References

1. Cao, A.Q., De Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3991–4001 (2022)
2. Charatan, D., Li, S.L., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19457–19467 (2024)

3. Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In: European Conference on Computer Vision. pp. 370–386. Springer (2024)
4. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
5. Fang, L., Hu, X., Zou, Y., Zhang, H.: Cogstereo: Neural stereo matching with implicit spatial cognition embedding. In: 2026 IEEE International Conference on Robotics and Automation (ICRA). IEEE
6. Gan, W., Liu, F., Xu, H., Mo, N., Yokoya, N.: Gaussianocc: Fully self-supervised and efficient 3d occupancy estimation with gaussian splatting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 28980–28990 (2025)
7. Hou, J., Li, X., Guan, W., Zhang, G., Feng, D., Du, Y., Xue, X., Pu, J.: Fastocc: Accelerating 3d occupancy prediction by fusing the 2d bird's-eye view and perspective view. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 16425–16431. IEEE (2024)
8. Huang, R., Mikolajczyk, K.: No pose at all: Self-supervised pose-free 3d gaussian splatting from sparse views. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 27947–27957 (2025)
9. Huang, Y., Thammatadatrakoon, A., Zheng, W., Zhang, Y., Du, D., Lu, J.: Gaussianformer-2: Probabilistic gaussian superposition for efficient 3d occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 27477–27486 (June 2025)
10. Huang, Y., Zheng, W., Zhang, B., Zhou, J., Lu, J.: SelfOcc: Self-Supervised Vision-Based 3D Occupancy Prediction . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 19946–19956. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2024). https://doi.org/10.1109/CVPR52733.2024.01885, https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01885
11. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9223–9232 (2023)
12. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision – ECCV 2024. pp. 376–393. Springer Nature Switzerland, Cham (2025)
13. Iwase, S., Liu, K., Guizilini, V., Gaidon, A., Kitani, K., Ambruş, R., Zakharov, S.: Zero-shot multi-object scene completion. In: European Conference on Computer Vision. pp. 96–113. Springer (2024)
14. Jevtić, A., Reich, C., Wimbauer, F., Hahn, O., Rupprecht, C., Roth, S., Cremers, D.: Feed-forward SceneDINO for unsupervised semantic scene completion. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2025)
15. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph. **42**(4), 139–1 (2023)
16. Li, H., Hou, Y., Xing, X., Ma, Y., Sun, X., Zhang, Y.: Occmamba: Semantic occupancy prediction with state space models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 11949–11959 (2025)
17. Li, J., Lu, M., Liu, J., Wang, H., Gu, C., Zheng, W., Du, L., Zhang, S.: Sliceocc: Indoor 3d semantic occupancy prediction with vertical slice representation. In:

2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 15762–15768. IEEE (2025)
18. Li, X., Zheng, Y., Li, P., Chen, Y., Zhang, Y.Q., Ding, W.: Enhancing indoor occupancy prediction via sparse query-based multi-level consistent knowledge distillation. IEEE Robotics and Automation Letters **10**(11), 11690–11697 (2025). https://doi.org/10.1109/LRA.2025.3615532
19. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
20. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d: Multi-view 3d object detection with sparse spatial-temporal fusion (2022)
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
22. Lu, Y., Zhu, X., Wang, T., Ma, Y.: Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries. Advances in Neural Information Processing Systems **37**, 79618–79641 (2024)
23. Matsuki, H., Murai, R., Kelly, P.H., Davison, A.J.: Gaussian splatting slam. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18039–18048 (2024)
24. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
25. Myung, I.J.: Tutorial on maximum likelihood estimation. Journal of mathematical Psychology **47**(1), 90–100 (2003)
26. National Institute of Standards and Technology: Gamma function (2025), https://dlmf.nist.gov/5, nIST Digital Library of Mathematical Functions, Release 1.1.10
27. Oquab, M., Darcet, T., Moutakanni, T., Vo, H.V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Howes, R., Huang, P.Y., Xu, H., Sharma, V., Li, S.W., Galuba, W., Rabbat, M., Assran, M., Ballas, N., Synnaeve, G., Misra, I., Jegou, H., Mairal, J., Labatut, P., Joulin, A., Bojanowski, P.: Dinov2: Learning robust visual features without supervision (2023)
28. Qian, R., Cao, H., Deng, T., Yuan, S., Xie, L.: Splatssc: Decoupled depth-guided gaussian splatting for semantic scene completion. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 40, pp. 8520–8528 (2026)
29. Shi, D., Wang, W., Chen, D.Y., Zhang, Z., Bian, J.W., Zhuang, B.: Revisiting depth representations for feed-forward 3d gaussian splatting. In: 2026 International Conference on 3D Vision (3DV). pp. 1071–1080. IEEE (2026)
30. Song, S., Yu, F., Zeng, A., Chang, A.X., Savva, M., Funkhouser, T.: Semantic scene completion from a single depth image. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1746–1754 (2017)
31. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing **568**, 127063 (2024)
32. Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10208–10217 (2024)
33. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International conference on machine learning. pp. 10096–10106. PMLR (2021)

34. Tang, P., Wang, Z., Wang, G., Zheng, J., Ren, X., Feng, B., Ma, C.: Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017)
36. Veicht, A., Hong, S., Baráth, D., Pollefeys, M.: Zipsplat: Fewer gaussians, better splats. arXiv preprint arXiv:2606.05102 (2026)
37. Wang, H., Wei, X., Zhang, X., Li, J., Bai, C., Li, Y., Lu, M., Zheng, W., Zhang, S.: Embodiedocc++: Boosting embodied 3d occupancy prediction with plane regularization and uncertainty sampler. In: Proceedings of the 33rd ACM International Conference on Multimedia. pp. 925–934 (2025)
38. Wang, J., Liu, Z., Meng, Q., Yan, L., Wang, K., Yang, J., Liu, W., Hou, Q., Cheng, M.: Opus: occupancy prediction using a sparse set. In: Advances in Neural Information Processing Systems (2024)
39. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17850–17859 (2023)
40. Wimbauer, F., Yang, N., Rupprecht, C., Cremers, D.: Behind the scenes: Density fields for single view reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9076–9086 (2023)
41. Wu, Y., Zheng, W., Zuo, S., Huang, Y., Zhou, J., Lu, J.: Embodiedocc: Embodied 3d occupancy prediction for vision-based online scene understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 26360–26370 (October 2025)
42. Yang, L., Kang, B., Huang, Z., Zhao, Z., Xu, X., Feng, J., Zhao, H.: Depth anything v2. Advances in Neural Information Processing Systems **37**, 21875–21911 (2024)
43. Yao, J., Li, C., Sun, K., Cai, Y., Li, H., Ouyang, W., Li, H.: Ndc-scene: Boost monocular 3d semantic scene completion in normalized device coordinates space. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9421–9431. IEEE Computer Society (2023)
44. Yu, H., Wang, Y., Chen, Y., Zhang, Z.: Monocular occupancy prediction for scalable indoor scenes. In: Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXX. p. 38–54. Springer-Verlag, Berlin, Heidelberg (2024). https://doi.org/10.1007/978-3-031-73404-5_3, https://doi.org/10.1007/978-3-031-73404-5_3
45. Yu, Z., Shu, C., Deng, J., Lu, K., Liu, Z., Yu, J., Yang, D., Li, H., Chen, Y.: Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin (2023)
46. Zhang, G., Fan, L., He, C., Lei, Z., Zhang, Z., Zhang, L.: Voxel mamba: Group-free state space models for point cloud based 3d object detection. Advances in Neural Information Processing Systems **37**, 81489–81509 (2024)
47. Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: Gs-lrm: Large reconstruction model for 3d gaussian splatting. In: European Conference on Computer Vision. pp. 1–19. Springer (2024)
48. Zhu, Z., Wang, S., Xie, J., Liu, J.J., Wang, J., Yang, J.: Voxelsplat: Dynamic gaussian splatting as an effective loss for occupancy and flow prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2025), to appear

49. Zuo, S., Zheng, W., Han, X., Yang, L., Pan, Y., Lu, J.: Quadricformer: Scene as superquadrics for 3d semantic occupancy prediction. In: Advances in Neural Information Processing Systems (2025)
50. Zuo, S., Zheng, W., Huang, Y., Zhou, J., Lu, J.: Gaussianworld: Gaussian world model for streaming 3d occupancy prediction. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6772–6781 (2025)