OccFusion: Depth Estimation Free Multi-sensor Fusion for 3D Occupancy Prediction

Ji Zhang[0009−0000−1759−5184]⋆, Yiran Ding[0009−0005−8624−8545]⋆, and Zixin Liu[0009−0003−2353−7847]

Wuhan University, Hubei, China, 430072
{jizhang, yrding, liuzixin}@whu.edu.cn

Abstract. 3D occupancy prediction based on multi-sensor fusion, crucial for a reliable autonomous driving system, enables fine-grained understanding of 3D scenes. Previous fusion-based 3D occupancy predictions relied on depth estimation for processing 2D image features. However, depth estimation is an ill-posed problem, hindering the accuracy and robustness of these methods. Furthermore, fine-grained occupancy prediction demands extensive computational resources. To address these issues, we propose OccFusion, a depth estimation free multi-modal fusion framework. Additionally, we introduce a generalizable active training method and an active decoder that can be applied to any occupancy prediction model, with the potential to enhance their performance. Experiments conducted on nuScenes-Occupancy and nuScenes-Occ3D demonstrate our framework's superior performance. Detailed ablation studies highlight the effectiveness of each proposed method.

Keywords: 3D feature learning · 3D occupancy prediction · Multi-modal learning · Depth estimation free · Multi-sensor fusion

1 Introduction

Accurate and complete perception of 3D surroundings in urban contexts is crucial for autonomous driving, facilitating tasks such as map construction and vehicle motion planning, thereby ensuring safe and reliable driving. Recent years have seen a surge in research on semantic occupancy perception [1, 9, 13, 16, 48, 51]. Unlike 3D object detection [5,24,30,57], which typically employs bounding boxes to approximate the location of dynamic objects, semantic occupancy perception models the entire sensor field, encompassing static objects and areas beyond the immediate interest. This approach yields finer-grained 3D scene representations, aligning more closely with real-world driving scenarios, making it a promising research direction.

In previous works on semantic surrounding perception [4,16,21,33,34,39,41, 46,48,49,51,56], converting 2D features to 3D through depth prediction has been a conventional approach [4, 21, 33, 41, 48, 49, 56]. However, it is widely recognized

⋆ These authors contributed equally.

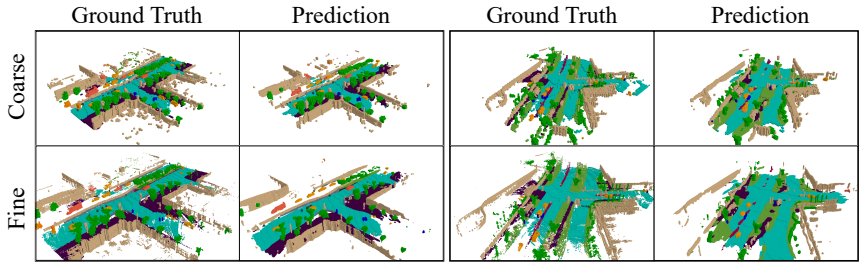

Fig. 1: Visualization of our coarse-grained and fine-grained prediction results. The first row shows the ground truth and prediction for two coarse-grained samples, while the second row displays the ground truth and prediction for the same two samples at a fine-grained level. Better viewed when zoomed in.

that lifting 2D image features to 3D [38] inherently attempts to solve an ill-posed problem. The robustness of depth estimation cannot be guaranteed, and considering its use in downstream tasks, the instability of depth estimation poses significant risks to various driving tasks [24].

By employing multi-modal methods, depth information can be introduced through LiDAR data, mitigating the ill-posed nature of the problem. However, the challenge remains in effectively integrating 2D image features with 3D LiDAR features without depth estimation. While previous literature [3,17] has indicated that the fusion of multi-modal data can provide redundancy and higher accuracy, to date, only a few studies have focused on multi-modal 3D semantic occupancy prediction [36,48], and these methods have relied on depth estimation for image features, resulting in suboptimal robustness and accuracy (see Fig. 2).

On the other hand, representative fusion-based occupancy prediction methods [36,48] are based on the CONet (**C**ascade **O**ccupancy **Net**work) architecture, which refines coarse-grained voxels, improving precision while conserving computational resources. However, we point out that splitting operations for most high-confidence voxels are unnecessary and only increase computational load. Additionally, existing models use specific loss functions to address voxel class imbalance from a micro perspective [36, 45, 48] but overlook the long-tail effect of training data scenes from a macro perspective. New methods are needed to enable the model to selectively learn from more challenging samples, thereby enhancing robustness.

We propose the OccFusion framework, which eliminates depth estimation for image features. Unlike previous methods that blend image features into point cloud features and suffer from density discrepancies between camera and LiDAR features [22, 47, 55], our OccFusion method uses preprocessed LiDAR points to sample image features. Specifically, we voxelize the space around the vehicle and preprocess each voxel's point cloud: for voxels with sparse LiDAR points, we uniformly generate synthetic point clouds; for voxels with dense LiDAR points, we use the farthest point sampling algorithm [40] to select a subset of points. Then, we project the point cloud onto the image using camera intrinsics and

extrinsics to establish correspondences between 2D camera features and 3D LiDAR features. We use respective encoders to obtain 3D and 2D features, and then perform deformable cross attention [61], using LiDAR voxel features concatenated with point coordinates as queries and corresponding camera features as keys, directly fusing 3D LiDAR and 2D camera features. The features of each LiDAR point within a voxel are averaged to obtain the selected camera feature for that voxel, which is then added to the corresponding LiDAR voxel feature before normalization, resulting in the multi-modal feature. To achieve fine-grained results, we improve upon CONet [48] by introducing an active occupancy decoder, which selectively splits challenging voxels to learn fine-grained features, significantly reducing model complexity. Finally, we propose an active training method, allowing the model to prioritize learning from more difficult samples. Experiments show that this simple strategy further improves model performance and can be generalized to the training of other models.

Our contributions can be summarized as follows:

- We introduce a novel point-to-point multi-modal feature fusion framework for 3D occupancy prediction, OccFusion, which eliminates the need for depth estimation of image features during the fusion process.
- We propose an efficient point cloud preprocessing algorithm that produces denser and more uniform point clouds, which, as demonstrated by experiments, significantly enhances feature fusion.
- We propose an active occupancy decoder and an active training method, both of which can be naturally transferred to other occupancy models to improve their performance.
- Experiments on nuScenes show that our method achieves state-of-the-art (or comparable) performance across all categories, with significant improvements in the accuracy of small objects. Detailed ablation studies validate the effectiveness of each proposed module and method.

2 Related Work

2.1 Vision-Based 3D Occupancy Prediction

Effectively representing the 3D environment around the vehicle remains a core issue in autonomous driving. Voxel-based representation discretizes 3D space into a voxel grid, computing features for each voxel in the grid to represent the scene. This method provides finer granularity features than BEV (**B**ird's **E**ye **V**iew) based methods [24, 30, 38, 54, 59], aligning more closely with real-world driving scenarios. The lack of direct geometric inputs and localization information [3] makes purely camera-based 3D occupancy prediction [4, 16] challenging. Recent works [4, 21, 26, 33, 48, 58] have utilized depth prediction methods to generate occupancy features. However, depth prediction is notoriously ill-posed, and these methods often suffer from unstable depth estimation. While camera-only approaches offer promising prospects, multi-modal methods deliver higher accuracy and more reliable results, crucial for the safe and trustworthy deployment of autonomous driving.

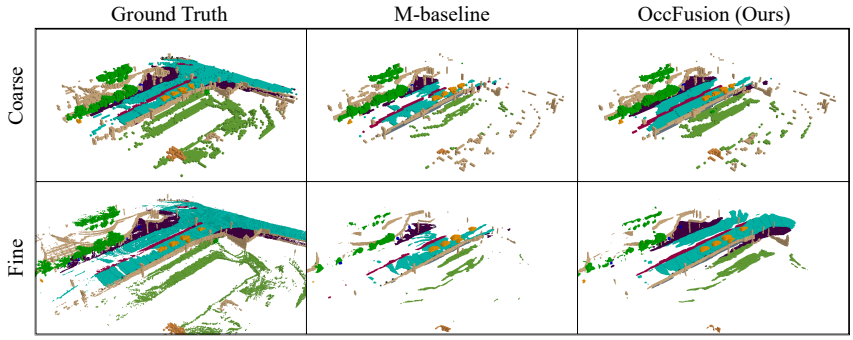

Fig. 2: Comparison of our method with one of the existing SOTA multi-modal baseline [48] under challenging samples. The first row compares M-baseline [48] with the proposed method for the coarse occupancy prediction task, while the second row compares M-CONet [48] with our OccFusion at the fine stage. Better viewed when zoomed in.

2.2 Feature Fusion of Camera and LiDAR

LiDAR provides precise localization and reflectance information, complementing camera features [3,31,36,48]. However, LiDAR point clouds are often sparse and vary greatly in density, and lack detailed semantic information such as color and object edges [30]. Despite the greater expense associated with using multiple sensors, multi-modal semantic occupancy prediction methods [36, 48] can integrate the strengths of both LiDAR and camera, outperforming methods based solely on either. However, existing multi-modal approaches face challenges in multi-channel fusion, current works [31,36,48] still rely on depth estimation to extract image features, which is an inefficient method. In contrast, our innovative fusion approach does not estimate depth but directly integrates camera and LiDAR features, ensuring efficiency and robustness.

2.3 Active Learning and Hard Example Mining

The imbalance of spatial semantic categories and variability in samples can impact model training, hence, random sample selection may degrade accuracy [29]. Inspired by previous active learning research [10, 12, 19], which suggests that samples with high entropy require additional attention, our active occupancy decoder refines features only for coarse voxels with the highest entropy. During training, we draw inspiration from hard example mining [8, 43, 44], prioritizing samples with greater uncertainty. Experiments demonstrate that this approach significantly improves model accuracy.

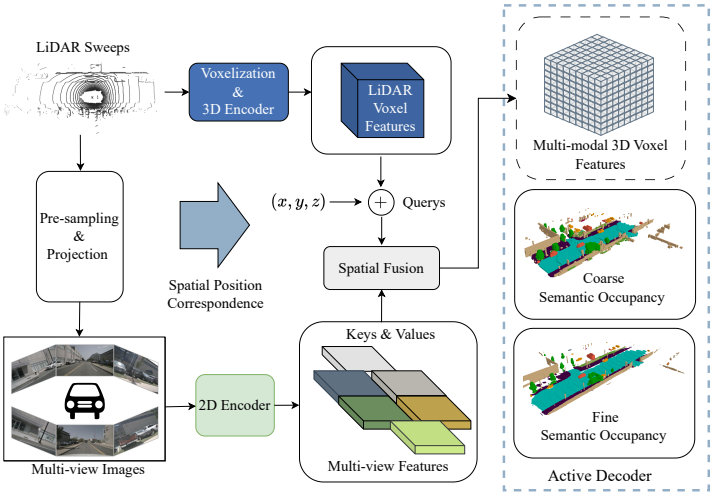

Fig. 3: The overall architecture of our method. Raw LiDAR points are processed by a 3D encoder to extract voxelized features, which, concatenated with point coordinates, serve as queries. Multi-view image features, obtained directly through a 2D encoder from surround-view images, act as keys. Enhanced point clouds are then subjected to point-to-point fusion, resulting in multi-modal 3D voxel features. An active decoder adaptively refines predictions in challenging areas.

3 Method

3.1 Overview

Figure 3 illustrates the architecture of our method. We employ VoxelNet [60] and 3D sparse convolutions [53] to embed raw LiDAR points into voxelized features $F^L \in R^{\frac{D}{S} \times \frac{H}{S} \times \frac{W}{S} \times C}$(where S is the stride). For camera images, we use ResNet101 [11] with FPN [27] as the backbone to extract multi-view features $F^{mv} \in R^{N \times H^c \times W^c \times C}$, without performing any depth-related operations. With the voxel grid established, the initial state's point cloud is considered to reside within these voxels. Due to the sparsity of raw point clouds, to effectively sample image features, we employ specific sampling and generating methods (see Sec. 3.2) to ensure each voxel contains dense and relatively uniform LiDAR points. These LiDAR points are then projected onto images using camera intrinsic and extrinsic parameters, creating reference points. Using LiDAR points as intermediaries, we establish the correspondence between LiDAR features and camera features. Through spatial fusion (see Sec. 3.3), we fuse 3D LiDAR features concatenated with point coordinates (as queries) with 2D image features (as keys) on a point-to-point basis. For each LiDAR point, we obtain a feature, and by averaging these features within each voxel, we derive a C-dimensional feature for that voxel, which is then added to and normalized with the corresponding LiDAR feature. These features can directly yield coarse occupancy

predictions through a simple classification head. Due to the high computational complexity of directly predicting refined occupancy grids, we apply an active coarse to fine pipeline to our obtained coarse-grained multi-modal occupancy features, focusing fine-grained prediction only on voxels with the highest uncertainty (see Sec. 3.4). During the training phase, we experiment with enabling the model to actively learn from samples (see Sec. 3.5), which, as experiment shows (see Sec. 4.3), can further improve model accuracy.

Algorithm 1 LiDAR Points Sampling Algorithm

Input Raw LiDAR Point Clouds
Output 3D Reference Points
Require $N_p^V \geq 0$, $\tau \in \mathbb{N}$, $\theta \in \mathbb{N}$ $(\theta > \tau)$
for each voxel V **do**
  if $N_p^V \leq \tau$ **then**
    Uniformly generate up to θ points
  else if $\tau < N_p^V \leq \theta$ **then**
    continue
  else
    Initialize $S \leftarrow \{P_0\}$ with P_0 randomly chosen
    repeat
      $P_j \leftarrow \arg\max_{P \notin S} d(P, S)$
      $S \leftarrow S \cup \{P_j\}$
    until $|S| = \theta$
  end if
end for

3.2 3D LiDAR Feature Extraction and LiDAR Point Sampling Algorithm

The method for embedding raw LiDAR points into 3D voxelized features is consistent with [48]. In this process, 3D space is partitioned into a grid of size $D/S \times H/S \times W/S$ (where S is the stride). After partitioning the space, the number of LiDAR points in voxel V is denoted as N_p^V. We define two hyper-parameters: $\tau \in \mathbb{N}$ and $\theta \in \mathbb{N}$ $(\theta > \tau)$. For each voxel, there are three possible scenarios. First, due to the sparsity of LiDAR point clouds, many voxels contain no or few LiDAR points (*i.e.*, $N_p^V \leq \tau$); for these, we generate synthetic point clouds using a simple uniform generation method to increase the point count to θ. For voxels with an adequate number of LiDAR points (*i.e.*, $\tau < N_p^V \leq \theta$), no action is taken. The uneven spatial distribution of LiDAR point clouds results in some voxels containing too many LiDAR points (*i.e.*, $N_p^V > \theta$); for these, we use farthest point sampling (FPS) [40] to select θ points. Specifically, we start with a randomly chosen point P_0 as the initial point, forming a sample set $S = \{P_0\}$. We define the distance from a point P to the set as $d(P, S) := \min d(P, P_i), P_i \in S$,

calculate the distance $d(P_j, S)$ for all points other than P_0, find

$$\alpha = \underset{j}{argmax}\, d(P_j, S), P_j \notin S, \tag{1}$$

and add P_α to the set S, repeating this process until θ points are obtained. This point cloud sampling algorithm yields denser and more uniformly distributed point clouds in each voxel, facilitating effective sampling of image features.

3.3 Camera Feature Extraction and OccFusion: Point-to-Point Multi-modal Feature Fusion

We use a 2D encoder [11,27] to extract multi-view image features without elevating them to 3D. Instead, we introduce the OccFusion module to fuse 2D image features with 3D LiDAR features on a point-to-point basis. We project preprocessed point clouds onto multi-view images using camera parameters, establishing reference points. For each LiDAR point, we concatenate its coordinates with the corresponding LiDAR voxel feature to create a query. This query allows us to sample and fuse the relevant image features via deformable attention [61], with bilinear interpolation employed for accurate sampling at designated positions. The deformable attention mechanism and the associated process can be formalized as follows:

$$DA(z_q, p_q, x) = \sum_{m=1}^{N_{head}} W_m \sum_{k=1}^{N_{key}} A_{mqk} \cdot W_m^{'} x(p_q + \Delta p_{mqk}) \tag{2}$$

and

$$OccFuse(Q, X, V) = Norm(Q_V + \frac{1}{|V|} \sum_{l \in V} \frac{1}{|\mathcal{P}(l)|} \sum_{i \in \mathcal{P}(l)} DA(Q_V^l, i, X_i)). \tag{3}$$

In this context, Q represents the 3D LiDAR voxel features, and X is the 2D feature map of surround-view images. The result on the left side of Eq. (3) corresponds to the final feature F^V for a given voxel V, where l is a 3D reference point within V, and $\mathcal{P}(\cdot)$ denotes the projection from the LiDAR coordinate system to the image coordinate system. $\mathcal{P}(l)$ is the set of reference points corresponding to the LiDAR point l after projection. Notably, due to the shared field of view among cameras, a single LiDAR point may correspond to multiple reference points across several images upon projection (see Fig. 4). Moreover, a voxel always contains $|V| \in (\tau, \theta]$ reference points after pre-sampling (see Algorithm 1), and $|\mathcal{P}(l)|$ represents the number of reference points corresponding to a single LiDAR point l. We apply averaging to address these one-to-many relationships, then add the sampled feature to the corresponding LiDAR voxel feature and normalize the result, yielding a single feature vector F^V for each voxel. Q_V^l is the query corresponding to voxel V and 3D reference point l, and X_i is the feature map of the image containing 2D reference point i. W_m and $W_m^{'}$ are learnable parameters, A_{mqk} denotes the attention weight. In the prediction phase, the derived feature F^V is processed through a classification head, enabling direct coarse semantic occupancy prediction, see Fig. 4.

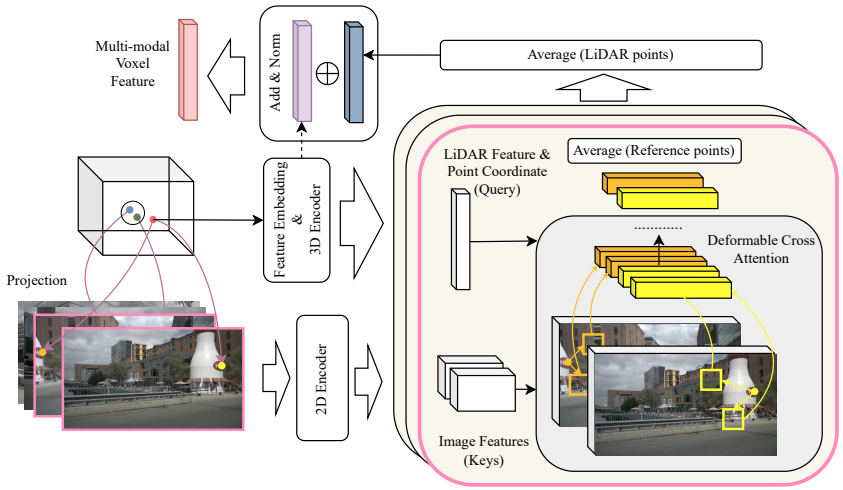

Fig. 4: Details of the OccFusion module. After pre-sampling, 3D reference points are projected onto images as 2D reference points. Synthetic point clouds (*points within the circle*) do not contribute to LiDAR feature generation. Due to overlapping camera fields of view, a single 3D reference point may correspond to multiple projected 2D reference points. Features for these reference points are averaged to derive features for each 3D point, which are then averaged to obtain the sampled camera feature. Finally, this feature is combined with the corresponding LiDAR feature through addition and normalization to produce the multi-modal feature.

3.4 Active Coarse to Fine Pipeline

CONet (**C**ascade **O**ccupancy **N**etwork) conserves computational resources by refining coarse occupancy instead of directly predicting refined occupancy features [36, 48]. In real-world scenarios, refining features for most voxels is unnecessary: for large objects like buses, coarse voxels within the occupied space can effectively represent the category using coarse occupancy features. In contrast, small objects such as traffic cones and bicycles require finer-grained predictions due to the sparsity of the coarse grid. To optimize resource use while enhancing recognition of small or overlapping objects, we introduce an entropy filter to determine the necessity of feature refinement for each voxel. Specifically, we employ classical information entropy to assess the need for fine-grained prediction in a voxel. We set a threshold δ representing the proportion of voxels requiring fine-grained feature prediction, and after passing coarse voxel features through a classification head, we obtain probabilities for each class in a voxel V denoted as $p_i^V, i = 1, \ldots, N_{class}$, where N_{class} is the total number of classes. Using the formula

$$\text{Entropy}(V) = -\sum_{i=1}^{N_{class}} p_i^V \log p_i^V, \tag{4}$$

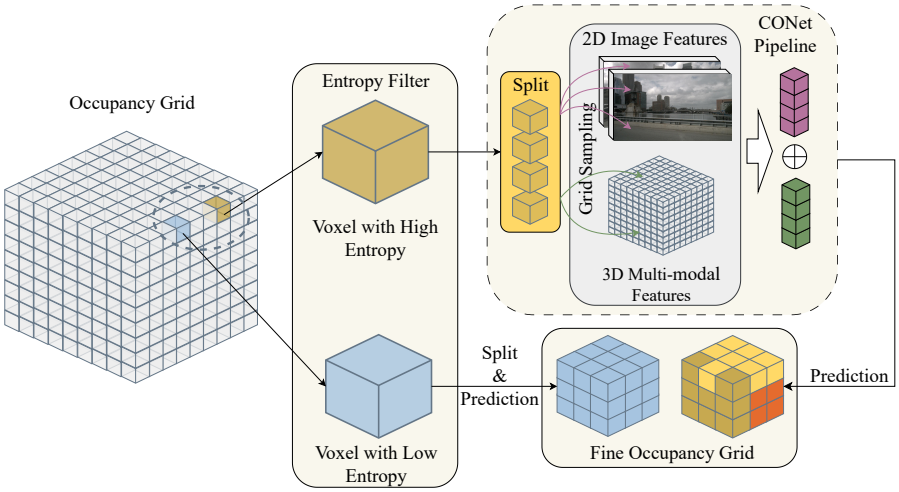

Fig. 5: Active coarse to fine pipeline. We refine features only for voxels with greater uncertainty.

we calculate the uncertainty within each coarse-grained voxel. When Entropy(V) is sufficiently high, we predict further fine-grained features for the voxel, *i.e.*, splitting the voxel into smaller voxels to serve as occupancy queries for feature sampling through corresponding 2D image features and 3D multi-modal features using camera intrinsic and extrinsic parameters. The details are the same as the coarse to fine pipeline in [48]. For voxels with lower entropy, we still split them, but for each smaller voxel, we directly use the category obtained from coarse-grained features, see Fig. 5. Note that our pipeline can naturally be extended to other occupancy prediction models, not just multi-modal models.

3.5 Active Training Method

Due to class imbalance among voxels within each sample, nearly all 3D occupancy prediction methods employ loss functions such as focal loss [28], OHEM loss [43] and semantic mIoU [4] to enhance model performance on minority semantic classes. However, we note that the complexity of different scenes varies due to factors such as lighting, weather, and surrounding environments. Inspired by classical hard example mining techniques [8,43,44], we designed an extremely simple active training approach that biases the model towards difficult samples, which, as demonstrated by experiments, significantly improves model performance. Specifically, at epoch t, starting with the model trained in the previous epoch as model t, training is divided into two stages. In the training stage, we train using the training set t filtered from the previous epoch to obtain model $t+1$. Then, using model $t+1$, we score the loss for each sample in the entire training set and rank these samples from high to low, see Fig. 6. The higher the loss for a sample, the greater the necessity for the model to re-learn that sample.

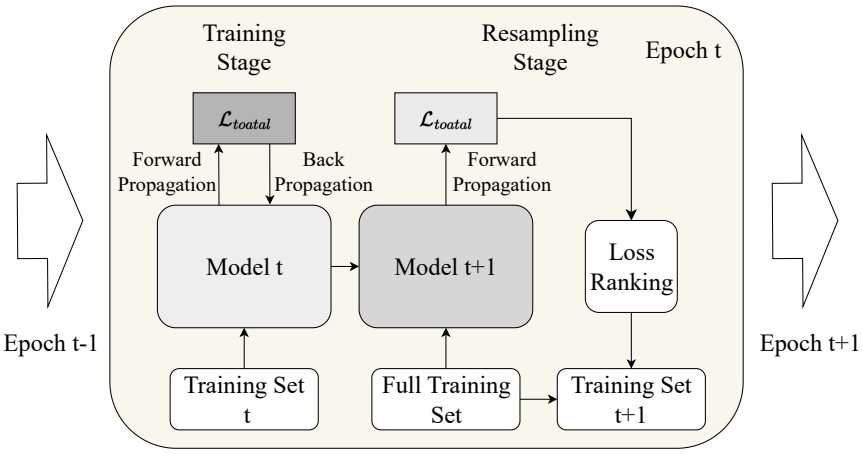

Fig. 6: Active training method. In the training stage, we train the model using the training set sampled from the previous stage. In the resampling stage, we use the model trained in the training stage to score the loss on the *full* training set, selecting the top K percent of samples to form the training set for the next training cycle.

Note, in the first round of training, we train the model using all samples. This active training method allows the model to learn from more challenging samples specifically. Although an additional resampling stage for loss ranking is required, this stage does not involve back-propagation, resulting in minimal computational overhead. Moreover, in each epoch (except the first), training only uses the top K percent of samples. Note that our training method is independent of the loss function, meaning it can be combined with any loss function from previous works to enhance model performance.

Here, our model loss is the sum of multiple loss functions, specifically, the cross-entropy loss $\mathcal{L}_{ce}$, lovasz-softmax $\mathcal{L}_{ls}$ [2], affinity loss $\mathcal{L}_{scal}^{geo}$ and $\mathcal{L}_{scal}^{sem}$ [4] (*i.e.*, geometric IoU and semantic mIoU) are combined as the model's loss function, formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{ce} + \mathcal{L}_{ls} + \mathcal{L}_{scal}^{geo} + \mathcal{L}_{scal}^{sem}. \tag{5}$$

4 Experiments

4.1 Experimental Setup

Dataset and Metrics. We conduct experiments on the challenging nuScenes dataset [3], the ground truth labels are from OpenOccupancy [48] and Occ3D [45]. The labels from Occ3D spans a range of $-40m$ to $40m$ for the X and Y directions and $-1m$ to $5.4m$ for the Z direction, with a voxel size of $0.4m$. For OpenOccupancy, the evaluation range for the X and Y axes is set to $[-51.2m, 51.2m]$, and for the Z axis, it is set to $[-3m, 5m]$. The voxel resolution is 0.2m, resulting in a final occupancy grid spatial scale of $40 \times 512 \times 512$. We utilize the metrics

Table 1: Performance on the nuScenes-Occupancy validation set [48]. *C*, *D*, *L*, *M* represent camera, depth, LiDAR, and multi-modal, respectively. Details of the baseline setup are available in the dataset [48].

Method	Input	IoU	mIoU	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [4]	C	18.4	6.9	7.1	3.9	9.3	7.2	5.6	3.0	5.9	4.4	4.9	4.2	14.9	6.3	7.9	7.4	10.0	7.6
TPVFormer [16]	C	15.3	7.8	9.3	4.1	11.3	10.1	5.2	4.3	5.9	5.3	6.8	6.5	13.6	9.0	8.3	8.0	9.2	8.2
3DSketch [6]	C&D	25.6	10.7	12.0	5.1	10.7	12.4	6.5	4.0	5.0	6.3	8.0	7.2	21.8	14.8	13.0	11.8	12.0	21.2
AICNet [20]	C&D	23.8	10.6	11.5	4.0	11.8	12.3	5.1	3.8	6.2	6.0	8.2	7.5	24.1	13.0	12.8	11.5	11.6	20.2
LMSCNet [42]	L	27.3	11.5	12.4	4.2	12.8	12.1	6.2	4.7	6.2	6.3	8.8	7.2	24.2	12.3	16.6	14.1	13.9	22.2
JS3C-Net [52]	L	30.2	12.5	14.2	3.4	13.6	12.0	7.2	4.3	7.3	6.8	9.2	9.1	27.9	15.3	14.9	16.2	14.0	**24.9**
C-CONet [48]	C	20.1	12.8	13.2	8.1	15.4	17.2	6.3	11.2	10.0	8.3	4.7	12.1	31.4	18.8	18.7	16.3	4.8	8.2
L-CONet [48]	L	30.9	15.8	17.5	5.2	13.3	18.1	7.8	5.4	9.6	5.6	13.2	13.6	34.9	21.5	22.4	21.7	19.2	23.5
M-CONet [48]	C&L	29.5	20.1	23.3	13.3	21.2	24.3	15.3	15.9	18.0	13.3	15.3	20.7	33.2	21.0	22.5	21.5	19.6	23.2
Co-Occ [36]	C&L	30.6	21.9	**26.5**	16.8	22.3	**27.0**	10.1	20.9	20.7	14.5	**16.4**	21.6	**36.9**	**23.5**	**25.5**	23.7	20.5	23.5
OccFusion (ours)	C&L	**32.4**	**22.4**	25.3	**17.0**	**22.5**	25.9	**16.5**	**22.4**	**24.0**	**16.1**	16.0	**22.1**	35.6	22.1	24.0	**23.9**	**21.3**	24.0

Table 2: 3D semantic occupancy prediction results on Occ3D [45] benchmark. *C*, *L*, *R* represent camera, LiDAR, and Radar, respectively.

Method	Input	mIoU	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. surf.	other flat	sidewalk	terrain	manmade	vegetation
MonoScene [4]	C	6.06	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65
BEVDet [15]	C	11.73	2.09	15.29	0.0	4.18	12.97	1.35	0.0	0.43	0.13	6.59	6.66	52.72	19.04	26.45	21.78	14.51	15.26
BEVFormer [24]	C	23.67	5.03	38.79	9.98	34.41	41.09	13.24	16.50	18.15	17.83	18.66	27.70	48.95	27.73	29.08	25.38	15.41	14.46
BEVStereo [23]	C	24.51	5.73	38.41	7.88	38.70	41.20	17.56	17.33	14.69	10.31	16.84	29.62	54.08	28.92	32.68	26.54	18.74	17.49
TPVFormer [16]	C	28.34	6.67	39.20	14.24	41.54	46.98	19.21	22.64	17.87	14.54	30.20	35.51	56.18	33.65	35.69	31.61	19.97	16.12
OccFormer [58]	C	21.93	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97
CTF-Occ [45]	C	28.53	8.09	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.00
RenderOcc [37]	C	26.11	4.84	31.72	10.72	27.67	26.45	13.87	18.20	17.67	17.84	21.19	23.25	63.20	36.42	46.21	44.26	19.58	20.72
BEVDet4D [14]	C	42.02	12.15	49.63	25.10	52.02	54.46	27.87	27.99	28.94	27.23	36.43	42.22	82.31	43.29	54.46	57.90	48.61	43.55
PanoOcc [50]	C	42.13	11.67	50.48	29.64	49.44	55.52	23.29	33.26	30.55	30.99	34.43	42.57	83.31	44.23	54.40	56.04	45.94	40.40
FB-OCC [25]	C	43.41	12.10	50.23	32.31	48.55	52.89	31.20	31.25	30.78	32.33	37.06	40.22	**83.34**	**49.27**	**57.13**	59.88	47.67	41.76
OctreeOcc [32]	C	44.02	11.96	51.70	29.93	53.52	56.77	30.83	33.17	30.65	29.99	37.76	43.87	83.17	44.52	55.45	58.86	49.52	46.33
Ming *et al.* [35]	C+L+R	46.67	**12.37**	50.33	31.53	**57.62**	**58.81**	33.97	41.00	47.18	29.67	**42.03**	48.04	78.39	35.68	47.26	52.74	**63.46**	**63.30**
OccFusion (ours)	C+L	**48.74**	12.35	**51.77**	**33.01**	54.56	57.65	**33.99**	**43.03**	**48.35**	**35.54**	41.22	**48.55**	83.00	44.65	**57.13**	**60.01**	62.46	61.25

of Intersection-over-Union (IoU) and mean Intersection-over-Union (mIoU) to evaluate our method's performance. Following [45, 48], we train our model on the training set and evaluated its performance on the validation set.

Implementation Details. Our method is based on CONet and is highly comparable to M-CONet [48] and Co-Occ [36]. For fairness, we adopt a foundational setup largely similar to [48]. Specifically, we utilize an ImageNet [7] pretrained ResNet101 [11] with FPN [27] as the 2D encoder for images, with an input image size of 1600×900. For LiDAR branch, we voxelize 10 LiDAR sweeps and use [53,60] as 3D encoder. During the training phase, we employ the AdamW [18] optimizer, with weight decay and initial learning rate set to 0.01 and 2e-4, respectively. A cosine learning rate scheduler with linear warm-up in the first 500 iterations is leveraged. Image augmentation strategies follow those used in BEVDet [15]. In the point pre-sampling process, hyper-parameters θ and τ are

set to 20 and 5, respectively. Our model is trained for 24 epochs on 8 A100 GPUs with a batch size of 8. Moreover, the effectiveness of the active training method proposed above (see Sec. 3.5) is tested separately (see Tab. 4).

4.2 3D Semantic Occupancy Prediction

We compare our method with recent state-of-the-art (SOTA) models, with accuracy data provided by [35,36]—all results reported by the authors themselves or obtained using open-source code (see Tabs. 1 and 2). Across two benchmarks, our method achieves SOTA mIoU and notably increases mIoU by a large margin (2.07) on Occ3D. Our approach also excels in small object categories. Specifically, on the nuscenes-occupancy benchmark, our model's IoU improves relative to M-CONet [48] by 27.8% for bicycle, 40.9% for motorcycle, 33.3% for pedestrian, and 21.1% for traffic cone. On Occ3D, our model also significantly enhances IoU in most small object categories. Notably, compared to M-CONet, our model reduces computational load by about 70% during the coarse to fine phase and maintains complexity comparable to vision-based methods (see Tabs. 5 and 6). On both benchmarks, compared to the best existing uni-modal methods, our model respectively achieves a relative mIoU increase of 41.8% and 10.7%, showcasing the significant advantages of multi-modal approaches.

4.3 Ablation Study

To validate the effectiveness of the methods we proposed and the computational complexity of our model, we conduct extensive ablation experiments on the OpenOccupancy benchmark [48].

The Role of Point Cloud Preprocessing. As noted in the literature [31], there is a density difference between LiDAR and camera features; without any preprocessing of the point cloud, "only 5% of camera features will be matched to a LiDAR point while all others will be dropped." By generating points in empty voxels, a denser point cloud can be achieved for intensive image feature sampling, alleviating this disparity. Farthest point sampling is only used in voxels with numerous LiDAR points, preserving the cloud's geometric features to some extent, reducing noise, and decreasing computational load during feature fusion. Omitting any of these methods would impair model accuracy, see Tab. 3.

Coordinate Concatenation. Although 3D-to-2D projection is generally more stable than depth estimation when provided with accurate intrinsic and extrinsic parameters, previous studies have shown that inaccuracies in extrinsic calibration can hinder the effectiveness of feature fusion [36]. We use deformable attention [61] to mitigate calibration errors. Notably, if LiDAR voxel features are used directly as querys, our method resembles Spatial Cross Attention (SCA) [24] in form. However, during feature fusion, we found that concatenating the coordinates of 3D reference points with LiDAR voxel features to form queries yields

Table 3: Ablation study on the proposed methods. *w/o* indicates without the corresponding method.

Methods	mIoU
w/o farthest point sampling	21.8
w/o points generation	21.2
w/o coordinate concatenation	22.0
w/o active training method	21.9
w/o active coarse to fine	**22.5**
OccFusion	22.4

Table 4: Ablation study on the resampling proportion in the resampling stage of the active training method.

Training resampling proportion (percent)	OccFusion mIoU	M-CONet mIoU
30	19.8	17.5
50	22.0	20.3
70	**22.4**	**21.2**

better results than using LiDAR voxel features alone, see Tab. 3. This might be because the coordinate information of each point can provide distinct offsets predictions for different reference points, thereby extracting the most effective camera features and point-by-point alleviating issues from imprecise calibration.

Generalizability of Active Training. Our proposed active training method is a very simple hard example mining technique. It's important to emphasize that this technique is designed to enhance the model's learning from challenging scenes, rather than biasing towards minority classes like loss functions such as focal loss [28] do. Our training method is independent of loss function design and applicable across all models. This means that by using our training method in conjunction with loss functions biased towards minority classes, we can enable the model to learn from difficult regions within challenging samples, thereby further improving model performance. Experiments show that the active training method, with a resampling ratio of 70%, simultaneously enhances the mIoU for both our OccFusion and M-CONet [48], demonstrating its high generalizability, see Tab. 4.

The Role of the Active Decoder. In Tabs. 3 and 5, we observe that removing the entropy gate mechanism in the decoder results in a 0.1 improvement in the model's mIoU. However, we point out that for practical applications of occupancy prediction, reducing model complexity can enhance inference speed, necessitating a balance between accuracy and complexity. The entropy gate reduces the computational load by approximately 70% during the feature refinement stage,

Table 5: Ablation study on the proportion of voxels refined in the active coarse to fine pipeline.

Coarse to fine proportion (percent)	mIoU
10	17.9
20	21.2
30	22.4
100	**22.5**

Table 6: Experiment on the computational efficiency. *GPU Mem.* represents the GPU memory consumption at training phase. ↓: the lower, the better. ↑: the higher, the better.

Methods	GPU Mem.(↓)	GFLOPs (↓)	mIoU (↑)
M-CONet	24.0 GB	3066	20.1
C-CONet	22.0 GB	2371	12.8
OccFusion	**17.0** GB	**1566**	**22.4**

achieving similar accuracy with significantly less complexity, making it a more efficient decoder. Moreover, like CONet [48], our decoder does not impose any restrictions on the model's head and can be extended to any occupancy prediction model to enhance their performance.

Computational Performance of Our Model. In Tab. 6, we showcase the computational performance of our OccFusion. Benefiting from our model's simpler mechanisms, it not only significantly enhances mIoU but also reduces the GFLOPs to half that of M-CONet. Notably, our model's GFLOPs are 34% lower than those of C-CONet (note that C-CONet is a uni-modal method), while nearly doubling the mIoU. During the training phase, our model also uses fewer computational resources. This highlights our model's efficiency in maintaining low computational demand while ensuring reliability as a multi-modal model, demonstrating its exceptional performance.

5 Conclusion

In this paper, we introduce OccFusion, a depth estimation-free multi-sensor fusion method that provides improved robustness over traditional methods. It incorporates a transferable active training method and an active occupancy decoder. Experiments on OpenOccupancy and Occ3D benchmarks confirm our method's superiority over current state-of-the-art models. Ablation studies further illustrates the effectiveness of our proposed components.

References

1. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: ICCV. pp. 9297–9307 (2019). https://doi.org/10.1109/ICCV.2019.00939
2. Berman, M., Triki, A.R., Blaschko, M.B.: The lovasz-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In: CVPR. pp. 4413–4421 (2018). https://doi.org/10.1109/CVPR.2018.00464
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11618–11628 (2020). https://doi.org/10.1109/CVPR42600.2020.01164
4. Cao, A., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: CVPR. pp. 3981–3991 (2022). https://doi.org/10.1109/CVPR52688.2022.00396
5. Chang, M.F., Lambert, J., Sangkloy, P., Singh, J., Bak, S., Hartnett, A., Wang, D., Carr, P., Lucey, S., Ramanan, D., Hays, J.: Argoverse: 3d tracking and forecasting with rich maps. In: CVPR. pp. 8748–8757 (2019). https://doi.org/10.1109/CVPR.2019.00895
6. Chen, X., Lin, K., Qian, C., Zeng, G., Li, H.: 3d sketch-aware semantic scene completion via semi-supervised structure prior. In: CVPR. pp. 4192–4201 (2020). https://doi.org/10.1109/CVPR42600.2020.00425
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009). https://doi.org/10.1109/CVPR.2009.5206848
8. Felzenszwalb, P.F., Girshick, R.B., McAllester, D.A., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1627–1645 (2010). https://doi.org/10.1109/TPAMI.2009.167
9. Firman, M., Aodha, O.M., Julier, S., Brostow, G.J.: Structured prediction of unobserved voxels from a single depth image. In: CVPR. pp. 5431–5440 (2016). https://doi.org/10.1109/CVPR.2016.586
10. Gal, Y., Islam, R., Ghahramani, Z.: Deep bayesian active learning with image data. In: ICML. pp. 1183–1192 (2017)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
12. Houlsby, N., Huszar, F., Ghahramani, Z., Lengyel, M.: Bayesian active learning for classification and preference learning. aeXiv preprint arXiv:1112.5745 (2011)
13. Hua, B.S., Pham, Q.H., Nguyen, D.T., Tran, M.K., Yu, L.F., Yeung, S.K.: Scenenn: A scene meshes dataset with annotations. In: 3DV. pp. 92–101 (2016). https://doi.org/10.1109/3DV.2016.18
14. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. CoRR **abs/2203.17054** (2022). https://doi.org/10.48550/ARXIV.2203.17054
15. Huang, J., Huang, G., Zhu, Z., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. aeXiv preprint arXiv:2112.11790 (2021)
16. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: CVPR. pp. 9223–9232 (2023). https://doi.org/10.1109/CVPR52729.2023.00890

17. Kim, J., Choi, J., Kim, Y., Koh, J., Chung, C.C., Choi, J.W.: Robust camera lidar sensor fusion via deep gated information fusion network. In: IV. pp. 1620–1625 (2018). https://doi.org/10.1109/IVS.2018.8500711
18. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
19. Kirsch, A., van Amersfoort, J., Gal, Y.: Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In: NeurIPS. pp. 7024–7035 (2019)
20. Li, J., Han, K., Wang, P., Liu, Y., Yuan, X.: Anisotropic convolutional networks for 3d semantic scene completion. In: CVPR. pp. 3348–3356 (2020). https://doi.org/10.1109/CVPR42600.2020.00341
21. Li, Y., Yu, Z., Choy, C.B., Xiao, C., Álvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: CVPR. pp. 9087–9098 (2023). https://doi.org/10.1109/CVPR52729.2023.00877
22. Li, Y., Yu, A.W., Meng, T., Caine, B., Ngiam, J., Peng, D., Shen, J., Lu, Y., Zhou, D., Le, Q.V., Yuille, A.L., Ta, M.: Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. In: CVPR. pp. 17161–17170 (2022). https://doi.org/10.1109/CVPR52688.2022.01667
23. Li, Y., Bao, H., Ge, Z., Yang, J., Sun, J., Li, Z.: Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In: Williams, B., Chen, Y., Neville, J. (eds.) AAAI. pp. 1486–1494. AAAI Press (2023). https://doi.org/10.1609/AAAI.V37I2.25234
24. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV. pp. 1–18 (2022). https://doi.org/10.1007/978-3-031-20077-9_1
25. Li, Z., Yu, Z., Austin, D., Fang, M., Lan, S., Kautz, J., Álvarez, J.M.: FB-OCC: 3d occupancy prediction based on forward-backward view transformation. CoRR **abs/2307.01492** (2023). https://doi.org/10.48550/ARXIV.2307.01492
26. Li, Z., Yu, Z., Austin, D., Fang, M., Lan, S., Kautz, J., Álvarez, J.M.: Fb-occ: 3d occupancy prediction based on forward-backward view transformation. aeXiv preprint arXIv:2307.01492 (2023). https://doi.org/10.48550/arXiv.2307.01492
27. Lin, T., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. pp. 936–944. IEEE Computer Society (2017). https://doi.org/10.1109/CVPR.2017.106
28. Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P.: Focal loss for dense object detection. In: ICCV. pp. 2999–3007. IEEE Computer Society (2017). https://doi.org/10.1109/ICCV.2017.324
29. Liu, P., Wang, L., Ranjan, R., He, G., Zhao, L.: A survey on active deep learning: From model driven to data driven. ACM Comput. Surv. **54**(10s), 221:1–221:34 (2022). https://doi.org/10.1145/3510414
30. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: ECCV. pp. 531–548 (2022). https://doi.org/10.1007/978-3-031-19812-0_31
31. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: IEEE International Conference on Robotics and Automation, ICRA 2023, London,

UK, May 29 - June 2, 2023. pp. 2774–2781 (2023). https://doi.org/10.1109/ICRA48891.2023.10160968
32. Lu, Y., Zhu, X., Wang, T., Ma, Y.: Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries. CoRR **abs/2312.03774** (2023). https://doi.org/10.48550/ARXIV.2312.03774
33. Miao, R., Liu, W., Chen, M., Gong, Z., Xu, W., Hu, C., Zhou, S.: Occdepth: A depth-aware method for 3d semantic scene completion. arXiv preprint arXiv:2302.13540 (2023). https://doi.org/10.48550/arXiv.2302.13540
34. Min, C., Xiao, L., Zhao, D., Nie, Y., Dai, B.: Uniscene: Multi-camera unified pre-training via 3d scene reconstruction. arXiv preprint arXiv:2305.18829 (2023). https://doi.org/10.48550/arXiv.2305.18829
35. Ming, Z., Berrio, J.S., Shan, M., Worrall, S.: Occfusion: A straightforward and effective multi-sensor fusion framework for 3d occupancy prediction. CoRR **abs/2403.01644** (2024). https://doi.org/10.48550/ARXIV.2403.01644
36. Pan, J., Wang, Z., Wang, L.: Co-occ: Coupling explicit feature fusion with volume rendering regularization for multi-modal 3d semantic occupancy prediction. IEEE Robotics Autom. Lett. **9**(6), 5687–5694 (2024). https://doi.org/10.1109/LRA.2024.3396092
37. Pan, M., Liu, J., Zhang, R., Huang, P., Li, X., Liu, L., Zhang, S.: Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. CoRR **abs/2309.09502** (2023). https://doi.org/10.48550/ARXIV.2309.09502
38. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV. pp. 194–210 (2020). https://doi.org/10.1007/978-3-030-58568-6_12
39. Qi, C.R., Zhou, Y., Najibi, M., Sun, P., Vo, K., Deng, B., Anguelov, D.: Offboard 3d object detection from point cloud sequences. In: CVPR. pp. 6134–6144 (2021). https://doi.org/10.1109/CVPR46437.2021.00607
40. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS. pp. 5099–5108 (2017)
41. Reading, C., Harakeh, A., Chae, J., Waslander, S.L.: Categorical depth distribution network for monocular 3d object detection. In: CVPR. pp. 8555–8564 (2021). https://doi.org/10.1109/CVPR46437.2021.00845
42. Roldão, L., de Charette, R., Verroust-Blondet, A.: Lmscnet: Lightweight multiscale 3d semantic completion. In: 3DV. pp. 111–119 (2020). https://doi.org/10.1109/3DV50981.2020.00021
43. Shrivastava, A., Gupta, A., Girshick, R.B.: Training region-based object detectors with online hard example mining. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016. pp. 761–769 (2016). https://doi.org/10.1109/CVPR.2016.89
44. Sung, K.K.: Learning and example selection for object and pattern detection. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, USA (1995), https://hdl.handle.net/1721.1/9836
45. Tian, X., Jiang, T., Yun, L., Mao, Y., Yang, H., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. In: NeurIPS. pp. 64318–64330 (2023)
46. Vobecky, A., Siméoni, O., Hurych, D., Gidaris, S., Bursuc, A., Pérez, P., Sivic, J.: Pop-3d: Open-vocabulary 3d occupancy prediction from images. arXiv preprint arXiv:2401.09413 (2024). https://doi.org/10.48550/arXiv.2401.09413
47. Vora, S., Lang, A.H., Helou, B., Beijbom, O.: Pointpainting: Sequential fusion for 3d object detection. In: CVPR. pp. 4603–4611 (2020). https://doi.org/10.1109/CVPR42600.2020.00466

48. 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: ICCV. pp. 17804–17813 (2023). https://doi.org/10.1109/ICCV51070.2023.01636
49. Wang, Y., Chao, W.L., Garg, D., Hariharan, B., Campbell, M., Weinberger, K.Q.: Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In: CVPR. pp. 8445–8453 (2019). https://doi.org/10.1109/CVPR.2019.00864
50. Wang, Y., Chen, Y., Liao, X., Fan, L., Zhang, Z.: Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. CoRR **abs/2306.10013** (2023). https://doi.org/10.48550/ARXIV.2306.10013
51. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: ICCV. pp. 21672–21683 (2023). https://doi.org/10.1109/ICCV51070.2023.01986
52. Yan, X., Gao, J., Li, J., Zhang, R., Li, Z., Huang, R., Cui, S.: Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In: AAAI. pp. 3101–3109 (2021). https://doi.org/10.1609/AAAI.V35I4.16419
53. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10) (2018). https://doi.org/10.3390/S18103337
54. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., Zhou, J., Dai, J.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: CVPR. pp. 17830–17839 (2023). https://doi.org/10.1109/CVPR52729.2023.01710
55. Yin, T., Zhou, X., Krähenbühl, P.: Multimodal virtual point 3d detection. In: NeurIPS. pp. 16494–16507 (2021)
56. Zhang, C., Yan, J., Wei, Y., Li, J., Liu, L., Tang, Y., Duan, Y., Lu, J.: Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. arXiv preprint arXiv:2312.09243 (2023). https://doi.org/10.48550/arXiv.2312.09243
57. Zhang, Y., Zheng, W., Zhu, Z., Huang, G., Lu, J., Zhou, J.: A simple baseline for multi-camera 3d object detection. In: AAAI. pp. 3507–3515 (2023). https://doi.org/10.1609/aaai.v37i3.25460
58. Zhang, Y., Zhu, Z., Du, D.: Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In: ICCV. pp. 9399–9409 (2023). https://doi.org/10.1109/ICCV51070.2023.00865
59. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: CVPR. pp. 13750–13759 (2022). https://doi.org/10.1109/CVPR52688.2022.01339
60. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018). https://doi.org/10.1109/CVPR.2018.00472
61. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2021)