This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Camera-Only 3D Panoptic Scene Completion for Autonomous Driving through Differentiable Object Shapes

Nicola Marinello[1] Simen Cassiman[1] Jonas Heylen[2] Marc Proesmans[1,2]
Luc Van Gool[1,2,3,4]
[1] KU Leuven [2] TRACE vzw [3] ETH Zürich [4] INSAIT
{nicola.marinello,simen.cassiman,marc.proesmans,luc.vangool}@esat.kuleuven.be
jonas.heylen@trace.vision

Abstract

Autonomous vehicles need a complete map of their surroundings to plan and act. This has sparked research into the tasks of 3D occupancy prediction, 3D scene completion, and 3D panoptic scene completion, which predicts a dense map of the ego vehicle surroundings as a voxel grid. Scene completion extends occupancy prediction by predicting occluded regions of the voxel grid, and panoptic scene completion further extends this task by also distinguishing object instances within the same class; both aspects are crucial for path-planning and decision-making. However, 3D panoptic scene completion is currently underexplored. This work introduces a novel framework for 3D panoptic scene completion that extends existing 3D semantic scene completion models. We propose an Object Module *and* Panoptic Module *that can easily be integrated with 3D occupancy and scene completion methods presented in the literature. Our approach leverages the available annotations in occupancy benchmarks, allowing individual object shapes to be learned as a differentiable problem. The code is available at* https://github.com/nicolamarinello/OffsetOcc.

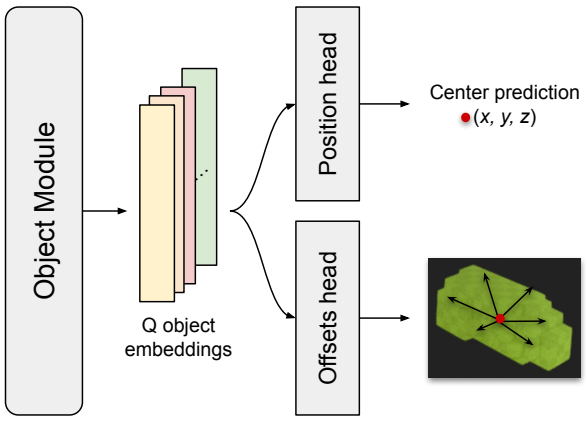

Figure 1. **OffsetOcc.** Our method OffsetOcc models object occupancy as a set prediction task, where objects are represented by their 3D center and a set of offsets that describe their shape. The set of offsets is learned from the dataset annotations. Classification head omitted for conciseness.

1. Introduction

Computer Vision (CV) [33] is, in large part, concerned with extracting conceptual information from sensory data. The field has seen significant advancements, largely driven by improvements in Deep Learning [16], allowing it to tackle increasingly difficult tasks. For the better part of the last decade, progress was mainly viewed in the light of benchmarks, providing handcrafted, narrow, and sometimes artificial metrics. However, recent advances have allowed methods to go beyond these isolated experiments to their incorporation into various applications showcasing their potential in the real world.

One of the most useful applications may well be enabling autonomous agents, analyzing sensory data to build up an accurate representation of their surroundings to guide their decision-making. This has motivated research into 3D object detection [27], which deals with localizing objects in 3D space, and the generation of birds-eye-view (BEV) maps [20], which extracts a 2D top view of the entire scene. However, while performance has improved greatly, these methods leave out important information for subsequent operational modules such as prediction and planning. Detection models output bounding boxes, which only provide coarse object shape estimates, ignoring uncountable classes (referred to as *stuff*, *e.g.* road) [34]. In contrast, BEV maps handle stuff and can provide more accurate shapes on the ground plane, but they lose important height information, which is hard to recover and could impact downstream

tasks.

These shortcomings lead to the introduction of Semantic Occupancy Prediction [42, 47], the task of labeling each visible point in space. In the case of autonomous driving, initial methods labeled LIDAR point clouds, but these are usually very sparse and leave out a lot of general knowledge about scene structure that is, again, very important to solve the actual end task of autonomy. Therefore, more recent work handles voxel occupancy prediction [34, 35, 39], tasked with completing a full 3D grid of the scene (*i.e*. semantic labeling of 3D or volumetric pixels, know as *voxels*). The latest research moves to camera-only settings [2], improving the learned features [21, 46, 49] and its efficiency [8, 10, 12, 17, 26, 32, 37, 44, 48].

It is important to note that occupancy prediction aims to reconstruct only the directly detectable voxels (*i.e*. detected in the current frame by the camera and/or LIDAR), which should be distinguished from Semantic Scene Completion [26, 34, 42] that intends to also reconstruct regions that are currently occluded by other portions of the scene. This is a very important distinction with big implications for real-world applications. Most models evaluate the reconstruction performance by using a "camera mask" which indicates which voxels are directly detectable in the current frame, since these regions are determined with relative certainty and can be considered ground truth, in contrast to undetected regions. However, while knowing how well a method handles directly visible areas is certainly important, it is an incomplete task, since this artificial mask that knows what has been detected and what not is not available in a real-world situation. Furthermore, models trained with this mask exploit hidden regions to improve their accuracy [24]. This can give a false sense of performance by discarding predictions in occluded regions (which might be very bad) that are not distinguishable during inference. Moreover, it is clear that humans can infer information about some of the occluded areas, especially about partially occluded objects, indicating that CV methods must share this ability to level the playing field for reasoning capabilities. Therefore, it is important to develop and evaluate models for these hidden regions as well; visible portions must be reconstructed accurately and occluded areas should contain reasonable predictions.

Furthermore, the planning and decision-making of an autonomous agent also depend on the actions of other agents in the scene. Therefore, it is important to distinguish each instance within the same class. This has lead to models that can produce panoptic occupancy and scene completion outputs [40, 45]. However, this topic is underexplored and many possibilities to distinguish instances remain to be tested or developed.

This work presents a 3D panoptic scene completion framework for autonomous driving using only camera inputs. It achieves this by using a separate object decoder through which shapes can be learned differentially and per class. Conventional methods that uplift 2D features to 3D and leverage self-attention mechanisms effectively infer overall object occupancy. However, approaches that distinguish individual object instances within the scene remain underexplored. Notably, the proposed method can be seamlessly integrated with more advanced occupancy decoders and enhanced with temporal reasoning, as explored in recent works.

Our contributions are summarized as follows:

- We present a novel framework that models individual object occupancy as a differentiable problem.
- We present an *Object Module* and *Panoptic Module* that can extend existing 3D semantic scene completion methods, enabling panoptic 3D scene completion.
- We release our implementation to promote further research.

2. Related work

Many works in this field primarily focus on semantic occupancy prediction, trying to improve the learned features, increasing model efficiency, or tackling the complex label annotation problem. Fewer works investigate the scene completion objective, which is closely related and can often benefit from improvements to the former. Lastly, very few, and only recently, have proposed panoptic scene completion methods to distinguish the different object instances within the same class.

Semantic Occupancy Prediction. To learn better features Li *et al*. [21] propose using additional losses such as depth and 2D segmentation, also explored by Zheng *et al*. [49] in the form of a semantic loss and additional temporal information from past frames, a practice that is becoming increasingly popular. One branch of works tackling efficiency handle it by way of a lighter representation, such as keeping the BEV as proposed by Yu *et al*. [44], a hybrid model with BEV and BEV slices put forth by Zhang *et al*. [46] or BEV and voxels by He *et al*. [8], extending the 2D representation to triplane view (TPV) as presented by Huang *et al*. [10], or using a low rank approximation with vertical vectors and horizontal matrices per Zhao *et al*. [48]. Tian *et al*. [34] propose using a full voxel grid, but starting with a coarser version that gets refined until the output dimension. Others reduce computational cost by choosing different operations, like Sze *et al*. [32] that use sparse convolutions, Wang *et al*. [37] that use a linear complexity attention with Receptance Weighted Key Value operations [30], Li *et al*. [17] that exchange transformers for Mamba [6], or Huang *et al*. [12] that use gaussians to model regions of interest, only allocating compute where it is required and with the additional benefit that the decoding is object-centric. Wang *et al*. [38] explore reliability by incorporating uncertainty learning, an

aspect that is also handled by Wang *et al*. [36] using diffusion models, which naturally incorporate a coarse-to-fine iterative refinement. Lastly, Pan *et al*. [28] try to avoid dense labeles by using NERFs and 2D labeling, which Pan *et al*. [29] follow up by adding a teacher-student pseudo-labeling setup. Huang *et al*. [11] even go a step further, using self-supervised learning by predicting the frames before and after.

While there has been great progress thanks to these works, it is only shown for visible areas, ignoring the multiple occluded regions which contain relevant information for modules further in the pipeline of autonomous vehicles. Moreover, by training these methods with the mask, they learn unrealistic object shapes, which can actually be more different from each other and, therefore, potentially make learning general concepts more difficult.

Semantic Scene Completion. In contrast to the previous paragraph, scene completion methods aim to reasonably reconstruct the entire scene, including occluded areas. Cao *et al*. [2] use a monocular system to show that RGB camera-only setups can also provide reasonable outputs, even without additional depth information. Li *et al*. [19] achieve this by explicitly handling occupancy and completion separately, first reconstructing the visible voxels and then using a masked autoencoder to complete the occluded areas. Wei *et al*. [41] employ Poisson Reconstruction [13] to fill in holes and use it to generate denser ground-truth. Due to the high sparsity, Ma *et al*. [26] propose to work in a strongly down-sampled representation, where all operations can be computed more efficiently after which it is upsampled and classes are predicted per mask. Lastly, sparsity is also exploited by Liu *et al*. [24], by only using sparse operations until the very end. To increase performance, they also add temporal information. Additionally, they propose RayIoU, to deal with the inaptitude of mean intersection over union (mIOU), the dominant metric in the field, when handling depth errors.

Despite extracting more information by filling in invisible parts, these methods rely on the models to learn any relationships between classes and their plausible shapes that are required to complete occluded areas. If a car is only partially visible, the model must learn that voxels in the occluded region should be related to it, which is a very complex learning problem, especially when labels are imperfect in these regions.

Panoptic Scene Completion. Wang *et al*. [40] combine 3D detection and segmentation into one framework, using sparsity techniques for efficiency and temporal reasoning for improved completion. By incorporating detection, they can separate the instances within each class. Yu *et al*. [45] build on Flash-Occ, for its efficiency, to first predict occupancy and then combine the related voxels into instances.

However, instance voxels are decoded individually, meaning that models must learn what it means to be part of an object, and that objects only have a single class. Within the voxel decoding modules there is no notion of an entity that it belongs to, they are decoded bottom-up and then the instance is estimated. This discards valuable top-down, object-centric information and, therefore, seems suboptimal.

3. Methodology

3.1. Problem setup

3D semantic scene completion is the task of reconstructing the complete surrounding environment of an ego vehicle, including occluded regions (within reason, *i.e*. it cannot be reasonably expected that objects which have never been seen, behind a building corner, are also predicted). This work focuses on the case where only camera inputs are available (*i.e*. no LIDAR inputs are used). The environment is represented as a voxel grid, with each voxel receiving a semantic label. Given the observations from N cameras at a specific timestamp $\mathbf{I}_t = \{I_t^1, I_t^2, \ldots, I_t^N\}$, the goal is to generate a grid $\hat{\mathbf{Y}}_t \in \{c_0, c_1, \ldots, c_M\}^{L \times W \times H}$ where c_0 denotes the empty class to indicate unoccupied space, $\{c_1, ...c_M\}$ denote the semantic classes, while L, W and H denote the length, width and height of the grid, respectively. The task can be extended to panoptic scene completion, where all voxels occupied by the same object instance are assigned the same object ID which is unique across all objects.

3.2. Object occupancy as a set prediction

We propose a novel framework *OffsetOcc* for the 3D panoptic scene completion task that models individual objects and their shapes as a differentiable set prediction problem. Inspired by DETR [3], we model each object's occupied voxels as a set of 3D vectors relative to the center of the object, as shown in Fig. 1. Each of these vectors is referred to as an *offset*, which aims to point towards the center of one of the voxels occupied by the object. This modeling allows us to effectively represent the occupancy of the object within the 3D grid. Such a representation is more efficient than computing a full grid mask of each object, which becomes computationally prohibitive. This approach facilitates a structured and precise representation of object shapes, supporting downstream tasks that require accurate object estimation.

3.3. Architecture

In this section, we detail the architecture of our model. We present an *Object Module* and *Panoptic Module* which can be used on top of a generic baseline model. The different components of this baseline can be instantiated with various methods proposed in the literature.

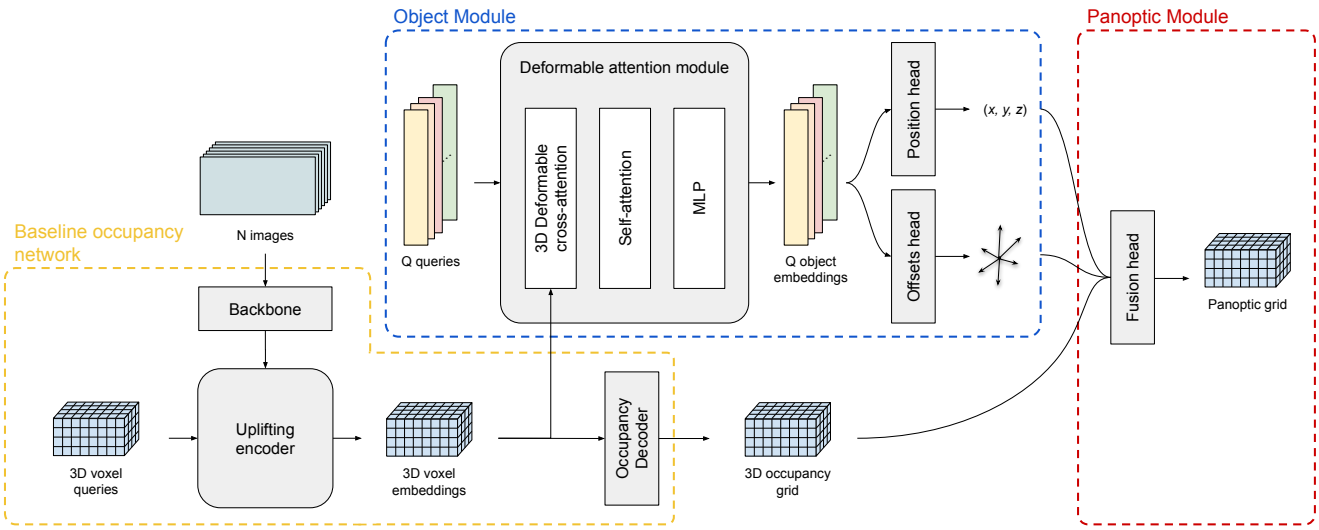

Figure 2. **Overview of our framework OffsetOcc.** Our approach can be integrated with modern occupancy models and consists of two key components: an *Object Module* and a *Panoptic Module*. The *Object Module* utilizes features extracted by an uplifting encoder to detect objects in the voxel grid and predict their occupancy using the offset occupancy mechanism. The *Panoptic Module* then merges these object predictions with the occupancy grid output to produce a panoptic occupancy map. Classification head in the *Object Module* omitted for conciseness.

As shown in Fig. 2, the model takes multi-view images as input, from which features are extracted via a backbone. Subsequently, an *Encoder* uses these camera features to generate 3D representations in a (down-sampled) voxel grid, which we will refer to as voxel embeddings. These then pass through the *Decoder* to generate the baseline voxel occupancy predictions. Our proposed *Object Module* also takes these voxel embeddings as input, and provides instance-level occupancy predictions. Lastly, the baseline voxel occupancy prediction and the instance-level predictions are passed to our proposed *Panoptic Module*, which generates the final panoptic voxel occupancy. The structure and responsibility of each module is further detailed in the paragraphs below. Since the proposed modules are meant to work with generic baselines, we focus on the structure of our proposed method; specific implementations of the baseline modules are detailed in Sec. 4.2.

Backbone. A multi-view image feature extractor of choice. A common set-up in the literature is the combination of a CNN backbone with an FPN on top for multi-level feature extraction.

Encoder. The encoder lifts 2D image features to a 3D representation, providing a per-voxel feature vector. Optionally, the voxel grid at this stage can be predicted as a down-sampled version of the final prediction to reduce the computational burden.

Decoder. The decoder takes in the voxel embeddings and computes the final label predictions for each location in the grid. This prediction is at full resolution, thus, if the intermediate voxel grid was down-sampled, the decoder must also take care of up-sampling.

Object Module. Inspired by DETR's detection approach [3], we extend its capabilities to predict object shapes in the form of voxel occupancy, in addition to their class and location. Our method utilizes a stack of M transformer layers composed of 3D deformable cross-attention [35, 51] and self-attention layers. Starting with Q learnable object queries, the model produces Q object embeddings using the cross-attention layers to extract contextual information from the voxel feature grid, while self-attention layers allow the exchange of information between object embeddings. After this stack, the embeddings are further enhanced by a feed-forward network (FFN), and passed to a class head, a position head, and an offsets head which predict the object's class, the 3D object center location, and shape occupancy, respectively. Specifically, for each object i we predict the label $\hat{y}^i \in \{c_0, c_1, \ldots, c_M\}$, the 3D location $\hat{\mathbf{c}}^i = (x, y, z)$, and a set of K offsets relative to the center of the object. The set of offsets for an object is defined as $\hat{\mathbf{o}}^{\mathbf{i}} = \{\hat{\mathbf{o}}_1^i, ..., \hat{\mathbf{o}}_K^i\}$, where each $\hat{\mathbf{o}}_k^i = (o_{k,x}^i, o_{k,y}^i, o_{k,z}^i)$ is a 3D vector. For each offset, the linear layer also predicts an occupancy score $\hat{s}_k^i$. The occupancy score spans the range $[0, 1]$. Formally we describe an object occupancy as follows. First, given the object center and the set of offsets, we define the object point cloud as the set

$$\hat{V}^i = \{\hat{\mathbf{v}}_k^i = \hat{\mathbf{c}}^i + \hat{\mathbf{o}}_k^i, \forall k \in \{1, 2, \ldots, K\} \mid \hat{s}_k^i \geqslant 0.5\} \quad (1)$$

Subsequently, an occupancy mask is obtained by voxeliz-

ing the resulting point cloud. The number of offsets K is a parameter and must be sufficiently large to describe objects of any size. The model is trained to assign a score ≥ 0.5 to a subset of offsets $K^{'} \leqslant K$. The direction of offsets with score ≥ 0.5 aim to align with the center of the grid voxels occupied by the object. Offsets with a score below 0.5 are discarded as they are predicted to be unnecessary to describe the object occupancy. Within this framework, the object center localizes each object within the grid, while the relative offsets describe its occupancy shape. This framework allows us to model the object's occupancy within the grid as a continuous and differentiable problem. Objects classified as background are discarded. By extending the DETR approach, our model inherently achieves panoptic capabilities, as each object's occupancy is decoded independently from a separate query.

Panoptic Module. This module is responsible for merging the occupancy predictions from the baseline and the *Object Module* to generate panoptic occupancy predictions. We adopt a simple, parameter-free approach, where each voxel classified as an object by the baseline decoder is assigned an instance ID based on the *Object Module* prediction. To smooth out the prediction and correct small localization errors, we use a voting scheme using the voxels within radius r of the reference location. The most voted ID (*i.e.* the ID that is assigned most often to the sampled set of nearby voxels) is assigned to the reference voxel, and if no meaningful ID was sampled, then the empty class is assigned.

3.4. Training methodology

In this section, we present our model's training methodology, which uses two stages. In the first stage, the baseline model is trained using the semantic labels of the occupancy grid according to $\mathcal{L}_{\text{SSC}}$. In the second stage, the *Object Module* and *Panoptic Module* are added and trained using $\mathcal{L}_{\text{objects}}$, while the baseline model is frozen.

Baseline model supervision. The baseline model is supervised based on the predictions of the decoder with the ground truth grid $\mathbf{Y}_t \in \{c_0, c_1, \ldots, c_M\}^{L \times W \times H}$ through the Semantic Scene Completion (SSC) loss $\mathcal{L}_{\text{SSC}}$ [2].

Object module supervision. The computation of the objects loss $\mathcal{L}_{\text{objects}}$ stems from two components: an object detection loss $\mathcal{L}_{\text{det}}$ and an occupancy loss $\mathcal{L}_{\text{occ}}$.

The object detection loss is similar to the approach in DETR [3] that combines a classification loss $\mathcal{L}_{cls}$ and a center distance loss $\mathcal{L}_{dist}$. We use a focal loss [23] for the classification and L_2 loss for the distance loss, resulting in

$$\mathcal{L}_{\text{det}} = \lambda_1 \mathcal{L}_{\text{cls}} + \lambda_2 \mathcal{L}_{\text{dist}}, \tag{2}$$

We match object predictions to ground truth with the Hungarian algorithm [15] using a cost function analogous to the loss in Eq. (2) (any unmatched predictions are associated with the background class), which defines a mapping σ_{det} that assigns object prediction indices i to ground truth indices j such that $j = \sigma_{\text{det}}(i)$ (where the ground truth has been padded with no-object to allow queries to predict no object).

For the occupancy loss, once the association at object level is established, we apply again the same concept at a voxel level, within each individual object. Let us consider a prediction – ground truth pair $((\hat{y}^i, \hat{\mathbf{c}}^i, \hat{V}^i); (y^{\sigma_{\text{det}}(i)}, \mathbf{c}^{\sigma_{\text{det}}(i)}, V^{\sigma_{\text{det}}(i)}))$. We want to find the best match between the point cloud points $\hat{V}^i$ and the list of ground truth voxel centers $V^{\sigma_{\text{det}}(i)}$ occupied by the object. This can be achieved by reapplying the Hungarian algorithm (bipartite matching) at the voxel level. Assuming K is always larger than the amount of voxels occupied by a single object, we convert the ground truth set to a list of K elements by padding with $\varnothing$. The matching cost function for a single pair offset – ground truth voxel is given by

$$\begin{aligned}\mathcal{L}_{\text{match}}(k, \sigma(k)) = &- \mathbb{1}_{\{s_{\sigma(k)} \neq \varnothing\}} \hat{s}_k \\ &+ \mathbb{1}_{\{s_{\sigma(k)} \neq \varnothing\}} \mathcal{L}_{\text{dist}} \left(\hat{\mathbf{v}}_k, \mathbf{v}_{\sigma(k)}\right)\end{aligned} \tag{3}$$

where σ is one of the possible matching permutations. Here we also use the L_2 norm as a distance cost between the predicted offset and the matched voxel center.

Once the optimal match σ_{occ} is determined we compute the loss as

$$\mathcal{L}_{\text{occ}} = \sum_{k=1}^{K} \left[-\log \hat{s}_k + \mathbb{1}_{\{s_{\sigma_{\text{occ}}(k)} \neq \varnothing\}} \mathcal{L}_{\text{dist}}(\hat{\mathbf{v}}_k, \mathbf{v}_{\sigma_{\text{occ}}(k)}) \right]. \tag{4}$$

In practice we also use a focal loss for the occupancy $\hat{s}$ and balance the classification and regression components with coefficients, which leads to

$$\mathcal{L}_{\text{occ}} = \lambda_3 \mathcal{L}_{\text{focal}} + \lambda_4 \mathcal{L}_{\text{dist}}. \tag{5}$$

Finally, the object loss is given by

$$\mathcal{L}_{\text{objects}} = \mathcal{L}_{\text{det}} + \mathcal{L}_{\text{occ}}. \tag{6}$$

Location and point cloud supervision decoupling. During training, unlike during inference, we add the predicted offsets to the *ground truth location* $\mathbf{c}^i$ of the matched object, instead of the predicted location $\hat{\mathbf{c}}^i$. This approach decouples the learning of object centers and offsets, ensuring that the offset loss does not depend on the accuracy of the predicted objects' locations. As a result, the model focuses on learning the actual object occupancy without compensating for inaccurate location predictions, allowing each component to learn independently.

Lastly, we only train on objects that have at least one visible voxel from the cameras, even though they may occupy several non-visible voxels.

4. Experiments

4.1. Datasets

nuScenes dataset. nuScenes [1] is a large-scale, multi-modal dataset containing 1,000 driving scenes, divided into 700 scenes for training, 150 for validation, and 150 for testing. Each scene is approximately 20 seconds long and recorded at a frequency of 20 Hz. The dataset provides RGB images captured from six cameras with a 360° horizontal field of view, along with LIDAR point cloud data. For the object detection task, nuScenes includes annotations of 3D bounding boxes on keyframes at 2 Hz, labeled with 10 *thing* classes.

Occ3D-nuScenes. Occ3D-nuScenes [34] extends the nuScenes dataset by incorporating voxel-wise semantic occupancy annotations, making it suitable for 3D occupancy prediction. The occupancy grid covers a range of $-40\,\mathrm{m}$ to $40\,\mathrm{m}$ along the x and y axes, and from $-1\,\mathrm{m}$ to $5.4\,\mathrm{m}$ along the z-axis in the ego coordinate frame. The voxel grid is defined with a resolution of $0.4\,\mathrm{m} \times 0.4\,\mathrm{m} \times 0.4\,\mathrm{m}$, allowing for dense spatial representation. Occ3D-nuScenes provides semantic labels for 17 categories, which include 16 known object classes and an additional "empty" class. Additionally, it includes a visibility mask to indicate occluded voxels in the camera view. Since the Occ3D-nuScenes dataset does not include panoptic annotations, we generate them offline by intersecting semantic voxel annotations with 3D bounding boxes.

Evaluation metrics. We use intersection over union (IOU) to evaluate scene occupancy by distinguishing between occupied and non-occupied voxels, independent of semantic labels. This binary occupancy map is essential for capturing spatial structure in 3D environments. To assess semantic segmentation performance, we compute MIOU over semantic classes, accounting for both occupancy and class-specific accuracy. The standard evaluation protocol for 3D semantic occupancy relies on the visibility mask, computing errors only on visible voxels. However, this approach overlooks the accuracy of object shape reconstruction, which is crucial for safe path planning and aligns more naturally with a panoptic framework. To provide a more comprehensive evaluation and facilitate future comparisons, we also report performance without applying the visibility mask. We also report the Panoptic Quality (PQ) metric for the panoptic LIDAR segmentation task. PQ is defined as the product of Recognition Quality (RQ), which measures detection performance, and Segmentation Quality (SQ), which evaluates the segmentation consistency of correctly matched instances [14].

4.2. Experimental settings.

The input images are 1600×1200 and augmented with GridMask [4] and photometric distortion.

For the baseline model we take inspiration from [20, 35]. We use a ResNet101 [7] pre-trained on Imagenet [5] and an FPN [22] as the backbone. We implement the encoder with a transformer that uses a grid of 1D sinusoidal positional voxel queries $\mathbf{Q} \in [-1, 1]^{L/2\times W/2\times H/2\times D}$, and a stack of four layers with deformable cross-attention [51] and 3D deformable self-attention [35, 51]. The cross-attention layers project reference points (the voxel grid centers) into camera views via projection matrices, aggregating image features potentially from multiple views (in which case features for that point are averaged). The hidden network dimension D is 128 and the voxel embeddings are predicted in a grid at half resolution. The decoder up-samples the voxel embeddings to the output resolution of $200 \times 200 \times 16$ using trilinear interpolation and predicts the class per voxel with a linear layer.

In the *Object Module* we set the number of object queries $Q = 900$, the number of layers $M = 4$, and the number of offsets $K = 2197$. In the *Panoptic Module* we set $r = 9$ and use the Manhattan distance.

The loss weights are set as follows: $\lambda_1 = 0.5$, $\lambda_2 = 0.02$, $\lambda_3 = 0.125$, $\lambda_4 = 0.0125$. Each stage runs for 50 epochs with AdamW [25], a learning rate of 2×10^{-4} and an exponential learning rate decay scheduler.

4.3. Main results

In this section we evaluate the performance of our model for the task of 3D semantic occupancy and panoptic LIDAR segmentation as a proxy for panoptic occupancy performance. In Fig. 3 we include qualitative results of our model.

3D semantic occupancy. We evaluate our model on the nuScenes validation set, with results summarized in Tab. 1. The model achieves 28.0 MIOU and 43.9 IOU, delivering performance comparable to existing works. Since our goal is to develop a model capable of predicting complete object occupancies, we also report, in Tab. 1, the model's performance evaluated without the visibility mask, providing a more comprehensive assessment of its reconstruction capabilities. Under these settings, it achieves 17.2 MIOU and 24.9 IOU. For completeness, we also report the performance of previous work trained with the visibility mask.

Panoptic LIDAR segmentation. Results on panoptic LIDAR segmentation are reported in Tab. 2. For evaluation, LIDAR points were labeled according to their corresponding voxel labels in the occupancy grid. The model achieved a PQ score of 29.4, demonstrating the validity of our approach. It is likely that the gap with the best-performing models can be attributed to two main factors. First, our framework does not retrain the base model which might limit the relevant features that it can extract; second, our method does not incorporate temporal consistency across frames, which could be especially useful for improving predictions of distant and (partially) occluded objects.

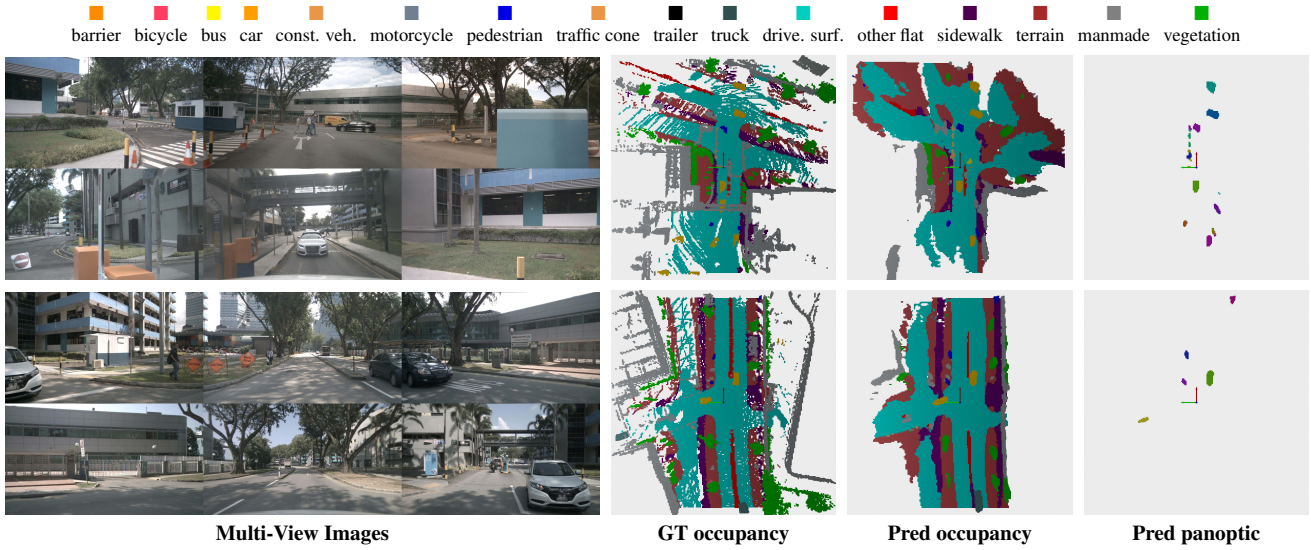

Figure 3. **Qualitative results on Occ3D-nuScenes validation set.** Our model accurately differentiates and segments different objects in the scene.

Method	Image Backbone	Temporal	Train w/ mask	Evaluate w/ mask	mIoU	IoU
MonoScene [2]	R101-DCN	✗	✗	✓	6.1	-
BEVDet [9]	R101-DCN	✗	✗	✓	19.4	-
OccFormer [46]	R101	✗	✗	✓	21.9	-
BEVFormer [20]	R101-DCN	✓	✗	✓	26.9	-
TPVFormer [10]	R101-DCN	✓	✗	✓	27.8	-
CTF-Occ [34]	R101-DCN	✗	✗	✓	28.5	-
SparseOcc [24]	R50	✓	✗	✓	30.9	-
PanoOcc [40]	R101-DCN	✓	✗	✓	32.5	-
TPVFormer‡ [10]	R50	✓	✓	✓	34.2	66.8
OccFormer‡ [46]	R50	✗	✓	✓	37.4	70.1
BEVFormer [20]	R101-DCN	✓	✓	✓	39.2	-
PanoOcc [40]	R101-DCN	✓	✓	✓	**44.5**	**75.0**
OffsetOcc (Ours)	R101	✗	✗	✓	28.0	43.9
OffsetOcc (Ours)	R101	✗	✗	✗	17.2	24.9

Table 1. **3D Occupancy prediction performance on the Occ3D-nuScenes dataset.** "Temporal" indicates that the model uses past frames when generating predictions. "Train w/ mask" and "Evaluate w/ mask" indicate whether the model has been trained using the camera mask and whether the performance has been measured using the camera mask, respectively. ‡ indicates performance measured by [26]. Best performance is **bolded** and second best is underlined. Per-class performance can be found in the supplementary tables in Appendix A.

Method	Input Modality	PQ	$PQ^{\dagger}$	RQ	SQ
EfficientLPS [31]	LIDAR	62.0	65.6	73.9	83.4
Panoptic-PolarNet [50]	LIDAR	63.4	67.2	75.3	83.9
Panoptic-PHNet [18]	LIDAR	74.7	**77.7**	84.2	88.2
LidarMulitiNet [43]	LIDAR	**81.8**	-	**90.8**	**89.7**
PanoOcc [40]	Camera	62.1	66.2	75.1	82.1
OffsetOcc (Ours)	Camera	29.4	38.3	40.1	66.5

Table 2. **LIDAR panoptic segmentation results on nuScenes validation set.** Our OffsetOcc exhibits reasonable performance, albeit clearly behind the state of the art (SOTA). Many recent improvements included in the SOTA can readily be included in our method which will likely reduce the gap significantly.

Variation	PQ^{Th}	RQ^{Th}	SQ^{Th}
OffsetOcc w/o decoupling	15.8	22.0	**69.9**
OffsetOcc	**17.0**	**23.7**	69.7

Table 3. **Offsets loss and location error decoupling.** Decoupling object location error from point cloud supervision improves performance on object classes, validating the effectiveness of our approach. The models were trained on 50% of the dataset and evaluated on the full validation set of nuScenes.

4.4. Discussion and ablation

In this section, we analyze the impact of our offsets loss and location error decoupling, the computational cost introduced by the *Object Module* for panoptic occupancy, and evaluate the effect of the majority voting radius r in the *Panoptic Module*.

Offsets loss and location error decoupling. Since the offsets L_2 loss focuses on reconstructing the object point cloud as accurately as possible, it is more effective, during supervision, to compare the predicted point cloud with the ground truth point cloud, without factoring in the localization error

of the object center. As discussed in Sec. 3.4, during supervision we sum the predicted offsets to the ground truth object center. The loss between the resulting point cloud and the ground truth point cloud will therefore only penalize inaccurate offsets. We compare in Tab. 3 the proposed model with a model that does not employ this design choice: *i.e.* during supervision the point cloud is determined as the sum of the predicted object location and predicted offsets. We compare two models that use voting radius $r = 0$ to evaluate the raw offsets quality. The results indicate that decoupling produces better object segmentation, achieving 17.0 PQ^{Th} compared to 15.8 PQ^{Th} for the model without the decoupling strategy. Notably, decoupling improves RQ^{Th} by enhancing the shape of predicted objects, resulting in a higher number of true positives, which validates the effectiveness of our design choice.

Panoptic module voting radius. We analyze the impact of the majority voting radius r, as described in Sec. 3.3, using panoptic metrics. As shown in Fig. 4, performance initially improves as r increases, as a larger radius helps to smooth out predictions. However, beyond the optimal value of 9, performance declines because a larger r leads to incorrect ID assignments, as the majority voting starts considering irrelevant voxels. Notably, PQ^{Th} increases mostly due to RQ^{Th} as the majority voting mechanism smooths predictions and therefore increases the number of true positives.

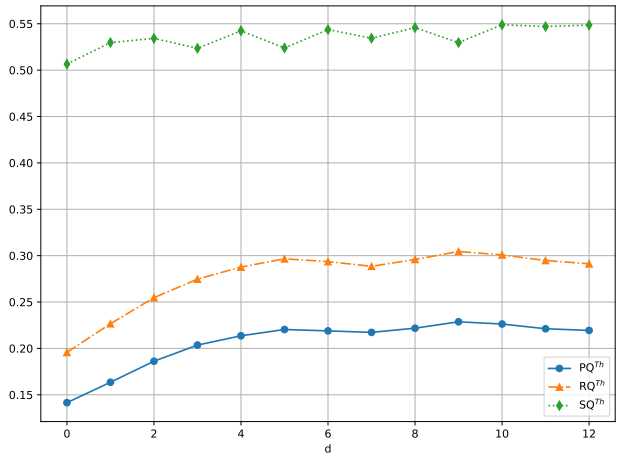

Figure 4. **Effect of the voting radius.** Increasing r initially improves performance by smoothing predictions. However, beyond the optimal value of 9, performance declines as the voting process starts to incorporate irrelevant voxels, leading to incorrect ID assignments. Tested on the mini set of nuScenes.

OffsetOcc complexity. Table 4 presents an analysis of the overhead introduced by our framework. Our framework introduces minimal overhead, making it an efficient extension to existing 3D semantic occupancy models. The parameter increase is negligible, with only 2.3 million (M) additional parameters. During training, memory consumption remains low, as we freeze the base model's weights and optimize only the *Object Module*, significantly reducing computational demands. During evaluation, memory usage increases due to the voting mechanism in the *Panoptic Module*. Despite this, the overall latency increase is modest, allowing the model to maintain efficiency for real-world deployment.

Method	Params	Memory	Latency	FPS
Baseline	44.2M	14.4G/2.5G	500 ms	2.0
Baseline w/ OffsetOcc	46.5M	2.2G/14.0G	714 ms	1.4

Table 4. **Overhead of our framework OffsetOcc.** Our framework is lightweight in both parameter count and latency while enabling panoptic occupancy. We show train/inference memory. Each frame includes 6 cameras. Tested on the Nvidia L40S GPU.

5. Conclusion

In this work, we propose a novel approach to 3D panoptic scene completion. In particular, we introduce a framework that leverages annotated occupancy maps to learn object shapes, a crucial aspect for path planning in autonomous driving contexts. Our method identifies objects and estimates their occupancy using a set of offsets relative to their positions. This approach enables the formulation of object shapes learning as a continuous and differentiable problem.

Limitations. The sequential nature of the Hungarian algorithm applied at the object level, introduces a bottleneck and increases the total training time. This issue is partially mitigated by increasing the number of training processes, such as using multiple GPUs, where each process handles a smaller subset of objects.

Future work. Our method highlights the potential of modeling object shapes as a continuous and differentiable problem, laying a foundation for future work in this area. Future work will focus on integrating temporal reasoning to improve frame-to-frame consistency, as well as refining hyperparameters and a more comprehensive *Panoptic Module* design to enhance panoptic performance.

Acknowledgments. We would like to thank Pierre-François De Plaen for his meaningful insights and comments. The resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government. We greatly acknowledge the Flanders AI Research program.

References

[1] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A mul-

timodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 6

[2] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3991–4001, 2022. 2, 3, 5, 7, 13

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 3, 4, 5

[4] Pengguang Chen, Shu Liu, Hengshuang Zhao, Xingquan Wang, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020. 6

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. IEEE Computer Society, 2009. 6

[6] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint*, 2023. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6

[8] Yulin He, Wei Chen, Tianci Xun, and Yusong Tan. Real-time 3d occupancy prediction via geometric-semantic disentanglement. *arXiv*, 2024. 2

[9] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv*, 2021. 7, 13

[10] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9223–9232, 2023. 2, 7, 13

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

[12] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Gaussianformer: Scene as gaussians for vision-based 3d semantic occupancy prediction. *arXiv*, 2024. 2

[13] M Kazhdan, M Bolitho, and H Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 3

[14] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9404–9413, 2019. 6

[15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 5

[16] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature 2015 521:7553*, 521:436–444, 2015. 1

[17] Heng Li, Yuenan Hou, Xiaohan Xing, Xiao Sun, and Yanyong Zhang. Occmamba: Semantic occupancy prediction with state space models. *arXiv*, 2024. 2

[18] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11809–11818, 2022. 7

[19] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M. Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. pages 9087–9098, 2023. 3

[20] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *ECCV 2022. Lecture Notes in Computer Science*, 13669:1–18, 2022. 1, 6, 7, 13

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

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 6

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 5

[24] Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin Wang. Fully sparse 3d occupancy prediction. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024. 2, 3, 7, 13

[25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 6

[26] Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong Zhang, and Yuan Xie. Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19936–19945, 2024. 2, 3, 7, 13

[27] Jiageng Mao, Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. 3d object detection for autonomous driving: A comprehensive survey. *International Journal of Computer Vision*, 131:1909–1963, 2023. 1

[28] Mingjie Pan, Li Liu, Jiaming Liu, Peixiang Huang, Longlong Wang, Shanghang Zhang, Shaoqing Xu, Zhiyi Lai, and Kuiyuan Yang. Uniocc: Unifying vision-centric 3d occupancy prediction with geometric and semantic rendering. *arXiv*, 2023. 3

[29] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang

Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 12404–12411, 2024. 3

[30] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Xingjian Du, Matteo Grella, G. V. Kranthi Kiran, Xuzheng He, Haowen Hou, Jiaju Lin, Przemysław Kazienko, Jan Kocon, Jiaming Kong, Bartłomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Guangyu Song, Xiangru Tang, Bolun Wang, Johan S. Wind, Stanisław Woźniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Qinghua Zhou, Jian Zhu, and Rui Jie Zhu. Rwkv: Reinventing rnns for the transformer era. In *Conference on Empirical Methods in Natural Language Processing*, pages 14048–14077. Association for Computational Linguistics (ACL), 2023. 2

[31] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 38 (3):1894–1914, 2021. 7

[32] Samuel Sze and Lars Kunze. Real-time 3d semantic occupancy prediction for autonomous vehicles using memory-efficient sparse convolution. In *IEEE Intelligent Vehicles Symposium*, pages 1286–1293. Institute of Electrical and Electronics Engineers Inc., 2024. 2

[33] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022. 1

[34] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36:64318–64330, 2023. 1, 2, 6, 7, 13

[35] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8406–8415, 2023. 2, 4, 6

[36] Guoqing Wang, Zhongdao Wang, Pin Tang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma. Occgen: Generative multi-modal 3d occupancy prediction for autonomous driving. *arXiv*, 2024. 3

[37] Junming Wang, Wei Yin, Xiaoxiao Long, Xingyu Zhang, Zebin Xing, Xiaoyang Guo, and Qian Zhang. Occrwkv: Rethinking efficient 3d semantic occupancy prediction with linear complexity. *arXiv*, 2024. 2

[38] Song Wang, Zhongdao Wang, Jiawei Yu, Wentong Li, Bailan Feng, Junbo Chen, and Jianke Zhu. Reliocc: Towards reliable semantic occupancy prediction via uncertainty learning. *arXiv*, 2024. 2

[39] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17850–17859, 2023. 2

[40] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17158–17168, 2024. 2, 3, 7, 13

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

[42] Guixing Xu, Wei Liu, Zuotao Ning, Qixi Zhao, Shuai Cheng, and Jiwei Nie. 3d semantic scene completion and occupancy prediction for autonomous driving: A survey. *2023 4th International Conference on Computers and Artificial Intelligence Technology, CAIT 2023*, pages 181–188, 2023. 2

[43] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022. 7

[44] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv*, 2023. 2

[45] Zichen Yu, Changyong Shu, Qianpu Sun, Junjie Linghu, Xiaobao Wei, Jiangyong Yu, Zongdai Liu, Dawei Yang, Hui Li, and Yan Chen. Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center. *arXiv*, 2024. 2, 3

[46] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9433–9443, 2023. 2, 7, 13

[47] Yanan Zhang, Jinqing Zhang, Zengran Wang, Junhao Xu, and Di Huang. Vision-based 3d occupancy prediction in autonomous driving: a review and outlook. *arXiv*, 2024. 2

[48] Linqing Zhao, Xiuwei Xu, Ziwei Wang, Yunpeng Zhang, Borui Zhang, Wenzhao Zheng, Dalong Du, Jie Zhou, and Jiwen Lu. Lowrankocc: Tensor decomposition and low-rank recovery for vision-based 3d semantic occupancy prediction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9806–9815, 2024. 2

[49] Yupeng Zheng, Xiang Li, Pengfei Li, Yuhang Zheng, Bu Jin, Chengliang Zhong, Xiaoxiao Long, Hao Zhao, and Qichao Zhang. Monoocc: Digging into monocular semantic occupancy prediction. *Proceedings - IEEE International Conference on Robotics and Automation*, pages 18398–18405, 2024. 2

[50] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 7

[51] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2020. 4, 6