PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science (2024) 92:483–498
https://doi.org/10.1007/s41064-024-00308-9

ORIGINAL ARTICLE

Self-Supervised 3D Semantic Occupancy Prediction from Multi-View 2D Surround Images

S. Abualhanud[1] · **E. Erahan**[2] · **M. Mehltretter**[1]

Received: 25 April 2024 / Accepted: 23 July 2024 / Published online: 18 September 2024

Abstract

An accurate 3D representation of the geometry and semantics of an environment builds the basis for a large variety of downstream tasks and is essential for autonomous driving related tasks such as path planning and obstacle avoidance. The focus of this work is put on 3D semantic occupancy prediction, i.e., the reconstruction of a scene as a voxel grid where each voxel is assigned both an occupancy and a semantic label. We present a Convolutional Neural Network-based method that utilizes multiple color images from a surround-view setup with minimal overlap, together with the associated interior and exterior camera parameters as input, to reconstruct the observed environment as a 3D semantic occupancy map. To account for the ill-posed nature of reconstructing a 3D representation from monocular 2D images, the image information is integrated over time: Under the assumption that the camera setup is moving, images from consecutive time steps are used to form a multi-view stereo setup. In exhaustive experiments, we investigate the challenges presented by dynamic objects and the possibilities of training the proposed method with either 3D or 2D reference data. Latter being motivated by the comparably higher costs of generating and annotating 3D ground truth data. Moreover, we present and investigate a novel self-supervised training scheme that does not require any geometric reference data, but only relies on sparse semantic ground truth. An evaluation on the Occ3D dataset, including a comparison against current state-of-the-art self-supervised methods from the literature, demonstrates the potential of our self-supervised variant.

Keywords 3D Occupancy Prediction · 3D Perception · NeRF · Semantic Scene Completion

1 Introduction

The ability to perceive the environment, considering its geometric and semantic properties in 3D, is fundamental in the context of autonomous driving to ensure safe navigation and to support decision making. By reconstructing an accurate model of the 3D world, autonomous vehicles can better understand their surroundings, estimate the positions of potential obstacles, and interact seamlessly with complex dynamic environments including other traffic participants, which marks a critical step toward fully autonomous driving systems. The 3D perception of vehicles has long been governed by sensors that allow for direct distance measurements, such as LiDAR sensors, which excel in creating fairly accurate geometrical maps of the environment. Despite their success, LiDAR systems present significant challenges, including high costs and the generation of sparse point clouds devoid of color information, limiting the scope of 3D environmental understanding. In recent years, a shift towards camera-based perception is noticeable in the literature in the context of autonomous driving. This shift is driven by the lower cost and the compactness of cameras as well as by the high resolution of the captured radiometric information. Latter is of particular importance for recognizing objects and understanding complex scenes. Related methods are either solely based on cameras or use them in conjunction with other sensors to perceive and interpret the surrounding.

A traditional task in 3D perception is 3D object detection and reconstruction, which involves detecting and clas-

✉ S. Abualhanud
abualhanud@ipi.uni-hannover.de

E. Erahan
eashwara.erahan@de.bosch.com

M. Mehltretter
mehltretter@ipi.uni-hannover.de

[1] Institute of Photogrammetry and Geoinformation, Leibniz University Hannover, Hannover, Germany

[2] Robert Bosch GmbH, Hildesheim, Germany

sifying objects and estimating their dimensions, shapes and poses within the 3D environment (El Amrani Abouelassad et al. 2023). However, methods addressing this task are commonly limited to certain object classes for which a model is defined or has been learned during training. So-called background classes, such as buildings and ground surfaces, are commonly neglected, although being highly relevant for a holistic understanding of an environment. In contrast, 3D semantic occupancy prediction aims to classify and reconstruct all parts of an environment. It entails reconstructing the geometry and semantics of a scene, either using an explicit representation, such as a voxel grid (Tong et al. 2023; Huang and Huang 2022a; Wei et al. 2023b), assigning each voxel an occupancy status and a semantic label, or using an implicit representation, where geometry and semantics are encoded in continuous fields (Liu et al. 2024b; Hayler et al. 2023). This approach, also referred to as semantic scene completion, goes beyond the reconstruction of surfaces visible in the images used as input. The aim is to reconstruct a complete three-dimensional geometric representation of the observed environment, i.e., including shape completion of partially visible objects. While Structure from Motion and image matching are related concepts, they merely predict the closest occupied point in view resulting in a 2.5D reconstruction, without providing a full 3D representation or retaining information about object dimensions. 3D occupancy prediction also differs from most Neural Radiance Field (NeRF) based methods (Mildenhall et al. 2021). Once trained, a 3D occupancy prediction method is capable of reconstructing a 3D representation from arbitrary images, while the original NeRF concept is based on learning a 3D model for a specific set of training images.

Commonly, 3D semantic occupancy prediction models are trained in a fully-supervised manner, using 3D ground truth data. Motivated by the high costs of producing and annotating 3D data, in the present paper, we explore a different supervision method using 2D instead of 3D labels. 2D labels are defined in the image plane, which facilitates interpretation and requires fewer annotations. With advancements in depth estimation and semantic segmentation, even a semi-automatic labeling approach using pre-trained models could be established. Following the concept of NeRF (Mildenhall et al. 2021), 2D reference data can be used to learn a voxel-based 3D occupancy representation by treating the latter as an explicit neural field. From this field, images are rendered in a differentiable way, which allows the comparison against 2D reference data and the back-propagation of gradients derived from the utilized loss function (Wimbauer et al. 2023; Pan et al. 2023). We build our method on the work of Pan et al. (2023), but differ from their work by addressing dynamic objects and the challenges that arise from their violation of the assumption of a static environment, which is commonly made in the context of NeRF. Taking care of dynamic objects is necessary, as images from multiple time steps are used for the 3D semantic occupancy prediction. Moreover, Pan et al. (2023) derive reference data for the geometry from LiDAR point clouds, while we present a self-supervised training scheme that does not require any geometric ground truth; sparse semantic labels are, however, required.

In summary, the main contributions of this work can be outlined as follows:

- The development of a mask-based method to handle dynamic objects, such as cars and bicycles, to ensure multi-view consistency between images of multiple time steps in the case of 2D supervision. This is achieved by selectively discarding observations on dynamic objects only if they are actually in motion while being observed within the scene.
- The development of a self-supervised training scheme for reconstructing the 3D geometry of an observed environment. The presented approach is based on dense pseudo depth labels and ensures a balanced ray generation across different classes in the presence of sparse semantic labels.
- A comprehensive evaluation of the developed method, including comparisons of the geometric and semantic quality of variants trained with 3D, 2D and self-supervision as well as against two other self-supervised methods from the literature.

2 Related Work

2.1 Image-Based 3D Perception

Due to the limitations of LiDAR sensors, including high costs, sparsity, and the lack of color information, research related to 3D perception in the context of assisted and autonomous driving is increasingly focusing on camera-based methods such as image-based 3D object detection (Huang et al. 2021; Reading et al. 2021; Li et al. 2023a) and Bird's Eye View (BEV) semantic segmentation (Philion and Fidler 2020; Roddick and Cipolla 2020). In the latter approach, 2D observations from multiple images are transformed into a consistent 2D planimetric representation, neglecting height information. Using multi-view surround images, the transformation into a BEV representation is particularly challenging, as these images typically overlap only slightly or not at all. Methods addressing this challenge can be categorized into two main streams: forward projection and transformer-based backward projection.

In forward projection, Philion and Fidler (2020) learn per-pixel categorical depth distributions to transform image features into a pseudo 3D point cloud per image. Using pillar pooling (Lang et al. 2019), the features from multiple

images are merged into a unified discrete representation. To increase the accuracy of the BEV representation, Li et al. (2023b) enhance the depth prediction and Li et al. (2023a) build on temporal information to estimate depth and to integrate BEV feature maps from different timestamps. The second approach, transformer-based backward projection (Li et al. 2022; Wang et al. 2022; Liu et al. 2022), involves projecting 3D points or BEV queries back onto the 2D image plane to gather features. Following this approach, Li et al. (2022) project BEV queries to the image plane and employ spatio-temporal attention to aggregate features across various views from multiple timestamps, resulting in a comprehensive BEV representation.

2.2 3D Occupancy Prediction

Many previous camera-based 3D perception methods have focused on the detection (Huang et al. 2021; Reading et al. 2021; Huang and Huang 2022a; Li et al. 2023a,b) and shape estimation (El Amrani Abouelassad et al. 2023) of objects in 3D. While being effective for identifying and reconstructing objects of specific classes, these methods rely on a descriptive object model that has been defined or learned beforehand. Consequently, they often struggle with unique shapes and irregular structures and neglect so-called background classes, such as building facades and ground surfaces. In contrast, voxel-based occupancy prediction aims to provide an explicit representation of the environment by subdividing it into a voxel grid and by predicting the occupancy status of every voxel. Some methods also include the estimation of a semantic label per voxel, a task also known as semantic scene completion. This approach can still reconstruct meaningful geometry even when semantic classes are misidentified.

Semantic scene completion often relies on 3D observations such as LiDAR point clouds (Rist et al. 2021) or image-based depth information (Garbade et al. 2019; Li et al. 2019). In (Cao and De Charette 2022) however, a scene is reconstructed from a single monocular image by lifting the 2D features extracted with a 2D U-Net (Ronneberger et al. 2015) to all possible 3D locations along the viewing rays of the corresponding pixels. This 3D representation is then processed by a 3D U-Net (Ronneberger et al. 2015) to infer semantic labels. However, since single images commonly have a limited field of view, such a setup does not allow for a comprehensive scene reconstruction adequate for downstream tasks like path planning and obstacle avoidance, leading to a shift towards multi-view surround image settings.

In the context of 3D occupancy prediction from multi-view surround images, Huang et al. (2023b) use sparse LiDAR points for supervision, resulting in sparse occupancy predictions. Wei et al. (2023b) generate 3D voxel features at multiple scales using a transformer-based approach and combine them through transposed convolutional up-sampling. Moreover, a pipeline is introduced to generate dense semantic occupancy labels from sparse LiDAR point clouds. (Liu et al. 2024a) extends the BEV-based method of Li et al. (2022) for occupancy prediction, while (Ding et al. 2023) considers the temporal context by integrating multiple images over time to generate multi-scale 3D voxel features. Lastly, (Li et al. 2023c) combines both, the forward and backward projection approaches, for enhanced and more efficient occupancy prediction. However, these methods require 3D ground truth data during training, which is typically expensive to acquire, as it depends on additional sensors like LiDAR for generation and requires extensive human annotations.

Addressing this limitation, (Pan et al. 2023) employs a NeRF-like (Mildenhall et al. 2021) rendering approach to enable training with 2D reference data. In this method, the 3D occupancy map is divided into a density and a semantic field, both being predicted by a Multi-layer perceptron (MLP). This network is trained using 2D labels, specifically depth and semantic maps. Although this approach benefits from the less costly 2D labels compared to 3D ones, it still relies on geometric reference data which is captured with an additional sensor. An alternative strategy involves training a 3D occupancy prediction model in a self-supervised manner using photometric losses, i.e., optimizing for photometric consistency between pixels from multiple images that refer to the same object point. (Wimbauer et al. 2023) predicts a density field from monocular images after being trained with stereo and fisheye images. (Hayler et al. 2023) advances this concept by also predicting semantics, utilizing pseudo semantic labels during training that are generated with a pre-trained semantic segmentation model. SelfOcc (Huang et al. 2023a) extends self-supervision from single monocular images to a multi-view surround setting, employing BEV-based backward projection (Li et al. 2022) to predict a 3D representation. OccNeRF (Zhang et al. 2023) focuses on multi-frame photometric consistency to improve occupancy prediction and uses predicted 2D bounding boxes (Liu et al. 2023) and segmentation masks (Kirillov et al. 2023), as in Ren et al. (2024), to derive dense 2D semantic labels. Both, SelfOcc and OccNeRF, are considered in the evaluation in Sect. 5 to set the results achieved with the method presented in this work into context. Despite being cost-effective, current self-supervised methods perform significantly worse compared to their counterparts trained in a supervised manner in terms of the accuracy of the predicted semantic occupancy representation. Thus, further investigations are required to close this gap, motivating the methodology proposed in the present paper.

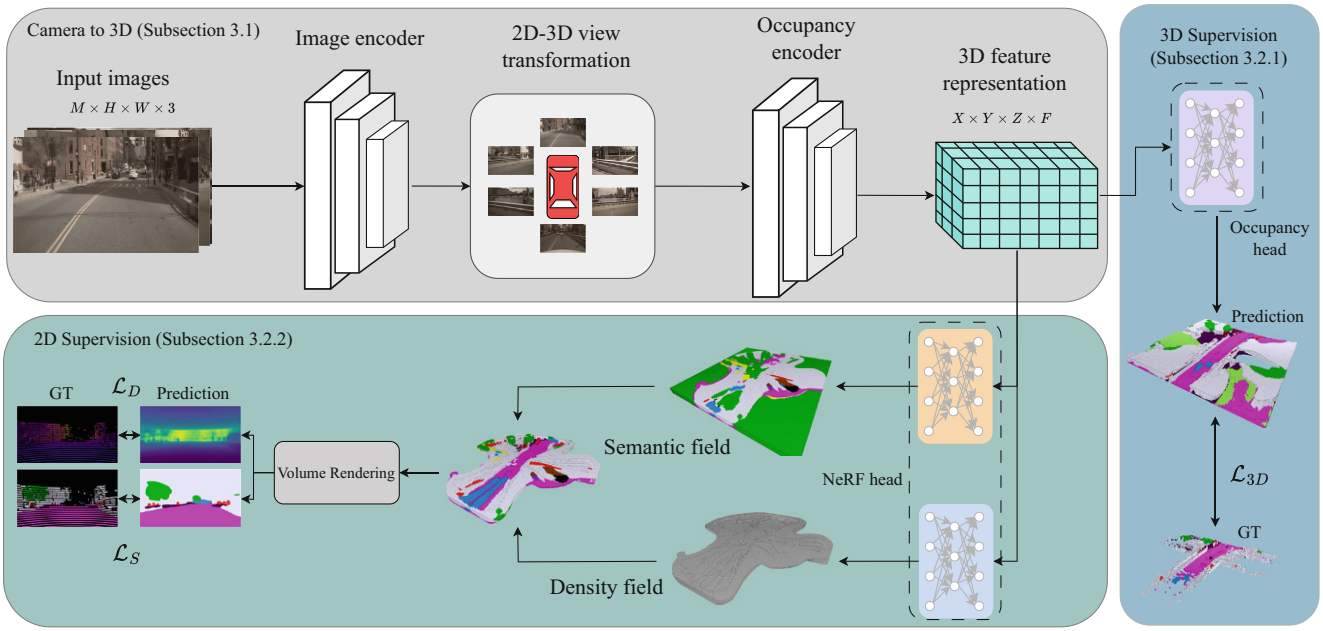

Fig. 1 This figure illustrates the workflow of the overall network, beginning with the extraction of 2D features from each input image. Subsequently, these features are transformed into a unified 3D space and are further encoded to generate a 3D feature representation. The resulting representation is input into either an occupancy head, consisting of an MLP to predict semantic occupancy maps, for the 3D supervision case based on 3D ground truth, or into a NeRF head consisting of two MLPs tasked with predicting the density and semantic fields. From these predictions, depth and semantic maps are rendered and compared to 2D reference data. The RGB input images are taken from Caesar et al. (2020)

3 Methodology

Given a set of color images $\mathcal{I}$, the goal of this work is to reconstruct a semantically enriched, explicit 3D representation of an observed environment in form of a voxel grid, in which every voxel has assigned an occupancy and a semantic class label. The images in $\mathcal{I}$ all have a size of $H \times W$ pixels and are captured by N_I cameras. The cameras are rigidly mounted on a vehicle and arranged in a multi-view surround setting, i.e., the viewing axes of the cameras are parallel to the ground and the images capture a field of view of 360° with only minimal overlap. In addition, $\mathcal{I}$ contains images captured at N_T different points in time t, so that $\mathcal{I}$ consists of $M = N_T \cdot N_I$ images in total. Here, we set $N_T = 3$ and use the current and two previous frames, where each frame consists of N_I images. Alongside the images, the associated interior and exterior orientation parameters (latter being defined in a vehicle centric coordinate system) and the pose of the ego vehicle defined in a global coordinate system at each time step t are given as input.

We subdivide this reconstruction task into the following steps: First, features are extracted from the images in $\mathcal{I}$, transferred into a 3D space and information from all images are aggregated to generate a coherent 3D feature representation (explained in Sect. 3.1). This 3D representation is further processed to produce an occupancy map that also encodes semantic information (explained in Sect. 3.2). Training our method can be carried out with 3D supervision (Sect. 3.2.1) or using NeRF-based rendering with either 2D reference data (Sect. 3.2.2) or employing our novel self-supervised training scheme (Sect. 3.2.3). An overview of the complete method is given in Fig. 1.

3.1 Camera to 3D

In the first step, features are extracted from each image in $\mathcal{I}$ separately, before these features are transformed into a 3D space, where they are fused with the features from all other images to obtain a coherent 3D feature representation. For this purpose, the approach described in Huang and Huang (2022a) and Li et al. (2023a) is adopted, employing temporal stereo matching and intermediate depth supervision for depth estimation. The result of this first step is a voxel-based 3D feature representation V_F (see Fig. 1):

$$g(\mathcal{I}) = V_F \in \mathbb{R}^{X \times Y \times Z \times F}, \tag{1}$$

where X, Y, Z refer to the three spatial dimensions in object space and F refers to the number of feature channels. This process g of obtaining a coherent 3D feature representation is the same for all variants of our method, the 3D supervised variant and the 2D supervised variants. Details on g are given in the following.

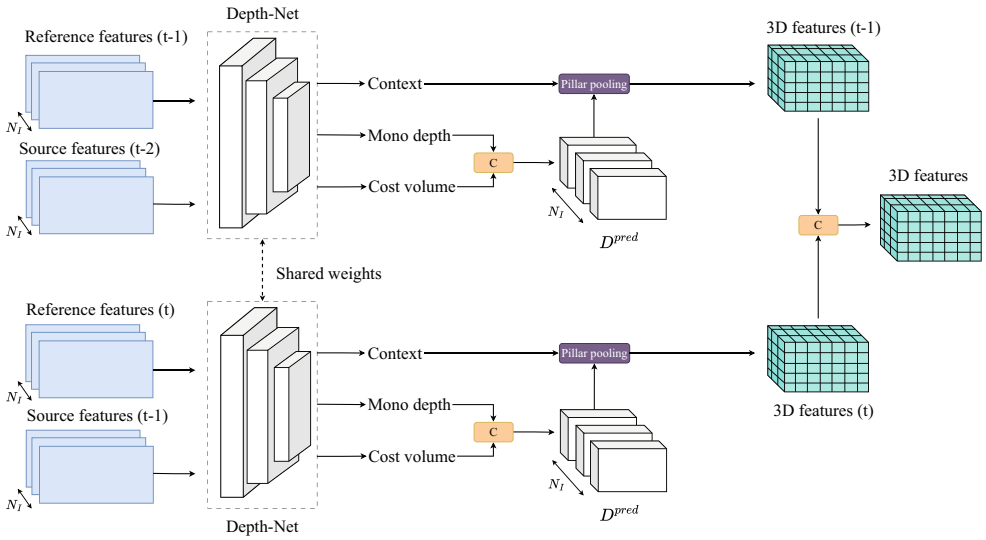

Fig. 2 The figure illustrates the part of the network for the 2D to 3D view transformation. Initially, 2D image features are extracted via an image encoder. These features are processed by Depth-Net, which predicts context features and monocular depth for the reference image. Additionally, a cost volume is generated through stereo matching with an image from a previous time step and is concatenated with the monocular depth map yielding D^{pred}. Subsequently, a pillar pooling operation takes D^{pred} and the context features from all views to construct a 3D feature representation that maintains coherence across all views. This figure is inspired by Li et al. (2023a)

3.1.1 Image Encoder

In our work, the Swin Transformer (Liu et al. 2021) is used to extract features from each image in $\mathcal{I}$, for its proven suitability in 3D and BEV perception (Pan et al. 2023; Huang and Huang 2022a). Features at different levels of abstraction are extracted: Shallow layers capture low-level details like edges and textures, while deeper layers encode high-level information like objects and their spatial relationships. The features from the different layers are combined through a Feature Pyramid Network (FPN) (Lin et al. 2017), so that the following processing steps can make use of both, low-level details and high-level understanding, to reconstruct and interpret the observed scene.

3.1.2 2D-3D View Transformation

The second part of our method focuses on the 2D to 3D transformation, which converts 2D local image features from multiple cameras and different time steps into a consistent 3D representation centered around the ego vehicle. This transformation is designed to integrate the spatial and temporal context to construct a comprehensive 3D feature representation. This transformation is carried out in two steps: First, depth information is derived from the images using a neural network-based component that we call Depth-Net. The depth information is used to transform the 2D features to a camera specific 3D coordinate system. Second, pillar pooling is used to fuse the features from images taken by different cameras and to obtain a consistent feature representation in 3D object space.

Depth-Net The component of our method tasked with depth and context prediction is referred to as Depth-Net. The implementation of this component is inspired by previous studies (Huang et al. 2021; Li et al. 2023b,a; Huang and Huang 2022a). Depth-Net takes as input the feature maps extracted from the N_I images of a reference time step t and from the N_I images of the previous time step $t-1$ (see Fig. 2), referred to as reference and source features, respectively. Depth-Net outputs 2D contextual features $F^{2d} = \{F_i^{2d} \in \mathbb{R}^{C_F \times H \times W}, i = 1, 2, \ldots, N_I\}$, with feature dimension C_F, and predicts monocular depth for every image of the reference time step. Additionally, Depth-Net performs dense matching on stereo images. For this purpose, a cost volume, encoding feature dissimilarity, is computed for every pair of reference and source feature maps that correspond to images taken by the same camera, which results in N_I cost volumes.

The monocular depth estimation and the cost volume corresponding to a camera are then concatenated to obtain $D^{\text{pred}} = \{D_i^{\text{pred}} \in \mathbb{R}^{C_D \times H \times W}, i = 1, 2, \ldots, N_I\}$, with depth dimension C_D. While stereo image-based matching allows for an accurate estimation of depth via forward intersection of rays, it relies on the assumption that the scene is static in the time between capturing the stereo images. Using images from different time steps, this approach is affected by moving objects, leading to incorrect depth estimates in the respective image regions. On the other hand, monocular depth remains unaffected by the motion of dynamic objects, as it relies on a single image only, but commonly delivers less accurate depth estimates. During training, Depth-Net receives supervision from the task head, i.e., the occupancy

or the NeRF head (cf. Fig. 1), as well as through intermediate supervision of the predicted depth by Depth-Net following Li et al. (2023b).

Pillar Pooling Given the context F_i^{2d} and depth features D_i^{pred}, a 3D feature representation is obtained following the approach presented in Philion and Fidler (2020):

$$F_i^{3d} = F_i^{2d} \otimes D_i^{\text{pred}}, \quad F_i^{3d} \in \mathbb{R}^{C_F \times C_D \times H \times W}, \tag{2}$$

where $\otimes$ represents the pixel-wise Kronecker product. An enriched point cloud F_i^{3d} is initially given per camera. The point clouds are then transformed into a global coordinate system, which, in this context, is defined by the ego vehicle's pose at the current time step. This transformation is based on the given parameters of interior $K_i \in \mathbb{R}^{3\times3}$ and exterior orientation $T_i \in \mathbb{R}^{4\times4}$ of a camera i, latter being defined with respect to a vehicle centric coordinate system. The integration of the individual point clouds is accomplished through a method called pillar pooling (Lang et al. 2019; Huang and Huang 2022b). In this context, the 3D space is divided into voxels, and features associated with points located within a voxel are aggregated by applying sum pooling (Huang and Huang 2022b). This process results in a 3D representation in which each voxel has assigned a feature vector, effectively capturing the spatial information from the different perspectives and fusing it into a coherent 3D representation (see Fig. 2).

3.1.3 Occupancy Encoder

The 3D features produced by the 2D-3D view transformation are further processed by a 3D encoder. A 3D ResNet-based architecture (He et al. 2016) is selected to derive multi-scale features within the 3D space, with an FPN (Lin et al. 2017) employed to fuse these multi-scale features. This encoding step encourages richer 3D features, by considering relations between feature vectors of adjacent voxels. Moreover, during the transformation of the 2D features into a 3D representation, certain voxels might be sparsely or not covered by any points, resulting in voxels with a poor or missing feature vector (Fan et al. 2023; Yu et al. 2023). The convolutional blocks within the occupancy encoder are designed to mitigate this issue. We refer to the result of the occupacy encoder as V_F (cf. Eq. 1).

3.2 3D Occupancy Prediction

The goal of the second part of our method is to transform the obtained 3D voxel-based feature representation V_F into an occupancy representation that also includes semantic information. Depending on the kind of supervision used during training, a different functional model is employed to realize this part: Conventionally, this process is learned via fully-supervised training using 3D ground truth data, a process that will be detailed in Sect. 3.2.1. Alternatively, supervising the training can also be performed on a 2D level by rendering 2D depth and semantic maps from this 3D representation. In this case, the loss term is computed using 2D ground truth labels. This approach will be elaborated on in Sect. 3.2.2. Lastly, building on the process of 2D supervision, we present a novel method for learning to estimate the geometry of the observed scene in a self-supervised manner. In this case, only sparse 2D semantic labels are needed as reference data. This approach is described in Sect. 3.2.3.

3.2.1 3D Supervision

The straight forward approach to guiding the network in learning to predict 3D occupancy involves the usage of 3D ground truth data for both, geometry and semantics. According to (Huang and Huang 2022a), the 3D features V_F (cf. Eq. 1) are converted into a semantic occupancy map using a feed-forward MLP, to which we refer as the occupancy head. The occupancy head estimates voxel logits $V_L \in \mathbb{R}^{X\times Y\times Z\times N_C}$ for each class C, with N_C referring to the total number of classes. These classes account for semantic categories, with an additional class indicating whether a voxel is occupied or not.

Optimization The output of the occupancy head V_L is compared against the 3D ground truth using the cross-entropy as loss function $\mathcal{L}_{occ}$. The overall training loss for the 3D supervised model is given as:

$$\mathcal{L}_{3D} = \lambda_{occ}\mathcal{L}_{occ} + \lambda_{lss}\mathcal{L}_{LSS}, \tag{3}$$

where $\mathcal{L}_{LSS}$ is the intermediate depth loss, which is a binary cross-entropy loss from Li et al. (2023b), where the predicted depth from Depth-Net is compared to LiDAR ground truth depth. λ_{occ} and λ_{lss} are hyper-parameters weighting the two loss terms.

3.2.2 2D Supervision

In the 2D supervision case, 2D labels, namely depth and semantic segmentation maps, are utilized to train our method in estimating a 3D semantic occupancy map. In this context, we follow the approach presented in Pan et al. (2023) and process V_F with a so-called NeRF head instead of using the occupancy head. The NeRF head enables the estimation of 2D geometric and semantic maps through volume rendering (Mildenhall et al. 2021). As the latter is differentiable, the gradient information obtained by comparing the 2D estimation against 2D ground truth data can be back-propagated to the 3D voxel-based representation, allowing to

learn a 3D semantic occupancy map from 2D reference data. More specifically, V_F is transformed into two components, accomplished using two separate MLPs: a density field $V_\sigma \in \mathbb{R}^{X\times Y\times Z}$ and a semantic field $V_S \in \mathbb{R}^{X\times Y\times Z\times(N_C-1)}$ (see Fig. 1).

Guided Ray Generation The 2D supervision of the network based on NeRF (Mildenhall et al. 2021) requires the generation of viewing rays to render 2D depth and semantic segmentation maps from V_σ and V_S. If ray generation were random, a disproportionate number of rays would be allocated to larger objects, such as buildings and roads, as they occupy a larger share of an image compared to smaller objects such as pedestrians and bicycles. In consequence, the latter would be underrepresented in the generated rays, which would negatively impact learning the estimation of geometry and semantics of these classes. To address this issue, we utilize the approach detailed in Pan et al. (2023), which generates rays during training based on class specific weights. According to (Pan et al. 2023), these weights are determined based on the class frequency in the training set and are assigned to a ray r as follows:

$$W_b(r) = \exp\left(\lambda_s \cdot \left(\frac{\max(B)}{B_c(r)} - 1\right)\right), \tag{4}$$

where B is a set of class frequencies, $B_c(r)$ is the class frequency for class c associated to ray r and λ_s is a smoothing factor.

Volume Rendering Depth and semantic segmentation maps are derived from the density V_σ and semantic fields V_S via volume rendering (Sun et al. 2022). The density σ and class logits s corresponding to a given sample location x are calculated via trilinear interpolation (Fridovich-Keil et al. 2022; Sun et al. 2022) between the neighboring voxels in the 3D space.

For a set of K sample points $\{x_k\}_{k=1}^K$ along a ray r, the alpha composition α and the accumulated transmittance τ are calculated according to the original NeRF approach (Mildenhall et al. 2021):

$$\alpha(x_k) = 1 - \exp(-\sigma(x_k)\delta_k), \tag{5}$$

$$\tau(x_k) = \exp\left(-\sum_{i=1}^{k-1}\sigma(k_i)\delta_i\right), \tag{6}$$

where $\delta_k = x_{k+1} - x_k$ is the distance between two adjacent sample points. The process for rendering a pixel in a semantic segmentation $\hat{S}$ and a depth map $\hat{D}$ from a ray r also aligns with the original NeRF method (Mildenhall et al. 2021), which involves accumulating the samples along the ray:

$$\hat{D}(r) = \sum_{k=1}^{K}\tau(x_k)\alpha(x_k)x_k, \tag{7}$$

$$\hat{S}(r) = \sum_{k=1}^{K}\tau(x_k)\alpha(x_k)s(x_k)\,. \tag{8}$$

$\hat{D}$ and $\hat{S}$ are compared against 2D depth and semantic reference labels using a cross-entropy loss $\mathcal{L}_S$, following Pan et al. (2023). Meanwhile, the scale-invariant logarithmic loss $\mathcal{L}_D$ (Eigen et al. 2014; Pan et al. 2023), commonly used in the context of depth prediction, is utilized to supervise the training of our method with respect to the rendered depth:

$$D_{\text{diff}} = \frac{1}{N_\mathcal{R}}\sum_{r\in\mathcal{R}}(\log\hat{D}(r) - \log D(r)), \tag{9}$$

$$\mathcal{L}_D = \sqrt{D_{\text{diff}}^2 - \varphi\,(D_{\text{diff}})^2}, \tag{10}$$

where for a set of rays $\mathcal{R}$, $N_\mathcal{R}$ is the total number of rays, φ is a hyper-parameter referred to as variance focus which shifts the focus of the loss function towards pixels with higher error variance in log space. $D(r)$ is the reference depth for the pixel associated to a ray r.

Temporal Context Due to the limited spatial overlap of images within a surround camera setup, rays from different cameras are unlikely to intersect with one another. As a result, the majority of voxels are observed from a single image only. However, redundant observations from multiple images are important to learn multi-view consistency, i.e., the consistency of appearance and geometry of objects across views, to obtain an accurate 3D representation. To obtain redundant observations within a surround camera setup, we follow Pan et al. (2023) and integrate over time. This entails considering images from different time steps as if they were taken at the same time. For this purpose, we transform all observations into a common coordinate system which we define as the vehicle centric coordinate system at time step t. To transform an image observation from time step $t-1$, which is defined in a vehicle centric coordinate system, to this common coordinate system, we have to account for the motion ${}^tT_{t-1}$ of the vehicle between the two time steps, leading to modified parameters of exterior orientation $T_i^* = {}^tT_{t-1}\cdot T_i$. Within the context of volume rendering, this entails transforming the rays to be with re-

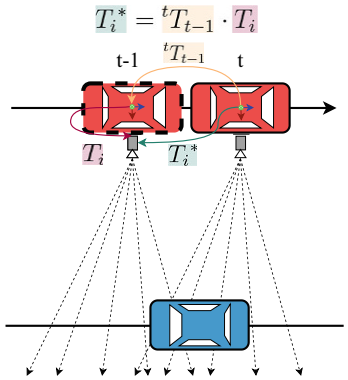

Fig. 3 A visual illustration demonstrating the use of temporal frames to increase the redundancy of observations. All cameras are transformed relative to the coordinate system of the current frame (t) based on the pose of the ego vehicle. This ensures that the 3D scene reconstruction aligns with the current frame. ${}^tT_{t-1}$ describes the movement of the ego vehicle between frames $t-1$ and t. T_i refers to the parameters of exterior orientation of a camera i with respect to a vehicle centric coordinate system. $T_i{}^*$ refer to the modified exterior orientation parameters, which are given with respect to the pose of the ego vehicle in the current frame (t)

spect to the current time step. For an illustration, refer to Fig. 3.

While this approach is suitable for static components of a scene, it presents difficulties with moving objects. The movement of such entities leads to ray misalignments, resulting in intersections at wrong locations. Such misalignments could significantly disrupt the training process. An approach suggested by Pan et al. (2023) involves omitting the generation of temporal rays, i.e., rays corresponding to image observations from previous frames, for pixels associated with dynamic objects, i.e., objects that can generally move. With the availability of semantic labels, classes prone to movement, such as cars, pedestrians and trucks, can be pinpointed. Whenever a pixel is found to correspond to one of these dynamic classes, no temporal rays are generated for this pixel; for the current frame, all generated rays are used during training independent of whether they intersect with a dynamic object.

Although this approach works relatively well since the majority of the scene is typically static, entirely disregarding certain classes for the multi-view consistency is not ideal. In particular, as dynamic objects hold significant importance in autonomous driving applications. Therefore, we suggest a novel, alternative approach to handle dynamic objects: We selectively ignore dynamic objects by determining their movement status. We exclude them only when they are in motion but still consider them when they remain stationary. Our approach entails creating binary dynamic masks prior to training, using the object positions in the global coordinate system and the dimensions of their 3D bounding boxes as in Tancik et al. (2023). The velocity of objects between images of successive time steps is determined by computing the numerical derivative of their positions over time. Using a predefined velocity threshold, objects in the scene are classified as moving when their velocities surpass this limit. In this case, the 3D bounding box is projected onto the image plane to generate a binary mask Mask (see Fig. 4), where the area corresponding to moving objects, is assigned a value of 1. The temporal weight for ray r is now given by:

$$W_t(r) = \begin{cases} 0, & \text{if Mask}(r) = 1 \\ \lambda_{\text{temp}}, & \text{if Mask}(r) = 0 \end{cases}, \tag{11}$$

where λ_{temp} is a hyper-parameter and the overall weight for a ray r is now given by:

$$W(r) = W_b(r) \cdot W_t(r), \tag{12}$$

where based on this weighting, rays are randomly selected. This method, which involves initially sampling rays based on class balance weights, is preferred over other techniques, e.g., applying weights at the loss level, as it increases the likelihood that rays corresponding to all semantic classes are included. This procedure ensures class balance already on the level of ray generation and not only at the level of the loss computation, where a very imbalanced set of rays may disturb the optimization process despite applying class depend weights.

Optimization The network is optimized by minimizing a composite loss, consisting of multiple weighted losses. The training loss for the 2D supervision is given by:

$$\mathcal{L}_{2D} = \lambda_d \mathcal{L}_D + \lambda_s \mathcal{L}_S + \lambda_e \mathcal{L}_E + \lambda_{\text{dist}} \mathcal{L}_{\text{dist}} + \lambda_{lss} \mathcal{L}_{LSS}, \tag{13}$$

where the entropy loss $\mathcal{L}_E$ (Sun et al. 2022) and the distortion loss $\mathcal{L}_{\text{dist}}$ (Barron et al. 2022) serve as regularization terms of the occupancy field, as in Pan et al. (2023), and their weights λ_e and λ_{dist} are defined as hyper-parameters. At inference time, NeRF rendering is bypassed. Instead, 3D occupancy is directly determined from the density and semantic fields by establishing a threshold to identify areas

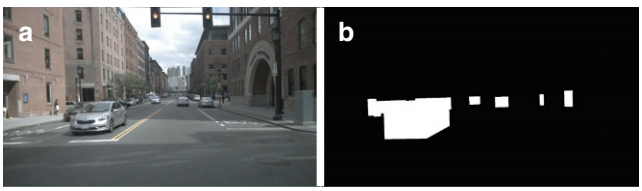

Fig. 4 This figure presents an example of masking moving objects to guide the generation of rays. **a** displays the RGB image, while **b** shows the corresponding binary mask. The RGB image is from Caesar et al. (2020)

considered occupied. Each occupied voxel gets assigned the semantic label with the highest probability among the logits of this voxel.

3.2.3 Self-Supervision

Lastly, we introduce a novel self-supervised training scheme, allowing to train our method, introduced in Sect. 3.2.2, without requiring any ground truth for the 3D geometry. For this purpose, the proposed training scheme relies on pseudo depth labels, instead of depth labels generated from ground truth LiDAR point clouds. These pseudo labels are produced with the method presented in Wei et al. (2023a). This method is trained using photometric losses, i.e., it is optimized with respect to photometric consistency between pixels in different images that depict the same 3D object point. In consequence, this method does not rely on any 3D measurements, e.g., captured with a LiDAR sensor, during training or inference.

Utilizing these depth maps instead of those derived from LiDAR point clouds offers a unique advantage: While their absolute accuracy may not match that of LiDAR measurements, no additional sensor is needed to obtain reference data. In addition, these pseudo depth maps are dense, i.e., they provide a depth estimate for every pixel, whereas LiDAR measurements are commonly sparse. This density supports the training process of our method, as more observations on the geometry are provided per voxel.

To utilize the additional depth labels effectively, we propose that additional training rays are generated. The challenge lies in avoiding the random generation of rays across the depth map to prevent the predominance of image-dominating classes – the challenge we have pointed out early in Sect. 3.2.2. Since we continue to employ sparse semantic labels as reference, for many pixels, the weight of the corresponding rays based on the class frequency cannot be directly determined as outlined in Eq. (4), as there is no corresponding semantic label for them. To overcome this limitation, we assign every pixel a semantic label by applying nearest neighbor interpolation based on the sparse semantic label map. For an illustration of this approach, refer to Fig. 5. Using the pseudo depth labels as a reference, the same optimization procedure is applied during training as for the variant of our method with 2D supervision, with the densified semantic segmentation maps guiding the ray generation.

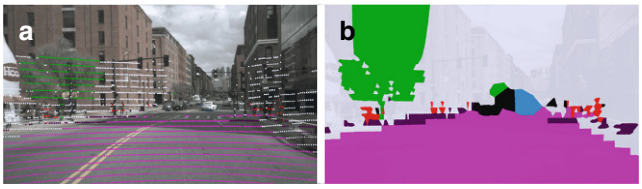

Fig. 5 This figure shows an example of densifying the sparse semantic segmentation map, employing a nearest neighbor interpolation strategy to assign class labels to unidentified pixels within an image. **a** displays the sparse semantic labels, with points enlarged for visibility, while **b** presents the resulting densified semantic segmentation map. The RGB image is from Caesar et al. (2020)

4 Experimental Setup

4.1 Evaluated Variants

In our experiments, we train and test four different variants of the presented method: The first variant of our method is trained using 3D ground truth for geometry and semantics and is referred to as "3D". The variant referred to as "2D" is trained with 2D ground truth. The reference data is generated from the projection of a semantically annotated 3D LiDAR point cloud onto the image plane. "2D Mask" is trained with the same reference data as "2D", but incorporates dynamic masks to guide the ray generation as described in Sect. 3.2.2. The final variant, "2D Pseudo", is equal to "2D Mask", but substitutes the depth labels derived from a LiDAR point cloud with pseudo depth labels generated with a pre-trained model named Surrounddepth (Wei et al. 2023a). However, we still utilize sparse semantic labels during training. This variant serves as a demonstration of the potential for self-supervised occupancy prediction by eliminating reliance on LiDAR sensor data during training. We compare our method to state-of-the-art self-supervised 3D occupancy prediction models, namely OccNeRF (Zhang et al. 2023) and SelfOcc (Huang et al. 2023a), where both, our and their methods, do not depend on LiDAR ground truth to retrieve geometric information. The numeric results for OccNeRF (Zhang et al. 2023) and SelfOcc (Huang et al. 2023a) are taken from the respective publications.

4.2 Datasets

For all experiments, the inputs consist of images, interior and exterior orientation parameters, and the pose of the ego vehicle over time, all obtained from the nuScenes dataset (Caesar et al. 2020). The dataset includes image sequences of 1000 scenes, each lasting 20 seconds, and are annotated at a 2 Hz frequency. The dataset is divided into 700 training scenes, 150 for validation, and 150 for testing. Its sensor setup includes $N_I = 6$ cameras for a 360° view and a 32-beam LiDAR. The dataset also captures different

Table 1 Comparison of mean and per-class IoU scores. See Sect. 4.1 for a description of the evaluated variants. As OccNeRF does not consider the "others" and "other flat" classes, mIoU* only considers the remaining classes. The metric values shown for SelfOcc (Huang et al. 2023a) and OccNeRF (Zhang et al. 2023) are taken from their respective papers. The best results per type of supervision (3D, 2D, and self-supervised) are bold. All results are given as [%]

Model	mIoU ↑	mIoU* ↑	Others	Barrier	Bicycle	Bus	Car	Construction vehicle	Motorcycle	Pedestrian	Traffic cone	Trailer	Truck	Drivable surface	Other flat	Sidewalk	Terrain	Manmade	Vegetation
3D	42.0	43.9	11.6	50.0	26.5	52.1	54.7	26.0	27.4	28.9	26.0	37.3	42.2	82.4	43.5	54.4	57.8	49.1	43.5
2D	24.4	25.2	**3.2**	26.2	12.1	21.3	**22.2**	**16.5**	11.1	13.7	13.7	**22.7**	**21.5**	67.3	32.3	43.7	44.3	**19.0**	**23.3**
2D Mask	**25.1**	**26.1**	2.0	**26.8**	**16.0**	**23.0**	20.9	15.4	**15.1**	**14.6**	**17.8**	21.7	21.0	**67.4**	**33.0**	**44.0**	**45.4**	18.8	**23.3**
2D Pseudo	**19.4**	**20.1**	**1.4**	**17.7**	**12.1**	**14.1**	**15.2**	**12.3**	**10.0**	**10.6**	**11.3**	**13.5**	**14.1**	**58.9**	**27.4**	**38.2**	**38.5**	15.3	**19.1**
OccNeRF	–	10.8	–	0.8	0.8	5.1	12.4	3.5	0.2	3.1	1.8	0.5	3.9	52.6	–	20.8	24.7	**18.4**	13.1
SelfOcc	9.3	10.5	0.0	0.1	0.6	5.4	12.5	0.0	0.8	2.1	0.0	0.0	8.2	55.4	0.0	26.3	26.5	14.2	5.6

times of the day and weather situations, including sunny, rainy, and night scenarios. In the case of the 2D supervised models, except for the self-supervised one, training utilizes their LiDAR point clouds to generate the 2D depth labels. In all 2D supervised models, sparse 2D semantic maps generated from their semantic point clouds are used. For the 3D supervised model, training is conducted using the Occ3D dataset (Tian et al. 2024), which extends nuScenes by voxel-based 3D ground truth for occupancy and semantics. The dataset differentiates between 17 semantic classes; an overview over these classes is given in Table 1. Additionally, the Occ3D dataset (Tian et al. 2024) is used for evaluating all models. The LiDAR point cloud data from nuScenes (Caesar et al. 2020) is further used for evaluating the depth prediction.

4.3 Implementation Details

The input images are rescaled to a size of 512×1408 pixels. The predicted 3D voxel occupancy grid is of dimensions $200 \times 200 \times 16$, which covers a range from $-40\,\text{m}$ to $40\,\text{m}$ along the x and y axes, and from $-1\,\text{m}$ to $5.4\,\text{m}$ along the z-axis around the ego vehicle in 3D object space. Consequently, each voxel within this grid has a resolution of $0.4\,\text{m}$ in all three dimensions. These values are consistent with the ground truth data from Tian et al. (2024). The values for λ_{occ}, λ_d, λ_s, λ_e, λ_{dist} and λ_{lss} in Eq. (3) and (13) are respectively assigned as 1, 1, 1, 0.01, 0.01 and 0.05. Only for "2D Pseudo", λ_{lss} is set to zero. The velocity threshold for generating the binary masks explained in Sect. 3.2.2 is set to $0.75\,\text{m}\,\text{s}^{-1}$. The training process is standardized across all experiments with the same random seed. The training is conducted over 12 epochs with a batch size of 16, utilizing 4 NVIDIA Tesla A100 GPUs. Each batch consists of $M = 18$ images: $N_I = 6$ images from $N_T = 3$ time steps (a current frame and the two previous frames). AdamW (Loshchilov and Carmon 2019) is utilized as optimizer, with a learning rate of 1×10^{-4} and a weight decay of 1×10^{-2}. The training process typically spans approximately 2–3 continuous days. For all variants of our method except the self-supervised one "2D Pseudo", the network parameters are initialized with values obtained from pretraining for 3D object detection Huang and Huang (2022a). In the case of the self-supervised variant, the parameters of the Swin Transformer (Liu et al. 2021) used as image encoder are initialized with values obtained from pretraining on ImageNet (Deng et al. 2009), all other parameters are randomly initialized.

4.4 Evaluation Metrics

For the evaluation of the predictions, we utilize metrics that measure both the semantics and occupancy, specifically the per-class Intersection over Union (IoU) and the Mean Intersection over Union (IoU) over all classes:

$$\text{IoU}_C = \frac{\text{TP}_C}{\text{TP}_C + \text{FP}_C + \text{FN}_C}, \tag{14}$$

$$\text{mIoU} = \frac{1}{N_C} \sum_{c=1}^{N_C} \text{IoU}_c, \tag{15}$$

where TP_C, FP_C, and FN_C represent the true positives, false positives, and false negatives for class C, respectively, and N_C is the number of classes. Additionally, we employ metrics that focus solely on evaluating the predicted geometry, including the completeness (Comp), accuracy (Acc), and F-score:

$$\text{Comp} = \text{mean}_{p \in P} \left(\mathbb{1} \left(\text{dist}(p, \hat{P}) < 0.6 \right) \right), \tag{16}$$

$$\text{Acc} = \text{mean}_{\hat{p} \in \hat{P}} \left(\mathbb{1} \left(\text{dist}(\hat{p}, P) < 0.6 \right) \right), \tag{17}$$

$$\text{F-score} = \frac{2 \times \text{Acc} \times \text{Comp}}{\text{Acc} + \text{Comp}}, \tag{18}$$

where $\hat{P}$ is the set of predicted points, P is the set of ground truth points. To evaluate these metrics, the occupied voxels are first converted into a point cloud using the center of each voxel. $\mathbb{1}(\cdot)$ is an indicator function that returns 1 if its argument is true and 0 otherwise. dist$(\cdot)$ calculates the nearest neighbor Minkowski distance. Furthermore, we assess the predicted depth using the Absolute Relative Difference (Abs Rel), Squared Relative Difference (Sq Rel), Root Mean Squared Error (RMSE), and Root Mean Squared Logarithmic Error (RMSE log):

$$\text{Abs Rel} = \frac{1}{|D|} \sum_{d \in D} \frac{\left| d - \hat{d} \right|}{\hat{d}}, \tag{19}$$

$$\text{Sq Rel} = \frac{1}{|D|} \sum_{d \in D} \frac{\left\| d - \hat{d} \right\|^2}{\hat{d}}, \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{|D|} \sum_{d \in D} \left\| d - \hat{d} \right\|^2}, \tag{21}$$

$$\text{RMSE log} = \sqrt{\frac{1}{|D|} \sum_{d \in D} \left\| \log d - \log \hat{d} \right\|^2}, \tag{22}$$

where D represents the ground truth depth map, d denotes the ground truth depth at a single pixel, and $\hat{d}$ signifies the depth predicted at that pixel.

5 Results and Analysis

The results are organized according to the kind of reference data used during training: We start with an analysis of the variant that is trained with 3D ground truth data in Sect. 5.1. In Sect. 5.2, the two variants trained with 2D ground truth data are evaluated. Finally, the self-supervised variant is assessed in Sect. 5.3.

Table 2 Evaluation of the quality of the geometry predicted by the different variants of our method. See Sect. 4.4 for a description of the evaluated variants. IoU_{Occ} is for the class "occupied", which entails all occupied voxels

	3D	2D	2D Mask	2D Pseudo
Completeness ↑ [%]	**91.80**	85.77	85.47	81.14
Accuracy ↑ [%]	**89.44**	64.81	64.73	57.10
F-Score ↑ [%]	**90.27**	73.25	73.11	66.47
IoU_{Occ} ↑ [%]	**72.71**	46.09	46.00	39.21

5.1 3D Supervision

Analyzing the numeric results presented in Tables 1 and 2, it is evident that the variant trained with 3D reference data performs best in both aspects, the correct estimation of semantics and geometry. This is to be expected as this kind of reference is most informative and even contains information on parts of the environment that are not observed in any of the images used as input. The latter further enables the "3D" variant to learn the completion of shapes for specific classes of objects, as can be seen in Fig. 6.

Since images taken at different time steps are used as input to the presented method, moving objects are particularly challenging to be reconstructed. Thus, we analyze the results for such objects in more detail: From Fig. 6a and b, it is evident that the feature map generated from a single frame (utilizing two frames to create stereo depth, with the depth output being used to predict one feature map for the reference frame, as shown in Fig. 2) has low feature activations in certain areas, e.g., on the road and in occluded regions. This suggests that those regions are not important to the network. On the other hand, the close surrounding

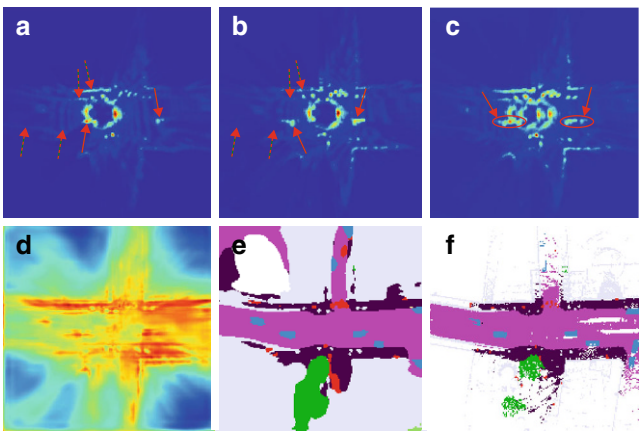

Fig. 6 This illustration shows our method's behavior from constructing a 3D feature representation to predicting the 3D occupancy for the 3D supervised case. The feature maps in this figure represent aggregated features from the 3D feature map along the z-axis, for better visualization. **a** Represents the previous feature map, **b** shows the current feature map, **c** shows the concatenation of both feature maps, **d** shows the convolved concatenated feature map, **e** shows the predicted semantic occupancy from a BEV, and **f** represents the ground truth semantic occupancy from a BEV. Dashed arrows highlight when a feature is missing in one feature map but present in another. Solid arrows direct attention to the positions of dynamic objects across feature maps. Ellipses are used to indicate the duplication of dynamic objects within the concatenated feature map. In the visualized feature maps, red corresponds to high activation, while blue indicates low activation. For color coding details of the visualized semantic occupancy maps, refer to Table 1

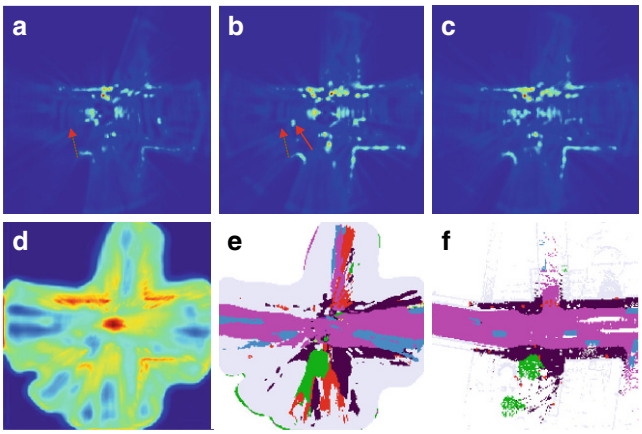

Fig. 7 This illustration shows our method's behavior from constructing a 3D feature representation to predicting the 3D occupancy for the 2D supervised case. For details, refer to the description of Fig. 6. **a** Previous feature map, **b** Current feature map, **c** Concatenated feature map, **d** Convolved feature map, **e** Predicted occupancy, **f** Ground truth occupancy

of the ego vehicle (in the center) and specific object instances receive clearly higher attention by the network. The concatenation of the individual feature maps, considering the movement of the ego vehicle between the two time steps, results in a richer feature map, i.e., the attention of the network is more distributed (see Fig. 6c). While the static elements within both feature maps are aligned due to considering the ego motion, dynamic objects present challenges due to their unaccounted motion, leading to replicated appearances of dynamic objects in the concatenated feature map. Despite not explicitly addressing dynamic object motion, this variant is implicitly able to compensate for this motion in the semantic occupancy predictions, avoiding duplicates or stretched shapes of dynamic objects (see Fig. 6e). This outcome might be due to the 3D supervision approach, which focuses not just on locating objects and reconstructing the observed parts of their shape, but also optimizes for correctly reconstructing occluded parts.

5.2 2D Supervision

Fig. 7 shows that the variant trained using 2D labels generally predicts an accurate representation of the scene, in particular for parts of the environment being visible from the perspective of the ego vehicle. While Table 1 shows promising quantitative results for variant "2D", a comparison to the results of the "3D" variant reveals a clear gap across all semantic classes. This gap is mainly due to two reasons: The model encounters challenges in accurately reconstructing certain objects, especially the ones belonging to classes of potentially dynamic objects. The reconstruction of parts of the environment that are not observed in any of the input images is clearly worse (cf. Figs. 6e and 7e), often extending objects indefinitely. This indicating that the completion task is harder to be learned in the case of 2D supervision, as a reference is only available for observed parts of the environment and not also for occluded parts as in the case of 3D supervision. However, the network can sometimes predict what is behind another object, constructing a complete 3D representation. This mainly occurs when the network has information on these parts of the scene from previous frames (Fig. 7a), leveraging concatenated 3D features to fill in missing observations in the current frame (Fig. 7b).

Additionally, this issue also likely arises from the fact that the concept of multi-view consistency, although being crucial for a proper 3D object representation, is not sufficiently learned, due to the lack of redundancy in the observations. Although observations from multiple time steps somewhat mitigate this issue, they do not benefit dynamic classes. Introducing temporal rays for dynamic classes aims to increase the redundancy and thus the probability of intersecting rays, increasing the number of voxels that are observed multiple times. Qualitatively, the variant "2D Mask" with temporal rays for dynamic classes (see Fig. 8) surpasses the one completely excluding dynamic classes "2D", better capturing dynamic object shapes and reducing false negatives. Quantitatively, improvements are seen in certain dynamic classes (see Table 1), though some classes show slight declines, possibly due to the IoU metric focusing on voxel intersection rather than shape accuracy.

Although the results have not reached the same level as relying on 3D supervision, we see promising progress and will further investigate this direction in future work. Further focusing on multi-view consistency could involve directly enforcing it, with the aim to avoid reliance on random multi-view observations of a voxel. One strategy, inspired by Wimbauer et al. (2023), involves rendering the current frame from temporally adjacent frames by projecting ray samples of the current frame onto the temporally adjacent frames. However, in our case, this would require dense semantic and depth labels, as each projected sample needs to be associated with a labeled pixel. This requirement further motivates the focus of our following experiment on training with dense pseudo depth labels in a self-supervised manner.

5.3 Self-Supervision

The variant "2D Pseudo" which is trained with a dense pseudo depth map shows lower performance with respect to the quality of the semantic and geometric estimates compared to the other variants (see Tables 1 and 2). Yet, having in mind that no 3D measurements are needed as reference to train this variant and that with a RMSE of 5.66 meters the pseudo depth maps are significantly worse than 3D LiDAR measurements (see Table 3), the obtained results can still be

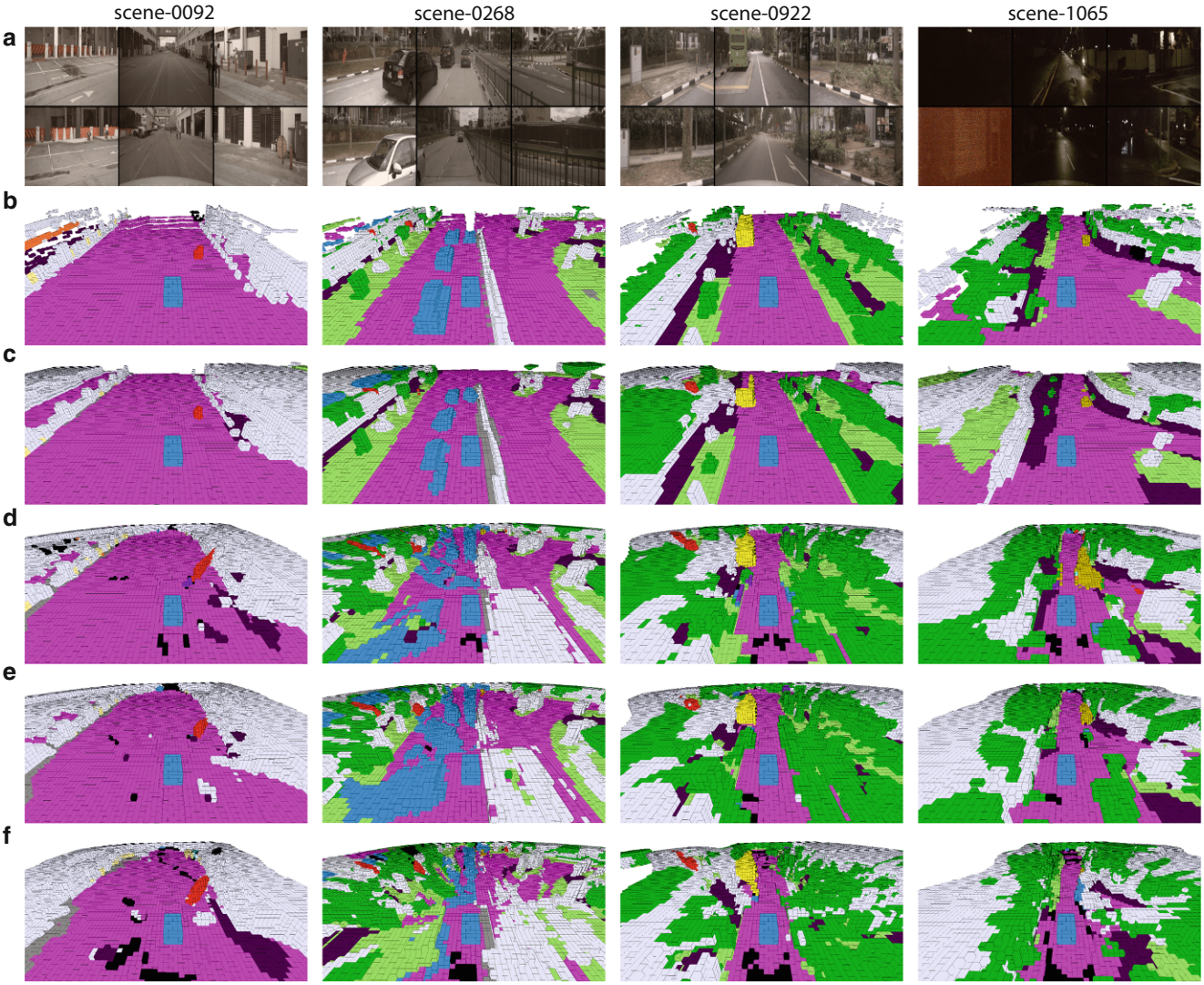

Fig. 8 Visualization of the semantic occupancy predictions of the different variants of our method. **a** shows the RGB input images, **b** shows the ground truth, **c–f** show the semantic occupancy maps predicted by the "3D", "2D", "2D Mask", and "2D Pseudo" variants of our method, respectively. The blue object, in the center of all scenes, represents the ego vehicle. For color coding details, refer to Table 1. The RGB images are taken from Caesar et al. (2020)

Table 3 Comparison of the depth prediction metrics. See Sect. 4.1 for a description of the evaluated variants of our method. SurroundDepth (Wei et al. 2023a), a depth estimation method we use as basis for training out self-supervised variant, is reevaluated for depth up to 40 m as a comparison. RMSE and RMSE log are given as [m]

Model	Abs Rel ↓	Sq Rel ↓	RMSE ↓	RMSE log ↓
2D	0.27	1.58	5.54	0.34
2D Mask	**0.26**	**1.52**	**5.48**	**0.33**
2D Pseudo	0.31	2.13	6.22	0.40
SurroundDepth	0.27	4.05	5.66	0.34

seen as a success. The latter is further supported by a comparison to other state-of-the-art self-supervised method for semantic occupancy prediction: The results in Table 1 show that our self-supervised variant "2D Pseudo" demonstrates better performance compared to both OccNeRF (Zhang et al. 2023) and SelfOcc (Huang et al. 2023a). It's important to note that they utilize dense semantic pseudo-labels, whereas we continue to use sparse ground truth semantic labels.

We further assess the depth maps rendered from the density field (see Fig. 9) for both 2D supervised and the self-supervised variants, by comparing them against the results of a state-of-the-art depth estimation method form the literature called SurroundDepth (Wei et al. 2023a). Given that the ground truth point cloud range is up to 40 m in x and y

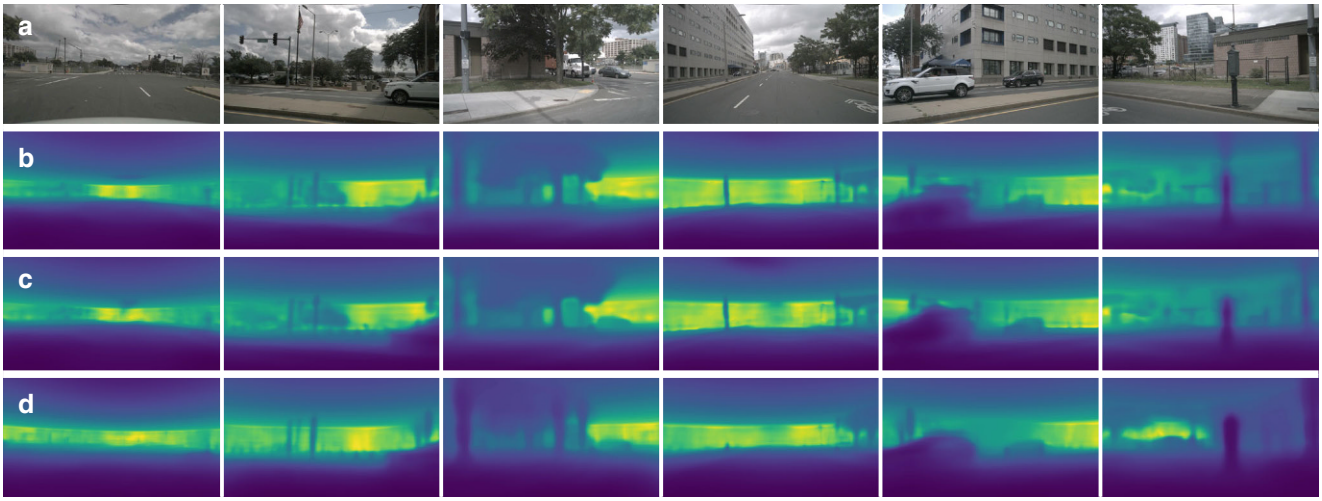

Fig. 9 Rendered depth maps from the predicted density field for the various variants of our method. **a** shows the RGB input images. **b–d** show the predicted depth for "2D", "2D Mask" and "2D Pseudo", respectively. The rendered depth maps appear blurry, which can (at least partially) be attributed to the trilinear interpolation technique employed during the rendering process. In the visualized depth maps, parts of the scene closer to the camera are depicted in blue, while those farther away appear in yellow. The RGB images are taken from Caesar et al. (2020)

direction, we specifically compare their performance within this range with that of SurroundDepth (Wei et al. 2023a), which has been reevaluated for depth values up to 40 m, using their checkpoint. Numerical results on this comparison are given in Table 3. Note that the "3D" variant is not considered in this comparison, as no NeRF-like density field is estimated, which prevents us from deriving a depth map in a straight forward manner. From Tables 2 and 3, we observe that the depth metrics deliver results for all variants that are comparable to the results of SurroundDepth (Wei et al. 2023a), with some variants even showing slight improvements, specifically those trained with 2D ground truth labels. The geometric metrics demonstrate the superiority of the 3D supervised variant "3D" in accurately predicting geometry, while "2D Pseudo" performs significantly worse in predicting accurate geometry due to the imprecise nature of its depth labels.

Future enhancements in the direction of self-supervision could focus on improving the estimated depth labels, possibly by increasing the temporal context, using images from more time steps. Moreover, we will explore approaches that consider the rather imprecise nature of pseudo depth labels during training, e.g., by not treating these labels as ground truth in the loss computation, but allowing for some margin of error (Deng et al. 2022).

6 Conclusions

In this work, we presented a Convolutional Neural Network-based method to predict the semantic and geometric properties of an environment in a 3D voxel representation from multiple color images arranged in a surround-view setup with minimal overlap. The focus of our contributions were put on addressing the challenges that arise from dynamic objects when using images taken at different time steps and on learning to reconstruct 3D geometry in a self-supervised manner. In exhaustive experiments we evaluated different variants of this method that are trained either fully-supervised using 3D or 2D reference data or in a self-supervised manner.

While the 2D-supervised models performed adequately in general, clear limitations could be seen with regards to the reconstruction of parts of the scene that are not sufficiently covered by observations from multiple images. While being important for learning to reconstruct complete shapes of objects, we assume that the concept of multi-view consistency is not properly learned in this case. In addition, we did not model the motion of dynamic objects in the scene, but simply discarded related observations if an object was moving, to avoid introducing errors caused by erroneously matching image rays. However, accounting for the motion of moving objects, e.g., using scene flow and following the concepts of Boeder et al. (2024), could further improve the results; a direction that we will investigate in future work. Moreover, we utilized annotations from the nuScenes dataset (Caesar et al. 2020) for the 3D bounding boxes to identify moving objects, but plan to eliminate this dependency by predicting optical flow between consecutive frames in future work. We further believe that enforcing multi-view consistency, rather than relying on the chance observation of a voxel from multiple views, could be beneficial.

Our training scheme for learning 3D reconstruction in a self-supervised manner proved to be efficient and outperformed the current state-of-the-art from the literature.

However, while this scheme does not require any geometric reference data, it relies on sparse semantic labels. To overcome this limitation and to make our training scheme completely self-supervised, dense semantic maps could also be predicted using pre-trained models, rather than depending on sparse nuScenes (Caesar et al. 2020) annotations. Despite the good results of our self-supervised variant, we maintain that a deeper exploration into 2D supervision, even with ground truth labels, is essential before fully embracing self-supervised techniques. The discrepancy between 2D and 3D supervision remains notable and warrants further investigation.

Acknowledgements We express our sincere gratitude to Robert Bosch GmbH and the Institute of Artificial Intelligence at Leibniz University Hannover for their generous provision of computing resources, which greatly facilitated the completion of this work.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest The authors declare that there is no conflict of interest.

References

Barron JT, Mildenhall B, Verbin D, Srinivasan PP, Hedman P (2022) Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5470–5479

Boeder S, Gigengack F, Risse B (2024) Occflownet: Towards self-supervised occupancy estimation via differentiable rendering and occupancy flow. arXiv preprint arXiv:240212792

Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, Krishnan A, Pan Y, Baldan G, Beijbom O (2020) nuscenes: a multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11621–11631

Cao AQ, De Charette R (2022) Monoscene: monocular 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3991–4001

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, pp 248–255

Deng K, Liu A, Zhu JY, Ramanan D (2022) Depth-supervised nerf: fewer views and faster training for free. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12882–12891

Ding Y, Huang L, Zhong J (2023) Multi-scale occ: 4th place solution for cvpr 2023 3d occupancy prediction challenge. arXiv preprint arXiv:230611414

Eigen D, Puhrsch C, Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. In Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, pp 2366–2374. MIT Press, Cambridge, MA, USA

El Amrani Abouelassad S, Mehltretter M, Rottensteiner F (2023) Vehicle pose and shape estimation in UAV imagery using a CNN. In: ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences X-1/W1-2023, pp 935–944 https://doi.org/10.5194/isprs-annals-X-1-W1-2023-935-2023

Fan L, Yang Y, Wang F, Wang N, Zhang Z (2023) Super sparse 3d object detection. IEEE Trans Pattern Anal Mach Intell 45(10):12490–12505

Fridovich-Keil S, Yu A, Tancik M, Chen Q, Recht B, Kanazawa A (2022) Plenoxels: radiance fields without neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5501–5510

Garbade M, Chen YT, Sawatzky J, Gall J (2019) Two stream 3d semantic scene completion. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 416–425 https://doi.org/10.1109/CVPRW.2019.00055

Hayler A, Wimbauer F, Muhle D, Rupprecht C, Cremers D (2023) S4c: Self-supervised semantic scene completion with neural fields. arXiv preprint arXiv:231007522

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778

Huang J, Huang G (2022a) Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:220317054

Huang J, Huang G (2022b) Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. arXiv preprint arXiv:221117111

Huang J, Huang G, Zhu Z, Ye Y, Du D (2021) Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:211211790

Huang Y, Zheng W, Zhang B, Zhou J, Lu J (2023a) Selfocc: Self-supervised vision-based 3d occupancy prediction. arXiv preprint arXiv:231112754

Huang Y, Zheng W, Zhang Y, Zhou J, Lu J (2023b) Tri-perspective view for vision-based 3d semantic occupancy prediction. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 9223–9232

Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY et al (2023) Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 4015–4026

Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) Pointpillars: fast encoders for object detection from point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12697–12705

Li J, Liu Y, Yuan X, Zhao C, Siegwart R, Reid I, Cadena C (2019) Depth based semantic scene completion with position importance aware loss. IEEE Robot Autom Lett 5(1):219–226

Li Y, Bao H, Ge Z, Yang J, Sun J, Li Z (2023a) Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. Proc AAAI Conf Artif Intell 37:1486–1494

Li Y, Ge Z, Yu G, Yang J, Wang Z, Shi Y, Sun J, Li Z (2023b) Bevdepth: acquisition of reliable depth for multi-view 3d object detection. Proc AAAI Conf Artif Intell 37:1477–1485

Li Z, Wang W, Li H, Xie E, Sima C, Lu T, Qiao Y, Dai J (2022) Bevformer: Learning bird's-eye-view representation from multi-cam-

camera images via spatiotemporal transformers. In: European conference on computer vision. Springer, pp 1–18

Li Z, Yu Z, Austin D, Fang M, Lan S, Kautz J, Alvarez JM (2023c) Fb-occ: 3d occupancy prediction based on forward-backward view transformation. arXiv preprint arXiv:230701492

Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2117–2125

Liu J, Zhang S, Kong C, Zhang W, Wu Y, Ding Y, Xu B, Ming R, Wei D, Liu X (2024a) Occtransformer: Improving bevformer for 3d camera-only occupancy prediction. arXiv preprint arXiv:240218140

Liu L, Wang B, Xie H, Liu D, Liu L, Tian Z, Yang K, Wang B (2024b) Surroundsdf: Implicit 3d scene understanding based on signed distance field. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 21614–21623

Liu S, Zeng Z, Ren T, Li F, Zhang H, Yang J, Li C, Yang J, Su H, Zhu J et al (2023) Grounding dino: Marrying dino with grounded pre-training for open-set object detection. arXiv preprint arXiv:230305499

Liu Y, Wang T, Zhang X, Sun J (2022) Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. Springer, pp 531–548

Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10012–10022

Loshchilov I, Carmon Y (2019) Decoupled weight decay regularization. In: International Conference on Learning Representations

Mildenhall B, Srinivasan PP, Tancik M, Barron JT, Ramamoorthi R, Ng R (2021) Nerf: representing scenes as neural radiance fields for view synthesis. Commun ACM 65(1):99–106

Pan M, Liu J, Zhang R, Huang P, Li X, Liu L, Zhang S (2023) Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. arXiv preprint arXiv:230909502

Philion J, Fidler S (2020) Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: Computer Vision–ECCV 2020: 16th European Conference Glasgow, August 23–28, 2020. Proceedings, Part XIV 16. Springer, pp 194–210

Reading C, Harakeh A, Chae J, Waslander SL (2021) Categorical depth distribution network for monocular 3d object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 8555–8564

Ren T, Liu S, Zeng A, Lin J, Li K, Cao H, Chen J, Huang X, Chen Y, Yan F et al (2024) Grounded sam: Assembling open-world models for diverse visual tasks. arXiv preprint arXiv:240114159

Rist CB, Emmerichs D, Enzweiler M, Gavrila DM (2021) Semantic scene completion using local deep implicit functions on lidar data. IEEE Trans Pattern Anal Mach Intell 44(10):7205–7218

Roddick T, Cipolla R (2020) Predicting semantic map representations from images using pyramid occupancy networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 11138–11147

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference Munich, October 5-9, 2015. proceedings, part III 18. Springer, pp 234–241

Sun C, Sun M, Chen HT (2022) Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 5459–5469

Tancik M, Weber E, Ng E, Li R, Yi B, Kerr J, Wang T, Kristoffersen A, Austin J, Salahi K, Ahuja A, McAllister D, Kanazawa A (2023) Nerfstudio: A modular framework for neural radiance field development. In: ACM SIGGRAPH 2023 Conference Proceedings SIGGRAPH '23.

Tian X, Jiang T, Yun L, Mao Y, Yang H, Wang Y, Wang Y, Zhao H (2024) Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. Adv Neural Inf Process Syst 36:64318–64330

Tong W, Sima C, Wang T, Chen L, Wu S, Deng H, Gu Y, Lu L, Luo P, Lin D et al (2023) Scene as occupancy. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 8406–8415

Wang Y, Guizilini VC, Zhang T, Wang Y, Zhao H, Solomon J (2022) Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In: Conference on Robot Learning PMLR, pp 180–191

Wei Y, Zhao L, Zheng W, Zhu Z, Rao Y, Huang G, Lu J, Zhou J (2023a) Surrounddepth: entangling surrounding views for self-supervised multi-camera depth estimation. In: Conference on Robot Learning PMLR, pp 539–549

Wei Y, Zhao L, Zheng W, Zhu Z, Zhou J, Lu J (2023b) Surroundocc: multi-camera 3d occupancy prediction for autonomous driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 21729–21740

Wimbauer F, Yang N, Rupprecht C, Cremers D (2023) Behind the scenes: Density fields for single view reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9076–9086

Yu Z, Shu C, Deng J, Lu K, Liu Z, Yu J, Yang D, Li H, Chen Y (2023) Flashocc: fast and memory-efficient occupancy prediction via channel-to-height plugin. arXiv preprint arXiv:231112058

Zhang C, Yan J, Wei Y, Li J, Liu L, Tang Y, Duan Y, Lu J (2023) Occnerf: self-supervised multi-camera occupancy prediction with neural radiance fields. arXiv preprint arXiv:231209243