LangOcc: Open Vocabulary Occupancy Estimation via Volume Rendering

Simon Boeder
Robert Bosch GmbH
simon.boeder@de.bosch.com

Fabian Gigengack
Robert Bosch GmbH
fabian.gigengack@de.bosch.com

Benjamin Risse
University of Münster
b.risse@uni-muenster.de

Abstract

The 3D occupancy estimation task has become an important challenge in the area of vision-based autonomous driving recently. However, most existing camera-based methods rely on costly 3D voxel labels or LiDAR scans for training, limiting their practicality and scalability. Moreover, most methods are tied to a predefined set of classes which they can detect. In this work we present a novel approach for open vocabulary occupancy estimation called LangOcc, that is trained only via camera images, and can detect arbitrary semantics via vision-language alignment. In particular, we distill the knowledge of the strong vision-language aligned encoder CLIP into a 3D occupancy model via differentiable volume rendering. Our model estimates vision-language aligned features in a 3D voxel grid using only images. It is trained in a weakly-supervised manner by rendering our estimations back to 2D space, where features can easily be aligned with CLIP. This training mechanism automatically supervises the scene geometry, allowing for a straight-forward and powerful training method without any explicit geometry supervision. LangOcc outperforms LiDAR-supervised competitors in open vocabulary occupancy with a mAP of 22.7 *by a large margin (*$+4.3\%$*), solely relying on vision-based training. We also achieve a mIoU score of* 11.84 *on the Occ3D-nuScenes dataset, surpassing previous vision-only semantic occupancy estimation methods (*$+1.71\%$*), despite not being limited to a specific set of categories.*

1. Introduction

Object detection is a fundamental task in autonomous driving, enabling vehicles to understand and navigate their surroundings. Traditionally, these tasks have been trained on predefined sets of classes, limiting their ability to fully comprehend complex and dynamic environments. To overcome this limitation, recent advancements have introduced 3D occupancy estimation, a popular method that represents scene geometry using a voxel grid [5, 8, 17, 41, 51]. This approach allows for geometry-based *generic* object detection, enabling autonomous vehicles to perceive any structure in their environment. However, most existing 3D occupancy estimation methods rely on expensive 3D ground-truth labels [15, 25, 50]. This requirement poses a significant challenge, as acquiring accurate 3D labels for large-scale datasets is both resource-intensive and impractical. Moreover, existing benchmarks usually only reflect a limited predefined set of classes. Consequently, there is a pressing need for novel methods to efficiently train occupancy models without relying on 3D labels, for example via self-supervised or camera-only learning. While some efforts have been made to avoid voxel labels, they either still necessitate labeled LiDAR point clouds or involve complex pseudo ground-truth generation techniques [14, 34, 48]. Furthermore, despite the ability to capture any geometry, the semantic understanding of these methods remains tied to a predefined set of classes. These limitations hinder the adaptability and flexibility of autonomous systems in comprehending diverse and evolving environments.

In this paper we propose a novel weakly-supervised occupancy estimation method which aligns geometric estimations with open vocabulary natural language features, hence allowing representations of any semantics and therefore eliminating the need for 2D or 3D semantic labels. To achieve this, we leverage the power of the popular CLIP model [37] and distill its representational power into 3D space through volume rendering. In particular, instead of predicting the probabilities of predefined classes, our model estimates vision-language aligned features per voxel. The model is trained by rendering these features in a differentiable manner from the 3D voxel space back to the 2D image space, where they are supervised by features precomputed by the off-the-shelf vision-language encoder CLIP [37]. The source code is available under https://github.com/boschresearch/langocc.

In summary, our contributions are:

- **Open vocabulary occupancy:** A novel vision-only architecture to model arbitrary geometries and semantics by aligning the semantic feature space with natural language, hence decoupling occupancy representations from predefined semantic class definitions.
- **Weakly-supervised learning:** Inspired by NeRF [32],

LangOcc trains language features and 3D scene geometry jointly and eliminates the need for 3D ground-truth labels and can be trained with images only. The learning targets are automatically derived from images using a pretrained CLIP model, bringing our approach close to self-supervised learning. The model generalizes to estimate any geometry and semantics without per-scene optimization like NeRF-approaches.

- **Feature subspace learning:** In addition we introduce a specialised dimensionality reduction strategy to increase segmentation performance when a set of task-specific classes is available.
- **State-of-the-art performance:** LangOcc outperforms competitors on open vocabulary occupancy estimation by a large margin, and achieves state-of-the-art results in camera-only semantic occupancy estimation.

2. Related Work

Three different lines of work are particularly relevant for our proposed method, namely object detection, occupancy estimation and open vocabulary perception.

2.1. Camera-based 3D Object Detection

Vision-based 3D object detection is crucial for autonomous applications and a widely studied field. Most recent approaches transform extracted 2D image features from a single or multiple views into a common 3D space (e.g. a Birds-Eye-View grid; BEV) where objects boxes are estimated. One popular approach is to lift 2D image features into 3D by estimating a depth distribution [12, 13, 35], while other methods such as [18, 26] project learned 3D queries onto the image plane to sample features. Methods like [30, 44] do not explicitly project features to 3D but instead follow an object-centric approach. All of these methods are typically trained to detect specific types of objects, and therefore do not provide a comprehensive understanding of the entire scene. This limitation has prompted the exploration of the more general occupancy estimation paradigm, which aims to perceive and understand the complete geometry of the scene.

2.2. Camera-based 3D Occupancy Estimation

Vision-based occupancy prediction, also known as semantic scene completion, involves estimating a dense representation of the 3D scene in terms of geometry and semantics from a set of input images [1, 41, 45]. Pioneering works on 3D occupancy estimation extend the well-known concepts of object detection to 3D space, e.g., by lifting the BEV into a voxel grid [4, 12, 15, 25, 42]. Following approaches mostly focus on efficient supervision [29, 31, 46], label efficiency [2, 8, 34] or performance improvements via specific model designs [16, 27, 38, 50, 52]. As the 3D occupancy prediction task is inherently complex, most models rely on 3D ground truth data, which can be resource-intensive to obtain. Consequently, there have been efforts to explore self-supervised learning approaches for training occupancy models using only image data [14, 48]. Specifically, volume rendering supervision (inspired by, e.g., NeRF [32] and classical volume rendering [19]) has demonstrated great potential as a training mechanism for occupancy estimation models. It enables the simultaneous supervision of geometry and semantics using 2D labels, which are easier to acquire than 3D voxel labels [2, 14, 48]. Despite these advancements, existing methods are often constrained by a set of predefined classes or rely on pretrained models to generate ground truth, lacking a true generic scene representation.

2.3. Open Vocabulary Perception

The goal of open vocabulary perception (or similarly *zero-shot semantic segmentation*) is to detect or segment object classes that were not explicitly seen during training, given a natural language query. With the help of multi-modal models like CLIP [37], many approaches have been developed in this regard. A common method is to extend CLIP to produce pixel-level features instead of a single image wide feature. MaskCLIP [53] modifies the last pooling layer of CLIP, while LERF [20] and CLIP-FO3D [49] extract patch-wise CLIP embeddings for an image in a sliding-window fashion. Further, methods like [9, 22, 28] train networks on pixel-level segmentation datasets and distill CLIP features simultaneously. OVR [47] trains a generalizable 2D object detector with language pretraining, while ViLD [10] distills CLIP knowledge into a 2-stage detector. OWL-ViT [33] directly attaches a detector to the CLIP image encoder. To enable 3D open vocabulary perception, distillation of vision-language features into NeRFs [20] or Gaussian Splatting [36, 54] have been explored, however these are only trained on a per-scene basis. CLIP-FO3D [49] directly distills extracted vision-language features into a given 3D point cloud via projection. Recently, there have also been efforts for open vocabulary occupancy estimation similar to our work. Most notably, POP-3D [43] trains a model to predict 3D occupancy and 3D vision-language features given just images, but requires LiDAR scans during training. Similarly, OVO [39] aligns voxel predictions with precomputed feature maps, but lacks geometry supervision and is only designed for small and simple scenes. Finally, OpenOcc [17] also represent the scene with voxels, but perform scene reconstruction on a per-scene basis like LERF [20].

3. Methodology

3.1. Problem Definition

Given a set of RGB images $I = \{I^1, I^2, ..., I^N\}$, the objective is to estimate the surrounding environment as a 3D voxel representation V on a defined grid. Each voxel in the

representation is assigned an occupancy probability $V_\sigma \in [0,1]^{X\times Y\times Z}$. Additionally, a vision-language aligned feature vector is estimated for each voxel $V_\psi \in \mathbb{R}^{X\times Y\times Z\times L}$ to model the semantics of the scene in a generic manner. These voxel features can be utilized in various downstream tasks, such as *zero-shot semantic occupancy estimation* or *open vocabulary retrieval*.

3.2. Model Architecture

The proposed model is outlined in Fig. 1. Initially, the input images I are transformed into 3D voxel features V_f using the prominent 2D-to-3D transformation network BEVStereo [24], similar to previous works. However, note that any other 2D-to-3D encoder, like [13, 15, 26], could be used instead. Afterwards, these voxel features are used to predict the density V_σ and the language aligned features V_ψ using a 3D CNN decoder and two separate MLP heads. The entire model is supervised using volume rendering supervision by rendering the estimated 3D features back to the 2D image space and comparing them with precomputed vision-language features (Sec. 3.3). Additionally, in Sec. 3.4, an optional method to enhance detection performance and training efficiency by pretraining a dimensionality reduction encoder on a given vocabulary is presented.

2D-to-3D Encoder Image features are first extracted from the input images I using a pretrained backbone architecture. Next, the features of the current frame and a specified amount of previous frames for temporal propagation, are projected into 3D space using known camera parameters and depth estimations. The 3D features are then pooled to a common 3D voxel grid of features $V_f \in \mathbb{R}^{X\times Y\times Z\times C}$, where X, Y, Z represent the resolution of the grid and C denotes latent dimension size. This architecture is based on BEVStereo [24], except that the features are pooled into a 3D voxel grid instead of a 2D Birds-Eye-View grid.

3D Head The voxel features V_f are processed by a 3D CNN decoder Φ_f, which computes local interactions to refine the features. Subsequently, for each voxel two separate MLP heads Φ_σ and Φ_ψ calculate the density probability σ and a vision-language feature $\psi \in \mathbb{R}^L$, where L represents the feature dimension size. The outputs of Φ_σ are transformed to probabilities using the sigmoid function denoted as $s(\cdot)$ Essentially, the scene geometry is represented by the density probabilities V_σ, which can also be interpreted as occupancy probabilities, while the semantics of the scene is represented by the vision-language features V_ϕ, which is by design not tied to any specific set of classes.

$$V_\sigma = s(\Phi_\sigma(\Phi_f(V_f))) \in [0,1]^{X\times Y\times Z} \quad (1)$$

$$V_\psi = \Phi_\psi(\Phi_f(V_f)) \in \mathbb{R}^{X\times Y\times Z\times L} \quad (2)$$

As will be explained in Sec. 3.3, this separation is required to enable training via volume rendering, which automatically supervises geometry without any explicit loss.

3.3. Volume Rendering Supervision

To supervise the entire model, we use differentiable volume rendering, a technique that gained popularity with the introduction of NeRF [32]. Similar to recent works [2, 14, 34, 48], instead of overfitting a network on a single scene, we use volume rendering as a differentiable operation to bring our predictions from the 3D voxel space back to the 2D image space, where ground truth labels are much easier to acquire.

After estimating the volumes V_σ and V_ψ, for each camera i in the current frame, a set of 3D rays D_r^i is generated, each originating from the camera origin o_i in the direction $d_i(u,v)$ of a pixel (u,v) of the image into the 3D voxel grid, using the camera extrinsic and intrinsic parameters. For each ray r, we then sample a number of 3D points $r(t) = o + td$ at different distances t along the ray and collect the density probabilities $\sigma(r(t))$ and language features $\psi(r(t))$ at these points from the predicted volumes using trilinear interpolation. We then accumulate the language features along each ray to render them to a single feature using the traditional differentiable rendering formulation [19], as in NeRF [48]. Specifically, a rendering weight $w(r(t))$ is computed for each sampled point on the ray by accumulating the interpolated density:

$$w(r(t)) = T\left(r(t)\right)\left(1 - \exp(-\sigma(r(t))\delta_t)\right), \text{ with} \quad (3)$$

$$T(r(t)) = \exp\left(-\sum_{j=1}^{t-1}\sigma(r(j))\delta_j\right), \quad (4)$$

where $T(r(t))$ represents the cumulative transmittance along the ray up to t and δ_t is the distance between the current and next sample. This weight determines the contribution of each point to the final value based on their estimated density. Given this weight, the final rendered 2D vision-language features can be computed by summing up the point features multiplied by their rendering weight.

$$\hat{\Psi}(r) = \sum_{t=1}^{N} w(r(t))\psi(r(t)) \quad (5)$$

Loss Function After rendering the 3D features V_ψ into 2D features $\hat{\Psi}$, a loss can be computed between the estimated features and some 2D ground truth features that we extract from the same input image using a vision-language aligned image encoder, such as CLIP [37]. In this work, we adopt the method proposed in [43] and extract pixel-level CLIP features using MaskCLIP [53]. For each ray, we fetch

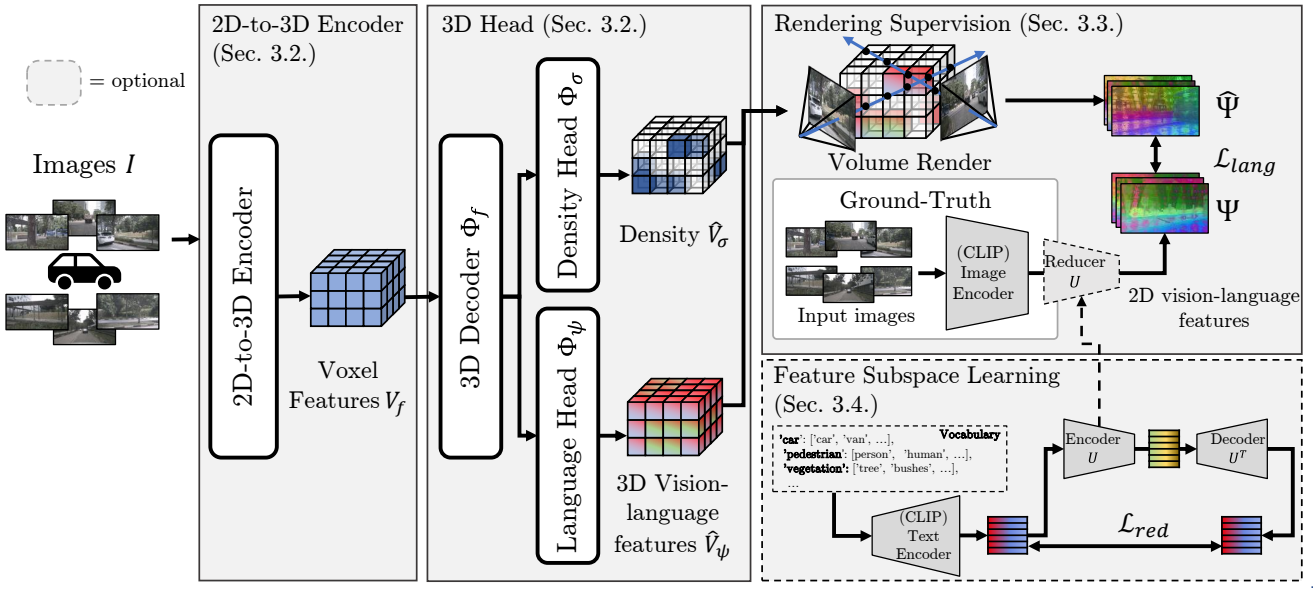

Figure 1. **Architecture of the proposed model.** A set of images is first transformed to 3D voxel features via BEVStereo [24] and a 3D CNN decoder. Next, two separate heads estimate the density probabilities and the generic scene semantics as vision-language features. The model is trained via differentiable volume rendering, using a loss between rendered estimated features and precomputed 2D features from MaskCLIP [53]. Optionally, to increase training efficiency and performance at the cost of expressiveness, *feature subspace learning* can be applied using a predefined vocabulary.

the target feature $\Psi(r)$ via bi-linear interpolation of the pre-computed MaskCLIP feature map I^i_ψ at the pixel coordinate (u, v). As a loss function, we propose the *Cosine Similarity Guided MSE*, which is a combination of the cosine similarity loss and the mean-squared error loss function. We have found that the MSE loss function has a much easier-to-optimize loss landscape, while the cosine similarity gives a better notion of how close the embeddings are in the CLIP space. Therefore, we optimize the MSE loss weighted by the cosine distance $\mathcal{C}$ for each ray, so that features already estimated well have less influence, while features with low cosine similarity to the target have a higher influence on the final loss:

$$\mathcal{L}_{lang}(\hat{\Psi}, \Psi) = \mathcal{C}(\hat{\Psi}, \Psi) * ||\Psi - \hat{\Psi}||^2 \tag{6}$$

$$\mathcal{C}(\hat{\Psi}, \Psi) = (1 - \frac{\hat{\Psi} * \Psi}{||\hat{\Psi}||_2 * ||\Psi||_2}) \tag{7}$$

Note that in our implementation of this loss, we do not back-propagate the cosine distance loss. **Essentially, we distill the knowledge of a strong pretrained 2D vision-language encoder into a 3D voxel-based model via volume rendering, while maintaining the language alignment functionalities.**

Simultaneously, the model is forced to learn correct scene geometry estimations in order to be able to render the features from 3D into different cameras correctly. Therefore, the scene geometry estimation is learned automatically, without any additional loss. It is important to note that the volume rendering technique is only applied during training. During inference, the model just takes the 2D images as input and outputs the scene geometry and 3D vision-language features.

Temporal Rendering In order to estimate the 3D geometry of the scene correctly, the volume rendering supervision method introduced above requires that the voxels are seen from multiple rays, as the depth of a ray is otherwise ambiguous. However, the field of view overlap between different cameras in a multi-view setup is usually very low (e.g., nuScenes [3]), which aggravates the learning of the correct densities without any explicit geometry supervision. To address this, we adopt the temporal rendering approach of recent works [2, 14, 34] and additionally render 2D feature maps for a set of temporally adjacent input images $I_t = \{I_{-t}, ..., I_{-1}, I_{+1}, ..., I_{+t}\}$ during training. For each frame during training, we also generate rays for all temporal frames in a predefined time horizon, and compute the same loss as described above. As we show in the experiments, this temporal rendering approach is crucial for the model to simultaneously learn geometry and semantics from just the feature distillation loss. However, rendering temporally adjacent frames introduces errors due to dynamic objects. We always render our predictions for the current time step, but compute a loss to ground truth feature maps from adjacent time steps, where objects might have moved. As previous work has shown, compensating these errors can lead to bet-

ter performance [2], but requires either ground truth flow data or an additional training task. As this affects only a fraction of voxels, we accept the false supervisory signals from temporal inconsistencies in this work and leave this problem for future work.

3.4. Feature Subspace Learning

While vision-language features offer strong representational power for scene semantics, training a model with the high-dimensional embedding space of vision-language encoders like CLIP imposes a significant computational and memory overhead. Also, not every task necessitates the full expressiveness of the vision-language encoder. In some cases, the requirement may be to detect a specific set of object categories (e.g. the zero-shot occupancy estimation task). Therefore, we propose a method adopted from [21] and train an autoencoder to reduce the embedding space of CLIP to a smaller, task-specific subspace. This offers a trade-off between open vocabulary expressiveness of the full embedding space and a more efficient and specialised lower dimensional space. A lower dimensional subspace specifically modeled for the task at hand can also increase segmentation performance as well as training speed.

Prior to training of our proposed model, we train a single linear transformation $U \in \mathbb{R}^{L \times L'}$ that maps from the original feature space L to the lower dimensional space L' as the encoder, and use the transposed transformation U^T as the decoder. Thus, the encoder and decoder share weights, which forces the matrix U to become orthogonal and reduces the overall amount of parameters to prevent overfitting. This autoencoder is trained solely on the vision-language features $t_i \in \mathbb{R}^L$ for $i \in \{1, ..., n\}$ of a set of n text prompts from a predefined vocabulary, computed via the corresponding text encoder of the vision-language encoder. The same loss as in [21] is used to train U.

$$t_i' = \frac{t_i U}{||t_i U||} \qquad \hat{t}_i = \frac{t_i' U^T}{||t_i' U^T||} \tag{8}$$

$$\mathcal{L}_{red} = \frac{1}{n} \sum_{i=1}^{n} \arccos(t_i, \hat{t}_i). \tag{9}$$

The dataset consists of just a few text prompts, enabling the training of U within seconds. By defining a vocabulary before training, we ensure that the lower-dimensional subspace $\bar{L}$ can focus on the required information and does not model unnecessary features. We can freely define the classes to be detected before the training. Furthermore, we are not bound to either ground truth classes [3, 41, 42, 45] or pretrained object detectors [14, 48]. After the autoencoder is trained, we can use the encoder U to reduce the dimensionality of the ground truth vision-language features L of the images, as the text and vision features are inherently aligned. We then reduce the dimensionality of our language head accordingly and train the model as before. This method thus offers a much more efficient training when detecting certain classes is required, by simply defining a vocabulary of categories of interest, without any overhead. We also refer to the trained encoder U as the *reducer* model.

3.5. Inference

At inference, the estimated embeddings can be used in a versatile way. In this work, we solve the tasks of *3D open vocabulary retrieval* and *zero-shot semantic occupancy estimation*. Results are provided in Sec. 4.

3D Open Vocabulary Retrieval We compute the language feature of a given text query using the text encoder, and then compute the similarity of this query feature with each voxel embedding via the dot product. The resulting similarities can be visualized (e.g., by using a heatmap), or used for binary classification using a threshold.

Zero-shot Semantic Occupancy Estimation Similarly, we can assign each voxel a category by defining a vocabulary that consists of text prompts describing the objects to be detected. For each category, we define multiple prompts that describe this class. Afterwards, for each query prompt, a feature is computed with the text encoder. Given the outputs V_σ and V_ψ of our model, we compute the similarity between each voxel feature with each query feature, and assign every voxel a class based on the query with the highest similarity to the voxels embedding. We also always define a *free* class that models unoccupied voxels, and set a voxel to *free* when the estimated density is below a threshold τ.

4. Experiments

4.1. Dataset and Task Description

We conduct all experiments on the nuScenes dataset [3]. For the *3D open vocabulary retrieval*, we use the benchmark provided by [43]. It consists of 105 samples, each with an open vocabulary text query and corresponding binary labels for the LiDAR point cloud, with the goal of retrieving all 3D points that are described by the query. The performance is measured by the mean-average-precision (mAP) for all points in the scene, and only for points visible in at least one camera (referred to as mAP (v)). For *zero-shot occupancy estimation*, we evaluate on the widely known Occ3D-nuScenes benchmark [41], which provides semantic voxel labels for the nuScenes dataset. We use a predefined vocabulary (see supplementary material) based on the classes given in the benchmark to assign a label to each voxel. The performance is measured in geometric IoU and in mean-IoU over all categories in the benchmark.

4.2. Implementation Details

For all tasks, we use the ResNet50 backbone [11] and an image resolution of 256×704. The density and language heads Φ_σ and Φ_ψ each consist of three hidden layers with a dimension of size 256. We train each network with a batch size of 4 for 18 epochs. We use a time horizon of 12 (to the future and past) for temporal rendering, and generate $32,786$ rays per sample, randomly distributed over all temporal frames in the horizon. Note that temporal rendering and thus temporal frames are only used during training. For each ray, we sample 100 points, and use the *nerfacc* [23] package for rendering. Results are provided for our model using the *Full* embedding space and the *Reduced* space when applying the feature subspace learning strategy. We use the same vocabulary, based on the Occ3D-nuScenes classes, to train the reducer U for each experiment. The reduced dimension size is set to $L' = 128$.

4.3. 3D Open Vocabulary Retrieval

We compare our results to POP-3D [43] on the benchmark provided by the authors. Their model is based on TPV-Former [15] and replaces the semantic head with a vision-language head similar to our model. The authors train their model using LiDAR scans available in nuScenes, both for learning geometry and for feature distillation. They also provide results for directly using MaskCLIP as a baseline, by projecting the LiDAR sweeps on the MaskCLIP feature maps. Results of this comparison are provided in Tab. 1. As is visible, our method outperforms both baselines, even though we use just vision-based supervision. We achieve a mAP score of 21.7 and 22.7 (for all points and only visible points, respectively) compared to the 17.5 and 18.4 of POP-3D, without using LiDAR data. These results clearly demonstrate the effectiveness of rendering supervision in distilling vision-language features into 3D. We account this performance gain mostly to the temporal rendering approach that allows our model to learn from many overlapping views to enhance both geometry and vision-language understanding. As expected, using the proposed feature subspace learning method decreases the open vocabulary performance of LangOcc, as we define a lower dimensional space on a specific set of classes, which decreases detection performance for open vocabulary queries that were not part of that set. As will be shown later, the *Reduced* version instead increases performance on the semantic occupancy estimation task. We show some qualitative results in Fig. 2 highlighting the open vocabulary capabilities of LangOcc. Given just images as input, the model estimates the 3D geometry and generic semantics around the vehicle, allowing to segment any object of interest given a text prompt. The model keeps all the vision-language capabilities of CLIP even in 3D space, and is also capable of segmenting small and thin objects like "metal poles" accurately. Additional qualitative results comparing CLIP features and estimated vision-language features are available in the supplementary material.

Table 1. **3D open vocabulary retrieval results on the benchmark provided by [43].** *mAP (v)* is calculated only on points visible to one of the cameras. The *Mode* indicates the modality used to train the model. *L* and *C* refer to LiDAR scans and camera images, respectively.

Method	Mode	mAP	mAP (v)
MaskCLIP [53]	L	-	14.9
POP-3D [43]	L	17.5	18.4
LangOcc (Full)	C	**21.7**	**22.7**
LangOcc (Reduced)	C	16.6	18.2

4.4. Zero-shot Semantic Occupancy Estimation

We evaluate our approach against other recent vision-only approaches on the Occ3D-nuScenes dataset [41] and show the results in Tab. 2. Both competitors use off-the-shelf semantic segmentation models to generate ground truth labels, but are bound to specific classes. Our proposed method surpasses SelfOcc [14] and OccNeRF [48] on both geometric IoU and semantic mIoU. LangOcc achieves a geometric IoU score of at least 51.59, showing that our model is able to estimate the scene geometry well without any photometric losses or explicit depth supervision. Both SelfOcc and OccNeRF explicitly supervise geometry, for example via multi-view stereo losses and RGB rendering. Seemingly, the density learned via volume rendering of vision-language features gives sufficient signal and is even better suited than using photometric losses to learn geometry. We hypothesize this is likely due to the high representational power of CLIP embeddings and because our model is forced to learn consistent features in 3D over many overlapping views. As a consequence the model gets better geometry supervision compared to usually very ambiguous photometric losses. We further provide a comparison between RGB and feature distillation losses for geometry in Sec. 4.5. Furthermore, both SelfOcc and OccNeRF use class-specific segmentation networks to estimate the voxel labels, while LangOcc can theoretically detect any class with the same model (in the *full* variation). Even though our model is trained without any explicit class definition, we outperform both competitors also in terms of semantic mIoU, highlighting the power of the estimated features. By specifying a vocabulary for the given task and using the proposed dimensionality reduction method (Sec. 3.4), we can further increase the semantic mIoU score from 10.71 to 11.84. To conclude, using just a single loss function and a straightforward training paradigm, our method achieves state-of-the-art performance on vision-only Occ3D-nuScenes, while still being capable of open-vocabulary detection. As mentioned above, the reducer U finishes training within a sec-

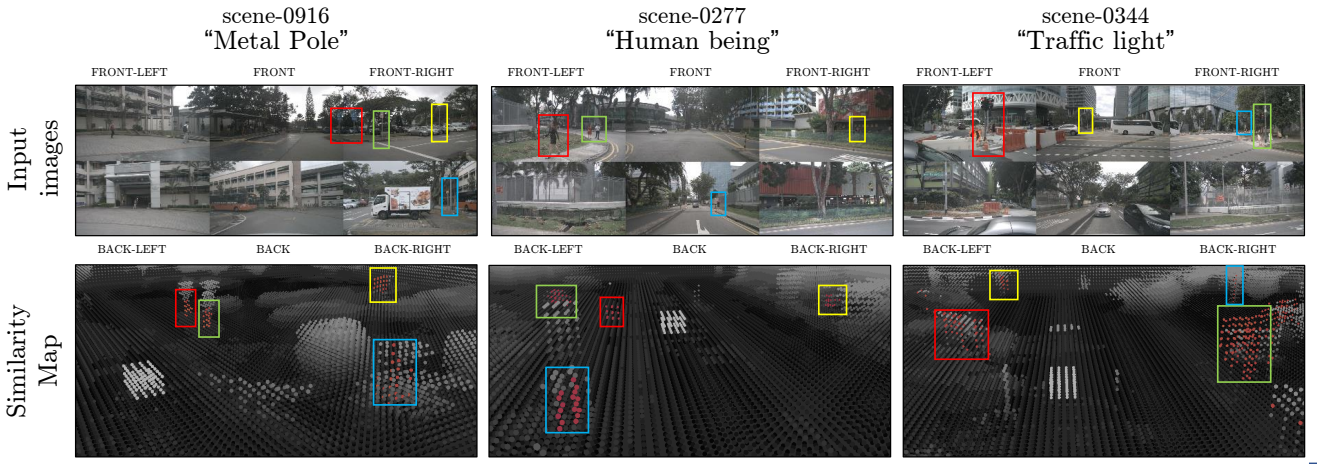

Figure 2. **Qualitative results showing open vocabulary retrieval on nuScenes [3].** Given a text query, we compute similarities between the text embedding and each estimated voxel embedding and highlight voxels with a high similarity score. Ego vehicle shown in white.

Table 2. **Semantic occupancy estimation results on the Occ3D-nuScenes benchmark [41] in terms of geometric IoU and semantic mIoU.** We compare with recent camera-only methods. Best and second best performing in **bold** and *italics*, respectively.

Method	IoU	mIoU
SelfOcc [14]	45.01	9.30
OccNeRF [48]	-	10.13
LangOcc (Full)	*51.59*	*10.71*
LangOcc (Reduced)	**51.76**	**11.84**

Table 3. **Ablation on the loss function used for $\mathcal{L}_{lang}$.**

Loss Function	MSE	CosSim	Cos-guided MSE
IoU	50.29	49.88	**51.59**
mIoU	9.41	9.89	**10.71**
mAP (v)	20.1	22.6	**22.7**

ond and thus does not impose any notable overhead. We also present results for methods trained with LiDAR or 3D voxel labels [13, 15, 41] to demonstrate the gap between LiDAR-supervised and vision-only approaches in Sec. 4.5. We provide the performance on individual classes and additional qualitative results in the supplementary material.

4.5. Ablations

Loss function We provide a comparison between using our proposed *Cosine Similarity Guided MSE* function and using either the MSE loss or the Cosine Similarity loss by training a model with each loss function. As the results in Tab. 3 show, our loss function leads to increased performance on each metric.

Temporal Horizon We ablate the temporal horizon of the model during training and show the results in Tab. 4. As expected, using no temporal rendering at all leads to very poor results, as the model can hardly learn any 3D geometry from the very few overlapping rays. Adding 4 future and past frames during rendering supervision already improves all scores significantly, such that LangOcc achieves a better open vocabulary retrieval performance than POP-3D [43]. The best performance on all tasks was achieved by using a horizon of 12, which seems to be a good trade-off between overlap of cameras and view diversity. Adding more temporal frames led to a decrease in performance. We hypothesize this is due to the large distance between the camera poses, so that many sampled points for rendering are not visible in the current frame and too many temporal inconsistencies become present.

Table 4. **Ablation on the temporal horizon.**

Horizon	0	4	8	12	16	20
IoU	15.78	40.85	49.71	**51.59**	50.54	49.74
mIoU	2.88	8.46	9.95	**10.71**	9.46	9.16
mAP (v)	9.8	20.0	22.6	**22.7**	21.8	20.2

Reduced Dimension Size Table 5 shows a comparison between using different subspace dimension sizes for the reducer U. We use the same vocabulary as in Sec. 4.4 for all models to train the autoencoder, but modify the target dimension size (with 512 being the *full* space). As observable, the open vocabulary performance decreases steadily, the smaller the subspace gets, as the model loses representational power and overfits more to the provided vocabulary. However, using the dimensionality reducer can offer improved performance on the zero-shot occupancy estimation task. The best performance can be achieved at $L' = 128$, which seems to be the optimal trade-off between task-specific expressiveness and not overfitting. Decreasing the dimension size further however, up to 32, still offers increased mIoU performance compared to the original space.

Table 5. **Ablation on the subspace dimensionality L'.**

L'	16	32	64	128	256	512
IoU	50.00	50.18	51.02	**51.76**	51.11	51.59
mIoU	7.52	10.86	11.18	**11.84**	11.08	10.71
mAP (v)	10.6	11.2	17.1	18.2	19.1	**22.7**

Only when the subspace dimensionality is decreased to 16, the performance decreases drastically. Interestingly, the geometric estimations only differ slightly from the full space at higher dimensions, while they decline at lower sizes. This is likely because higher dimensions have more capacity to encode information about geometry, while the lower dimensions have to focus more on the semantic features from the vocabulary. Also, feature vectors are more distinct in high dimensions, such that finding corresponding points in different views is much easier than in lower dimensions, where many feature vectors are similar.

Geometry Supervision To show that training our proposed model with just the feature distillation loss $\mathcal{L}_{lang}$ leads to state-of-the-art 3D geometry estimations, we directly compare our approach with using photometric losses for training, like it is done in [14, 48]. We train our proposed model with RGB rendering by replacing the language-feature head Φ_ψ with an RGB head that estimates the appearance of a voxel in terms of RGB and train with a *MSE* loss on the rendered RGB values. As is common in NeRF approaches, we choose to model the appearance with spherical harmonics [7, 40]. We compare the model on the geometric IoU score on the Occ3D-nuScenes benchmark. Training our model with this RGB supervision leads to a geometric IoU score of 41.10. Our proposed supervision method leads to a significantly better IoU score of 51.59, which confirms that the feature distillation loss provides a better supervision signal for the scene geometry than a photometric loss in our model architecture. We speculate that this originates from the rich information of vision-language features and their independence from the viewing angle that impose clear constraints on the scene geometry. Photometric losses on the other hand suffer from ambiguities like low-texture regions, locally similar pixel colors, different lighting conditions and dependence on the viewing angle which makes it difficult to extract a clear geometric signal.

Comparison to LiDAR-based models We provide a comparison between our camera-only approach and two recent methods using annotated 3D data. OccFlowNet [2] trains the model using annotated LiDAR point clouds, while CTF-Occ [41] trains with 3D voxel labels generated from aggregated point clouds. As is visible in Tab. 6, methods using annotated 3D data still outperform pure camera-only methods. However, the data used for training requires expensive manual labelling and additional sensor data, while our method is trained with just the camera images.

Table 6. **Comparing our model with state-of-the-art occupancy estimation methods using LiDAR or 3D voxel labels on Occ3D-nuScenes [41].** We compare only on the mIoU, as these works do not provide results for the geometry. *Mode* indicates the modality used during training. *3D*, *L*, *C* refer to semantic 3D voxel labels, semantic LiDAR points and camera images, respectively.

Method	Mode	mIoU
OccFlowNet [2]	L	26.14
CTF-Occ [41]	3D	28.53
LangOcc (Full)	C	10.71
LangOcc (Reduced)	C	11.84

5. Conclusion

In this paper, we have proposed a novel model that enables a generic open vocabulary scene representation and a weakly-supervised training mechanism that requires only images as input. By using differentiable volume rendering, we distill the rich knowledge of the vision-language encoder CLIP into a 3D occupancy estimation model and simultaneously learn to estimate scene geometry, without any explicit geometry supervision. This allows for generic 3D scene representations which are completely independent of specific class definitions. Our model requires no per-scene optimization like prior work [20, 54]. It significantly outperforms previous attempts for open vocabulary occupancy without using any LiDAR data. Additionally, we set the new state-of-the-art performance on vision-only semantic occupancy estimation on the Occ3D-nuScenes dataset, and further improve segmentation performance using the proposed feature subspace learning method. We conclude that by distilling knowledge of strong 2D visual encoders into 3D occupancy estimation models, stronger occupancy estimations are possible than with photometric methods like [14, 48]. Incorporating more generic feature representations like DINO (which has been shown to encode better geometric features than CLIP [6]) can be a promising future direction. Also, our work still lacks a mechanism to deal with dynamic objects, which leads to inconsistent supervisory signals during temporal rendering. In future work, the explicit modeling of scene dynamics could help to remove inconsistent signals. Moreover, the benchmark of [43] is relatively small and covers only common driving scene objects. To compare future open vocabulary approaches a larger and more diverse benchmark dataset would be beneficial. Finally and building on the strong performance of our model, additional research on open vocabulary occupancy is required to further investigate its applicability and potential performance gains, highly demanded in a variety of tasks such as autonomous driving.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In *Proc. of the IEEE/CVF International Conf. on Computer Vision (ICCV)*, 2019. 2

[2] Simon Boeder, Fabian Gigengack, and Benjamin Risse. Occflownet: Towards self-supervised occupancy estimation via differentiable rendering and occupancy flow. *arXiv preprint arXiv:2402.12792*, 2024. 2, 3, 4, 5, 8

[3] 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 multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 4, 5, 7

[4] Anh-Quan Cao and Raoul de Charette. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022. 2

[5] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *arXiv preprint arXiv:2306.16927*, 2023. 1

[6] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21795–21806, 2024. 8

[7] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinhong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5501–5510, 2022. 8

[8] Wanshui Gan, Ningkai Mo, Hongbin Xu, and Naoto Yokoya. A simple attempt for 3d occupancy estimation in autonomous driving. *arXiv preprint arXiv:2303.10076*, 2023. 1, 2

[9] Golnaz Ghiasi, Xiuye Gu, Yin Cui, and Tsung-Yi Lin. Scaling open-vocabulary image segmentation with image-level labels. In *European Conference on Computer Vision*, pages 540–557. Springer, 2022. 2

[10] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2

[11] 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

[12] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 2

[13] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 3, 7

[14] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. *arXiv preprint arXiv:2311.12754*, 2023. 1, 2, 3, 4, 5, 6, 7, 8

[15] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2023. 1, 2, 3, 6, 7

[16] Haoyi Jiang, Tianheng Cheng, Naiyu Gao, Haoyang Zhang, Wenyu Liu, and Xinggang Wang. Symphonize 3d semantic scene completion with contextual instance queries. *arXiv preprint arXiv:2306.15670*, 2023. 2

[17] Haochen Jiang, Yueming Xu, Yihan Zeng, Hang Xu, Wei Zhang, Jianfeng Feng, and Li Zhang. Openocc: Open vocabulary 3d scene reconstruction via occupancy representation. *arXiv preprint arXiv:2403.11796*, 2024. 1, 2

[18] Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. Polarformer: Multi-camera 3d object detection with polar transformer. In *Proceedings of the AAAI conference on Artificial Intelligence*, pages 1042–1050, 2023. 2

[19] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 2, 3

[20] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 2, 8

[21] Konstantin Kobs, Michael Steininger, and Andreas Hotho. Indirect: language-guided zero-shot deep metric learning for images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1063–1072, 2023. 5

[22] Boyi Li, Kilian Q Weinberger, Serge Belongie, Vladlen Koltun, and René Ranftl. Language-driven semantic segmentation. *arXiv preprint arXiv:2201.03546*, 2022. 2

[23] Ruilong Li, Hang Gao, Matthew Tancik, and Angjoo Kanazawa. Nerfacc: Efficient sampling accelerates nerfs. *arXiv preprint arXiv:2305.04966*, 2023. 6

[24] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1486–1494, 2023. 3, 4

[25] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9098, 2023. 1, 2

[26] 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. In *European conference on computer vision*, pages 1–18. Springer, 2022. 2, 3

[27] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023. 2

[28] Feng Liang, Bichen Wu, Xiaoliang Dai, Kunpeng Li, Yinan Zhao, Hang Zhang, Peizhao Zhang, Peter Vajda, and Diana Marculescu. Open-vocabulary semantic segmentation with mask-adapted clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7061–7070, 2023. 2

[29] Haisong Liu, Yang Chen, Haiguang Wang, Zetong Yang, Tianyu Li, Jia Zeng, Li Chen, Hongyang Li, and Limin Wang. Fully sparse 3d occupancy prediction, 2024. 2

[30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. In *European Conference on Computer Vision*, pages 531–548. Springer, 2022. 2

[31] Yuhang Lu, Xinge Zhu, Tai Wang, and Yuexin Ma. Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries. *arXiv preprint arXiv:2312.03774*, 2023. 2

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[33] Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, et al. Simple open-vocabulary object detection. In *European Conference on Computer Vision*, pages 728–755. Springer, 2022. 2

[34] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*, 2023. 1, 2, 3, 4

[35] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 2

[36] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20051–20060, 2024. 2

[37] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 3

[38] Xin Tan, Wenbin Wu, Zhiwei Zhang, Chaojie Fan, Yong Peng, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Geocc: Geometrically enhanced 3d occupancy network with implicit-explicit depth fusion and contextual self-supervision. *arXiv preprint arXiv:2405.10591*, 2024. 2

[39] Zhiyu Tan, Zichao Dong, Cheng Zhang, Weikun Zhang, Hang Ji, and Hao Li. Ovo: Open-vocabulary occupancy. *arXiv preprint arXiv:2305.16133*, 2023. 2

[40] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. 8

[41] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023. 1, 2, 5, 6, 7, 8

[42] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023. 2, 5

[43] Antonin Vobecky, Oriane Siméoni, David Hurych, Spyridon Gidaris, Andrei Bursuc, Patrick Pérez, and Josef Sivic. Pop-3d: Open-vocabulary 3d occupancy prediction from images. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 3, 5, 6, 7, 8

[44] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3631, 2023. 2

[45] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. *arXiv preprint arXiv:2303.03991*, 2023. 2, 5

[46] 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 preprint arXiv:2311.12058*, 2023. 2

[47] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14393–14402, 2021. 2

[48] Chubin Zhang, Juncheng Yan, Yi Wei, Jiaxin Li, Li Liu, Yansong Tang, Yueqi Duan, and Jiwen Lu. Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. *arXiv preprint arXiv:2312.09243*, 2023. 1, 2, 3, 5, 6, 7, 8

[49] Junbo Zhang, Runpei Dong, and Kaisheng Ma. Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2048–2059, 2023. 2

[50] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occu-

pancy prediction. *arXiv preprint arXiv:2304.05316*, 2023. 1, 2

[51] Yanan Zhang, Jinqing Zhang, Zengran Wang, Junhao Xu, and Di Huang. Vision-based 3d occupancy prediction in autonomous driving: a review and outlook. *arXiv preprint arXiv:2405.02595*, 2024. 1

[52] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9806–9815, 2024. 2

[53] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision*, pages 696–712. Springer, 2022. 2, 3, 4, 6

[54] Shijie Zhou, Haoran Chang, Sicheng Jiang, Zhiwen Fan, Zehao Zhu, Dejia Xu, Pradyumna Chari, Suya You, Zhangyang Wang, and Achuta Kadambi. Feature 3dgs: Supercharging 3d gaussian splatting to enable distilled feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21676–21685, 2024. 2, 8