VEON: Vocabulary-Enhanced Occupancy Prediction

Jilai Zheng[1], Pin Tang[1], Zhongdao Wang[2], Guoqing Wang[1], Xiangxuan Ren[1], Bailan Feng[2], and Chao Ma[1⋆]

[1] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2] Huawei Noah's Ark Lab
{zhengjilai,pin.tang,guoqing.wang,bunny_renxiangxuan,chaoma}@sjtu.edu.cn
{wangzhongdao,fengbailan}@huawei.com

Abstract. Perceiving the world as 3D occupancy supports embodied agents to avoid collision with any types of obstacle. While open-vocabulary image understanding has prospered recently, how to bind the predicted 3D occupancy grids with open-world semantics still remains under-explored due to limited open-world annotations. Hence, instead of building our model from scratch, we try to blend 2D foundation models, specifically a depth model MiDaS and a semantic model CLIP, to lift the semantics to 3D space, thus fulfilling 3D occupancy. However, building upon these foundation models is not trivial. First, the MiDaS faces the depth ambiguity problem, i.e., it only produces relative depth but fails to estimate bin depth for feature lifting. Second, the CLIP image features lack high-resolution pixel-level information, which limits the 3D occupancy accuracy. Third, open vocabulary is often trapped by the long-tail problem. To address these issues, we propose VEON for **V**ocabulary-**E**nhanced **O**ccupancy predictio**N** by not only assembling but also adapting these foundation models. We first equip MiDaS with a Zoedepth head and low-rank adaptation (LoRA) for relative-metric-bin depth transformation while reserving beneficial depth prior. Then, a lightweight side adaptor network is attached to the CLIP vision encoder to generate high-resolution features for fine-grained 3D occupancy prediction. Moreover, we design a class reweighting strategy to give priority to the tail classes. With only 46M trainable parameters and zero manual semantic labels, VEON achieves 15.14 mIoU on Occ3D-nuScenes, and shows the capability of recognizing objects with open-vocabulary categories, meaning that our VEON is label-efficient, parameter-efficient, and precise enough.

Keywords: Open Vocabulary · 3D Occupancy · 2D Foundation Models

1 Introduction

In recent years, the autonomous driving community has been paying increasing attention to the sophisticated, voxel-level understanding of the 3D space around

⋆ Corresponding author.

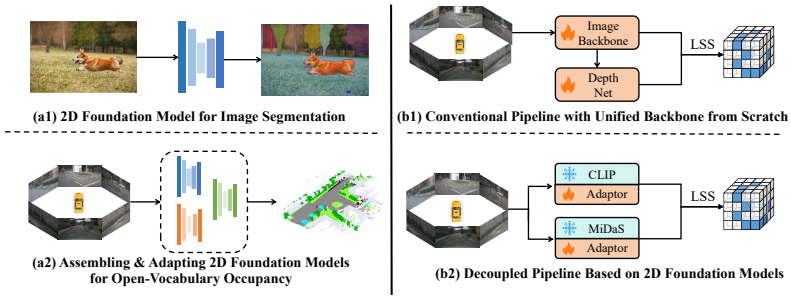

Fig. 1: Main idea of our VEON. **Left:** Referring to the strong data prior in 2D foundation models, we resort to unleashing their power for handling 3D open-vocabulary tasks. **Right:** Compared with the conventional practice of training a unified 2D backbone from scratch, we design a decoupled pipeline that assembles and adapts a depth model MiDaS [38] and a semantic model CLIP [37], for 3D open-vocabulary occupancy.

the ego car. This perception task of the new era, dubbed as *Occupancy Prediction*, aims to assign each voxel in the 3D space with semantic information, namely what (class of) object occupies each specific voxel. In this paper, we mainly focus on vision-centric open-vocabulary occupancy prediction. This practical setting stands out for *(1)* utilizing only surrounding images during inference and *(2)* recognizing objects of a variety of categories that could exist on the roads. Such geometrical and fine-grained information has been proven beneficial to not only the scene understanding but also the subsequent planning and control [18, 26].

While there has been a remarkable improvement in open-vocabulary image understanding benefiting from 2D foundation models [37] as shown in Fig. 1(a1), their 3D counterparts on occupancy prediction still lag far behind. This is mainly attributed to the lack of large-scale open-world 3D occupancy annotations, which is caused by the labor-intensive labeling process. In fact, as illustrated in Fig. 1(b1), existing solutions to open-vocabulary 3D occupancy prediction [23, 41, 45, 52] still rely on training an end-to-end deep network with depth estimation and semantic extraction modules from scratch. Considering the absence of abundant labeled open-vocabulary 3D data, the current strategy hinders the performance ceiling of open-vocabulary 3D occupancy predictors.

Inspired by the success of 2D open-vocabulary scene understanding, we alternatively resort to assembling 2D foundation models for open-world 3D occupancy and unleashing their power on 3D occupancy prediction as depicted in Fig. 1(a2). A naive pipeline characterized by module decoupling is given in Fig. 1(b2), where we utilize a depth foundation model MiDaS [38] to lift the semantics produced by the vision-language foundation model CLIP [37] from 2D image pixels to 3D grids, thus fulfilling the 3D occupancy task. However, leveraging these foundation models is not trivial and meets challenges. First, as MiDaS is trained to estimate relative depth consistent across tens of indoor and outdoor datasets, a domain gap exists between the pretrained relative depth and the bin depth required in feature lifting [36]. Thus, we propose to first adapt MiDaS [38] with a

Zoedepth [7] head for relative-to-metric depth transformation, and then convert metric depth to bin depth in a differentiable manner. Besides, we equip the MiDaS backbone with low-rank adaptation (LoRA) [17] to conduct domain transfer while reserving beneficial depth prior. Second, as CLIP [37] is trained through image-level paired consistency, the CLIP image features lack spatial pixel-level information. Also, the ViT [13] architecture causes a low-resolution compromise on the sizes of image features, which is fatal to scene understanding. To resolve this issue, we propose to attach a High-resolution Side Adaptor (HSA) beside the CLIP image encoder. It maintains high-resolution features to compensate for the information loss caused by the low-resolution CLIP encoder, and keeps lightweight by absorbing CLIP features. It can also slightly manipulate the CLIP attention bias, in order to make CLIP better suited to the requirement of spatial feature extraction. Finally, we also design a class reweighting loss to handle tail classes. By putting more emphasis on tail classes, our VEON could better learn to recognize various objects, sticking to the open-vocabulary essence.

Compared with the previous occupancy prediction methods, our VEON takes full advantage of the pretrained 2D foundation models with strong 2D data prior. It has much fewer trainable parameters while obtaining competitive performance. For example, with only 46.0M trainable parameters and no manual semantic annotations, our VEON model (with ViT-L backbone) achieves a competitive performance of 15.14 mIoU on the large-scale dataset Occ3D-nuScenes [9, 43]. It also demonstrates the capability of recognizing objects of unseen classes never explicitly annotated in the training dataset.

Our main contributions can be summarized as follows.

- We design a VEON framework to solve open-vocabulary 3D occupancy prediction by assembling and adapting a depth estimation foundation model (i.e., MiDaS [38]) and a vision-language foundation model (i.e., CLIP [37]).
- We propose to conquer the domain gap of applying MiDaS to occupancy prediction by relative-metric-bin transformation and low-rank adaptation.
- We attach a lightweight side adaptor network beside CLIP for extracting high-resolution and spatial-aware features that better suit scene understanding. And a class reweighting loss is designed to put emphasis on tail classes.
- Experiments show that our VEON can obtain competitive performance with very few trainable parameters and partial or even zero manual annotations.

2 Related Work

Vision-centric 3D occupancy prediction. Occupancy prediction aims at assigning semantic labels to all voxels around the ego car [42,43,46]. MonoScene [10] is the first work on predicting voxel-wise occupancy given monocular RGB camera inputs. OccDepth [33] exploits the stereo images and distills knowledge from them. TPVFormer [24] seeks a tri-perspective view representation to understand the scene. VoxFormer [27] designs a lightweight framework for occupancy prediction by explicitly specifying visible voxel queries. While early works typically experiment on the SemanticKitti dataset [4], recently, several occupancy

benchmarks have been built on larger-scale datasets. For instance, Occ3D [43] explores a three-stage label generation pipeline for dense semantic occupancy labels. Annotations are generated on nuScenes [9] and Waymo [40], and a novel CTF-Occ method is testified. Similarly, SurroundOcc [48], OpenOccupancy [47] and OccNet [44] also constructs their occupancy benchmarks on nuScenes [9].

Open-vocabulary 3D scene understanding. Foundation 2D vision-language models establish a strong connection between natural language and images. However, this connection is lacking in 3D scene understanding. One natural solution is to connect 3D data and language by utilizing 2D as a bridge. 3D-OVS [30] distills knowledge from CLIP [37] and DINO [11] into a neural radiance field (NeRF [34]), obtaining the capability of 3D open-vocabulary segmentation. PLA [12] leverages the geometric consistency between posed images and 3D scenes to learn language-driven 3D representation. OpenScene [35] predicts dense 3D scene representation via aligning the point features with CLIP. OVIR-3D [32] explores open-vocabulary 3D instance retrieval by first generating 2D text-aligned region proposals and then fusing them in 3D. OpenIns3D [25] proposes "Mask-Snap-Lookup" for open-vocabulary 3D instance segmentation.

Open-vocabulary 3D occupancy prediction. Predicting open-vocabulary occupancy remains an under-explored problem. Early works [41] mainly focus on small-scale scenes. Recently, POP-3D [45] introduced this task into the nuScenes dataset [9] for autonomous driving. POP-3D is trained from scratch with the conventional 2D-3D encoder architecture, and leverages language, point cloud, and images for training. Some self-supervised occupancy predictors, *e.g.* SelfOcc [23] and OccNeRF [52], can also be revised for open-vocabulary recognition by aligning with pseudo open-vocabulary labels in 2D.

3 Method

3.1 Problem Setup

3D occupancy prediction focuses on predicting the voxel-wise semantic state in 3D space. We divide the space around the ego car into $H \times W \times Z$ voxels and predict which class of object occupies each voxel, denoted as $\mathbf{O}$. Here H, W, Z are respectively the length, width, and height of the equally sliced voxel grid. During inference, the model will only input N_{cam} images $\mathbf{I} = \{I_i \mid i \in [1, N_{cam}]\}$ from surrounding cameras, implying a vision-centric occupancy prediction task. While in training, the corresponding point cloud $\mathbf{P}$ is also available. Notice that $\mathbf{P}$ is collected via LiDAR, without manual efforts.

Our model is designed to recognize objects in an open-vocabulary setting. Formally, the overall class set C_{all} can be divided into a seen class set C_s and an unseen class set C_u, where $C_s \cup C_u = C_{all}$ and $C_s \cap C_u = \varnothing$. During training, our model needs to fit its prediction $\mathbf{O}$ to the ground truth $\hat{\mathbf{O}}$, where all the labeled classes in $\hat{\mathbf{O}}$ are inside C_s. During inference, the model is required to provide open-set occupancy results (inside C_{all}). Our work focuses on two settings, i.e., $C_s \neq \varnothing$ and $C_s = \varnothing$. The former utilizes partial semantic labels, while the latter has no access to any semantic annotations.

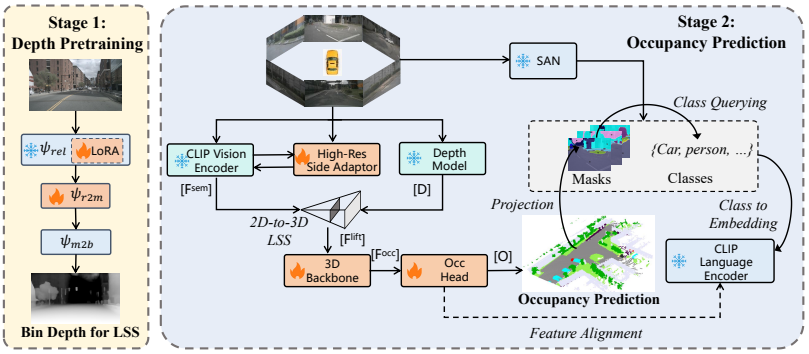

Fig. 2: Framework overview. Our VEON consists of two training stages: depth pretraining and occupancy prediction. **Left:** In stage 1, we adapt the MiDaS [38] backbone with a relative-metric-bin depth transformation adaptor to estimate the bin depth for LSS feature lifting [36]. Low-rank adaptation (LoRA) [17] is integrated for enhanced domain transfer. **Right:** In stage 2, we unleash the power of CLIP [37] via equipping a High-resolution Side Adaptor (HSA). The refined high-resolution CLIP semantic feature is lifted via LSS and goes through 3D convolutions for 3D occupancy. The network reserves the capability of recognizing open-vocabulary objects by aligning the 3D representation with CLIP language embeddings of certain classes, which is determined by the off-the-shelf 2D open-vocabulary segmentor SAN [50].

3.2 Framework Overview

Fig. 2 illustrates the basic framework of our VEON. The design rationale of VEON is to assemble and adapt two 2D foundation models, the large depth model MiDaS [8, 38] and the vision-language semantic-aware model CLIP [37], through a decoupled network structure. These two foundation models are trained with a vast number of 2D data, providing strong data prior for our VEON. As in Fig. 2, we divide the training procedure of VEON into two stages as below:

- **Stage 1: Depth Pretraining.** In stage 1, we adapt and tune a depth estimation model ϕ_{dp} from the foundation depth model MiDaS [8, 38]. ϕ_{dp} takes surrounding camera images $\mathbf{I}$ as input, and estimate bin depth $\mathbf{D}'$ for them, ready for future LSS [36].
- **Stage 2: Occupancy Prediction.** In stage 2, we equip the CLIP [37] vision encoder with a High-resolution Side Adaptor (HSA), in order to extract an enhanced semantic-aware 2D feature $\mathbf{F}^{\mathbf{sem}}$. Then, we lift $\mathbf{F}^{\mathbf{sem}}$ as $\mathbf{F}^{\mathbf{lift}}$ via LSS [36] based on the bin depth $\mathbf{D}'$ estimated through ϕ_{dp}. Finally, we process $\mathbf{F}^{\mathbf{lift}}$ via 3D convolutions, outputting the occupancy $\mathbf{O}$. During training, $\mathbf{O}$ will be projected and aligned with a 2D open-vocabulary segmentor.

We note that leveraging these two foundation models is not easy due to several challenges, including domain gap, low resolution, tail classes, etc. Thus, as in Fig. 2, we carefully design lightweight adaptors to unleash the power of these foundation models. We will go into particulars in the following sections.

3.3 Stage 1: Depth Pretraining

MiDaS is a monocular depth estimation model trained with tens of labeled depth datasets [22, 28, 39, 49, 51]. To combine various depth datasets with distinct biases as a whole, MiDaS [38] establishes a solution by estimating relative depth irrelevant to depth range and scale. In this way, the pretrained MiDaS backbone, denoted as ϕ_{rel}, could estimate precise relative depth across biased datasets.

Despite the strong data prior provided by MiDaS, there exists a gap between the pretrained MiDaS and our requirements. In fact, MiDaS [38] is trained for relative depth, but LSS [36] in 3D occupancy requires normalized bin depth for 2D-to-3D view transformation. Besides, the depth domain in autonomous driving is slightly different from that of the pretraining datasets. This motivates us to design the relative-metric-bin adaptor for end-to-end differentiable depth transformation. We also adopt the low-rank adaptation (LoRA) [17] to tune the MiDaS backbone for enhanced domain transfer.

Pipeline. We propose to attach a *relative-metric-bin adaptor* to the MiDaS backbone to transform the relative depth into bin depth for LSS and bridge the domain gap. As shown in Fig. 2 (left), we can formulate the depth estimation module as $\phi_{dp} = \phi_{rel} \circ \phi_{r2m} \circ \phi_{m2b}$. And the bin depth can be estimated by $\mathbf{D}' = \phi_{dp}(\mathbf{I})$. Here, $\circ$ means the cascade operation of networks. ϕ_{rel}, ϕ_{r2m} and ϕ_{m2b} are respectively the relative depth backbone, the relative-to-metric adapting network, and the metric-to-bin transformation, as presented below.

(1) Relative depth backbone ϕ_{rel}**.** We directly adopt the pretrained MiDaS to serve as ϕ_{rel}, and freeze it for reserving the data prior. However, as there exists a domain gap between the pretraining data and driving scenes, we apply low-rank adaptation (LoRA) [17] to all linear layers within the MiDaS backbone. Notably, this strategy adds only 0.3% additional parameters to the pretrained MiDaS, but significantly enhances domain transfer and unleashes the power of the depth foundation model as shown in Sec. 4.3.

(2) Relative-to-metric adapting network ϕ_{r2m}**.** Metric depth represents depth with absolute values (*e.g.* 50 meters). For building ϕ_{r2m}, we introduce the ZoeDepth [7] head as a lightweight network adaptor. This module collects features from decoder layers of the MiDaS backbone and leverages an enhanced bin-based strategy [5,6] for calculating the metric depth. We refer readers to the ZoeDepth [7] paper for detailed network architecture.

We optimize ϕ_{r2m} by fitting the metric depth $\mathbf{D}$ output from ϕ_{r2m} towards the ground truth depth $\hat{\mathbf{D}}$. Here $\hat{\mathbf{D}}$ is obtained by projecting the point cloud $\mathbf{P}$ onto the camera plane. Suppose d_i is the i-th pixel of $\mathbf{D}$, and $\hat{d}_i$ is the corresponding ground truth. We strictly follow [5, 7, 14] to formulate a pixel-wise scale-invariant depth loss L_{pix} (see the supplementary material for formulation). L_{pix} ensures the shape and smoothness of the output metric depth, beneficial to the subsequent bin depth transformation.

(3) Metric-to-bin transformation ϕ_{m2b}**.** To transform metric depth $\mathbf{D}$ to bin depth $\mathbf{D}'$ for LSS, we define N_{bin} depth bins with equal width w. Suppose the first depth bin has its center as d_{fc}, then the j^{th} depth bin ($0 \leq j < N_{bin}$) should cover the interval $[d_{fc} + (j - 0.5) \cdot w, d_{fc} + (j + 0.5) \cdot w]$ with bin center

$d_{fc} + j \cdot w$. Then, the metric depth d_i can be transformed into a N_{bin} dimension tensor d'_i (bin depth), with the j^{th} dimension representing the similarity score of d_i to the j^{th} depth bin. We formally define this similarity value d'_{ij} as:

$$d'_{ij} = \text{softmax}_j(\beta \cdot h_{ij}), \quad \text{where} \quad h_{ij} = -\left|d_i - d_{fc} - j \cdot w\right|, \tag{1}$$

and β is a constant. Then, we can define the ground truth one-hot depth bin distribution $\hat{d}'_{ij}$ as follows:

$$\hat{d}'_{ij} = \begin{cases} 1, & \text{if } \ |\hat{d}_i - d_{fc} - j \cdot w| \leq w/2 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

In this way, the bin depth map $\mathbf{D}'$ can be supervised with cross-entropy loss, with the loss defined as L_{bd}. The total loss in stage one L_{stg1} is the weighted sum of the pixel-wise depth loss L_{pix} and the bin depth loss L_{bd}.

3.4 Stage 2: Occupancy Prediction

In stage 2, we resort to CLIP for extracting a 2D semantic-aware feature $\mathbf{F^{sem}}$, and then lift $\mathbf{F^{sem}}$ from 2D to 3D via LSS [36] based on the bin depth map $\mathbf{D}'$ from Eq. 1. A trivial solution here is to directly fetch visual tokens from CLIP as $\mathbf{F^{sem}}$, but this meets challenges and we will discuss our improvement later. The LSS operation gives the initial 3D feature $\mathbf{F^{lift}}$. After that, the lifted feature $\mathbf{F^{lift}}$ will be processed via a series of ResNet3D [16] blocks, generating a dense semantic-aware representation, denoted as $\mathbf{F^{occ}}$.

Then, we decode the occupancy results from $\mathbf{F^{occ}}$ through two separate 3D convolution heads. For each voxel-wise occupancy representation F_i^{occ} from the i^{th} voxel of $\mathbf{F^{occ}}$, we respectively adopt: *(1)* a two-layer 3D convolution head to generate a tag $O_i^{bin} \in [0, 1]$ indicating binary occupancy state, i.e., whether the voxel is occupied by any object or not, and *(2)* a three-layer 3D convolution head to predict a semantic-aware embedding O_i^{sa} fitting the feature distribution of CLIP output. The embedding map $\mathbf{O^{sa}}$ above is responsible for determining the semantic class. During inference, suppose the j^{th} class in any class set C has its embedding from CLIP language encoder as F_j^{lan}, then the final occupancy result O_i for the i^{th} voxel can be formulated as follows:

$$O_i = \begin{cases} \text{argmax}_j \, O_i^{sa} \cdot F_j^{lan}, & \text{if } \ O_i^{bin} \geq \tau \\ 0, & otherwise \end{cases}, \tag{3}$$

where class 0 indicates the special class "free" and $0 < \tau < 1$. Here, we have $C = C_{all}$ for open-vocabulary occupancy prediction. In this way, we can assemble the 2D foundation models to formulate the 3D occupancy pipeline.

However, integrating CLIP meets two challenges. First, the resolution of CLIP features is small (16×44 for ViT-B, and 18×50 for ViT-L), hindering fine-grained scene understanding. We thus maintain an adaptor network beside the CLIP vision encoder, reserving high-resolution information. Second, the CLIP

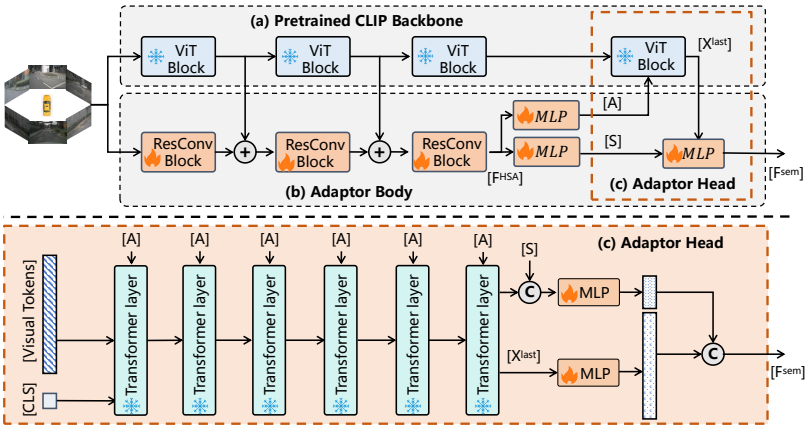

Fig. 3: Detailed network architecture of the High-resolution Side Adaptor (HSA). **Top**: Adaptor architecture. We maintain a series of residual convolution blocks beside the CLIP backbone and extract high-resolution spatial features. It fuses early layers of the CLIP visual tokens and outputs: (1) attention bias (**A**) for refining ViT feature extraction, and (2) supplementary matrix (**S**) for making up high-resolution information. **Bottom**: Attention bias **A** manipulates the attention of transformer layers in ViT, and **S** is fused before outputting the 2D semantic feature $\mathbf{F}^{\mathbf{sem}}$ for LSS lifting.

tokens focus more on image-level information than spatial information, limiting perception performance. We thus propose slightly manipulating the feature extraction process by adding attention bias to transformer layers inside CLIP.

High-resolution side adaptor (HSA). As illustrated in Fig. 3, our High-resolution Side Adaptor (HSA) can be divided into the adaptor body and the adaptor head. *The adaptor body* consists of several residual blocks [16] parallel with the CLIP encoder and fuses multi-layer visual tokens from CLIP into the HSA body. Take the ViT-L CLIP variant with 24 transformer layers as an example. We fuse visual tokens from the 6^{th} and 12^{th} CLIP layers into the features after the 1^{st} and 2^{rd} HSA body blocks. Since features in HSA have a higher resolution (32×88) compared with visual tokens in CLIP (18×50), we resize the CLIP visual tokens to be the same size as HSA features, and then fuse them with element-wise addition after channel alignment via 1×1 convolutions. As shown in Fig. 3, the HSA body accompanies the first 3/4 layers (18 out of 24 in ViT-L) of the CLIP backbone, resulting in a high-resolution feature map $\mathbf{F}^{\mathbf{HSA}}$.

On the other hand, *the adaptor head* is responsible for manipulating the feature extraction process of the last 1/4 layers of the CLIP backbone, making them more suitable for scene understanding. Specifically, we first apply two MLPs on $\mathbf{F}^{\mathbf{HSA}}$ to obtain an attention bias matrix $\mathbf{A}$ and a supplementary matrix $\mathbf{S}$, as visualized in Fig. 3. The first matrix $\mathbf{A}$ is the attention bias for CLIP visual tokens. Specifically, take the calculation of the attention process within the i^{th} transformer layer in ViT as an example. We formulate this process as follows:

$$\mathbf{X}_{i+1} = \text{softmax}(\mathbf{Q}_i\mathbf{K}_i^T + \mathbf{A}_i\mathbf{A}_i^T)\mathbf{V}_i. \tag{4}$$

Here $\mathbf{X}_i$ represents the visual tokens in the i^{th} layer, and $\mathbf{Q}_i$, $\mathbf{K}_i$ and $\mathbf{V}_i$ are the linear transformations of $\mathbf{X}_i$. The attention bias $\mathbf{A}_i\mathbf{A}_i^T$ for the i^{th} layer is added to $\mathbf{Q}_i\mathbf{K}_i^T$ for directing the transformer to pay more attention on the spatial information. Here, we neglect some elements (*e.g.*, multiple attention heads) for convenience. Please refer to the supplementary material for details.

We assemble the visual tokens $\mathbf{X^{last}}$ after the last transformer layer and the supplementary matrix $\mathbf{S}$ into the feature $\mathbf{F^{sem}}$ for LSS lifting. We first reshape and interpolate $\mathbf{X^{last}}$ to become the same shape as $\mathbf{S}$, and then construct $\mathbf{F^{sem}}$ via MLPs and concatenation:

$$\mathbf{F^{sem}} = \big[\mathrm{MLP}_1(\mathbf{X^{last}}), \mathrm{MLP}_2(\big[\mathbf{X^{last}}, \mathbf{S}\big])\big] \tag{5}$$

Here, square brackets denote feature concatenation. As in Fig. 3, the output channel number of MLP_1 is larger than that of MLP_2 (i.e., $3:1$), as $\mathbf{S}$ is only designed as supplement to $\mathbf{X^{last}}$ provided by the CLIP encoder.

Training strategy. Our VEON is optimized with joint supervision on $\mathbf{O^{bin}}$ and $\mathbf{O^{sa}}$. Specifically, for the binary occupancy state $\mathbf{O^{bin}}$, we adopt cross entropy (CE) to construct the binary occupancy loss L_{bin}. Its ground truth $\hat{\mathbf{O}}^{\mathbf{bin}}$ can be derived from the point cloud $\mathbf{P}$ via offline post-processing [43]. As for supervising the semantic-aware embedding map $\mathbf{O^{sa}}$, we enforce each embedding O_i^{sa} of the i^{th} voxel to match the (pseudo) ground truth CLIP class embedding for the i^{th} voxel, namely $\hat{O}_i^{sa}$. The assignment of $\hat{\mathbf{O}}^{\mathbf{sa}}$ is critical. Here we apply an off-the-shelf 2D open-vocabulary segmentor SAN [50] as the pseudo ground truth. In the case that $C_s = \varnothing$, we project the i^{th} voxel onto the surrounding images based on the intrinsic and extrinsic camera parameters, and fetch the CLIP language embedding of corresponding open-vocabulary class j (output from SAN [50]) as optimization target, namely $\hat{O}_i^{sa} = F_j^{lan}$. Otherwise, if $C_s \neq \varnothing$, $\hat{O}_i^{sa}$ is replaced with the ground truth class embedding if and only if the annotation exists.

Then, we construct the feature alignment loss L_{sa} via cosine similarity. In other words, the feature alignment loss for each voxel i is calculated as $1 - \mathrm{cosine}(O_i^{sa}, \hat{O}_i^{sa})$. Traditionally, the cosine loss items of all voxels are averaged for calculating L_{sa}. However, as tail classes seldom exist in the training set, the vast majority of voxels will be trained to align with stuff classes (*e.g.*, road, grass) in this case, which is harmful to open-vocabulary recognition. In this paper, we propose to reweight the loss component of each voxel as follows:

$$L_{sa} = \frac{1}{|C|} \sum_{j \in C} \frac{1}{N'_j} \sum_{\hat{O}_i = j} 1 - \mathrm{cosine}(O_i^{sa}, \hat{O}_i^{sa}). \tag{6}$$

Here $N'_j = \Big|\Big\{i \mid \hat{O}_i = j\Big\}\Big|$, and $\hat{O}_i$ is calculated similar to Eq. 3 except that O_i^{sa} is replaced with $\hat{O}_i^{sa}$. Eq. 6 averages the voxel-level loss items within each class first, and then across all the classes. As tail classes occupy a much smaller number of voxels, they are prioritized during network optimization. Experiments prove that this design significantly alleviates the problem of tail classes. Finally, the loss in the training stage 2 L_{stg2} is the weighted sum of L_{bin} and L_{sa}.

4 Experiments

4.1 Experimental Settings

Dataset. Throughout our experiments, we select Occ3D [43] for evaluating our proposed VEON. Occ3D [43] is built on 700 training scenes and 150 validation scenes in the nuScenes dataset [9]. For each scene snapshot, nuScenes provides 6 images from surrounding cameras, the camera parameters for view transformation, and LiDAR point clouds. Beyond that, Occ3D [43] additionally annotates voxel-level semantic labels to serve as a benchmark for 3D occupancy prediction. These label masks have the resolution of $200 \times 200 \times 16$, with X-axis, Y-axis and Z-axis ranging respectively as $[-40, 40]$, $[-40, 40]$ and $[-1.0, 5.4]$ meters. The voxel size is $(0.4, 0.4, 0.4)$ meters. Following nuScenes LiDAR segmentation [15], one class out of 18 classes (with a special class "free") is assigned to each voxel. We collect the IoUs on all 17 normal classes (excluding "free"), and a mean IoU (mIoU) of these classes as the evaluation metrics. Following the Occ3D-nuScenes protocol, we only consider visible voxels during evaluation.
Implementation. We implement our VEON based on the BEVDet codebase [1]. Our experimental settings follow BEVDet [19,21], with the same data sampling, cropping, and augmentation strategy. We also employ the bevpoolv2 [20] in BEVDet for fast LSS [36]. We select AdamW [31] to be our network optimizer, with learning rate as 10^{-4} and weight decay as 10^{-2}. All our experiments are performed on 8 NVIDIA Tesla V100 GPUs, with a batch size of 1 on each GPU. During training stage 1, we adopt the MiDaS [38] with BEiT-L backbone [3] as our depth foundation model ϕ_{rel} and initialize the weights by pretraining on a mixed set of 12 depth datasets [7]. The camera input size for ϕ_{dp} is 256×704. During training stage 2, we load the 2D open-vocabulary semantic segmentor SAN [50] to generate pseudo labels for occupancy supervision. Since a frozen CLIP encoder exists inside SAN, we reuse the CLIP image encoder within our VEON framework. We test two variants of VEON throughout our experiments. VEON-B adopts the ViT-B CLIP variant with 12 transformer layers for semantic extraction, while the larger VEON-L adopts the ViT-L CLIP with 24 layers. The input image size in stage 2 is set as 512×1408.

4.2 Main Results

In the sequel, we evaluate our VEON on the Occ3D-nuScenes validation set [43]. We first report the 3D occupancy prediction results with either zero or partial manual semantic labels (*i.e.*, either $C_s = \varnothing$ or $C_s \neq \varnothing$), and then prove the open-vocabulary capability of VEON both quantitatively and qualitatively.
Occupancy without semantic labels ($C_s = \varnothing$). In Tab. 1, we investigate the performance of our VEON models trained without any manual semantic annotations. The first 6 rows in Tab. 1 list some supervised occupancy prediction models trained with full manual annotations. Performance of MonoScene [10], TPVFormer [24], OccFormer [53], CTF-Occ [53] is directly collected from [43], while BEVFormer [29] and BEVDet [21] are trained and evaluated on our own

Table 1: Performance of our VEON on Occ3D-nuScenes occupancy benchmark [9,43] (validation set) with $C_s = \varnothing$. We compare the VEON variants with existing occupancy predictors trained with (rows 1-6) or without (rows 7-9) manual labels.

Method	sem.	oth.	bar.	bic.	bus	car	c. v.	mot.	ped.	t. c.	tra.	tru.	d. s.	o. f.	sid.	ter.	man.	veg.	mIoU
MonoScene [10]	✓	1.8	7.2	4.3	4.9	9.4	5.7	4.0	3.0	5.9	4.5	7.2	14.9	6.3	7.9	7.4	1.0	7.7	6.06
TPVFormer [24]	✓	7.2	38.9	13.7	40.8	45.9	17.2	20.0	18.9	14.3	26.7	34.2	55.7	35.5	37.6	30.7	19.4	16.8	27.83
OccFormer [53]	✓	5.9	30.3	12.3	34.4	39.2	14.4	16.5	17.2	9.3	13.9	26.4	51.0	31.0	34.7	22.7	6.8	7.0	21.93
CTF-Occ [43]	✓	8.1	39.3	20.6	38.3	42.2	16.9	24.5	22.7	21.1	23.0	31.1	53.3	33.8	38.0	33.2	20.8	18.0	28.53
BEVFormer [29]	✓	9.6	47.8	24.2	48.7	54.0	20.9	28.8	27.5	26.7	32.8	38.8	81.7	40.3	50.5	52.9	43.8	37.5	39.19
BEVDet [21]	✓	8.8	45.2	19.1	43.5	50.2	23.7	19.8	22.9	20.7	31.9	37.7	80.3	37.0	50.5	53.4	47.1	41.9	37.28
SelfOcc-BEV [23]	✗	0.0	0.0	0.0	0.0	9.8	0.0	0.0	0.0	0.0	0.0	7.0	47.0	0.0	18.8	16.6	11.9	3.8	6.76
SelfOcc-TPV [23]	✗	0.0	0.0	0.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	7.11	53.0	0.0	23.6	25.2	12.0	4.6	7.97
OccNeRF [52]	✗	0.0	0.8	0.8	5.1	12.5	3.5	0.2	3.1	1.8	0.5	3.9	52.6	0.0	20.8	24.8	18.5	13.2	9.54
VEON-B (Ours)	✗	0.5	4.8	2.7	14.7	10.9	11.0	3.8	4.7	4.0	5.3	9.6	46.5	0.7	21.1	22.1	24.8	23.7	12.38
VEON-L (Ours)	✗	0.9	10.4	6.2	17.7	12.7	8.5	7.6	6.5	5.5	8.2	11.8	54.5	0.4	25.5	30.2	25.4	25.4	15.14

with the visible mask protocol [2]. On the other hand, rows 7-9 in Tab. 1 list three occupancy predictors trained without any manual annotations, including two variants of SelfOcc [23] structured as BEVFormer [29] and TPVFormer [24], as well as the OccNeRF [52] occupancy predictor. Finally, the last two rows demonstrate the performance of our VEON-B and VEON-L variants, which differ only in their CLIP backbones. Notice that our VEON utilizes the pseudo depth and the binary occupancy label from Occ3D [43] for supervision, but these two can both be derived from the raw point cloud. From Tab. 1, we observe that our VEON-B and VEON-L respectively achieve a competitive performance of 12.38 and 15.14 mIoU. The VEON-L variant surpasses SelfOcc-BEV, SelfOcc-TPV, and OccNeRF respectively by 8.38, 7.17, and 5.60 in mIoU. The performance boost of VEON comes from its capability of recognizing various objects, including some tail categories such as barrier, construction vehicles, bus and truck. For example, VEON-B and VEON-L obtain 4.8 and 10.4 IoU in barrier, while the corresponding performance for SelfOcc-BEV, SelfOcc-TPV and OccNeRF is only 0.0, 0.0 and 0.8. Similar phenomena can be observed within other classes.

Occupancy with partial semantic labels ($C_s \neq \varnothing$). In the $C_s \neq \varnothing$ setting, we have X seen classes with annotations and Y unseen classes without annotations. In Tab. 2, we select the VEON-L variant with two different X/Y divisions ($X/Y = 9/8$ and $X/Y = 13/4$). The $X/Y = 0/17$ variant is also listed as a baseline. The left X and the right Y classes are respectively seen and unseen classes [15]. From Tab. 2, we see that the mIoUs of VEON-L variants rise with X, which basically comes from the additional seen classes. Besides, the IoUs on unseen classes (*e.g.*, sidewalk, vegetation) are always competitive, contributing to the performance boost from another aspect.

Open-vocabulary language-driven retrieval. To quantitatively measure the open-vocabulary capability of our VEON, we evaluate our models on an open-vocabulary language-driven object retrieval benchmark proposed in [45]. Given

Table 2: Results on the Occ3D-nuScenes occupancy benchmark [9, 43] with $C_s \neq \varnothing$.

Method	X: seen	Y: uns.	oth.	bar.	bic.	bus	car	c. v.	mot.	ped.	t. c.	tra.	tru.	d. s.	o. f.	sid.	ter.	man.	veg.	mIoU
VEON-L	0	17	0.9	10.4	6.2	17.7	12.7	8.5	7.6	6.5	5.5	8.2	11.8	54.5	0.4	25.5	30.2	25.4	25.4	15.14
VEON-L	9	8	0.9	14.3	4.4	26.6	15.0	7.5	7.4	5.6	5.0	8.3	9.2	48.7	0.1	24.9	30.6	24.8	24.5	15.16
VEON-L	13	4	1.6	19.7	4.5	28.1	24.8	9.4	11.1	8.6	7.3	15.1	18.4	58.9	24.0	26.5	29.6	26.8	25.2	19.94

Table 3: Results on the open-vocabulary language-driven retrieval benchmark [45].

Method / mAP (%)	train (all)	train (vis)	val (all)	val (vis)	test (all)	test (vis)
MaskCLIP+ [54]	-	13.5	-	18.7	-	12.0
POP-3D [45]	15.3	15.6	24.1	24.7	12.6	13.6
VEON-L (Ours)	**37.7**	**38.5**	**35.3**	**36.1**	**30.9**	**31.3**

an open-vocabulary language prompt, models need to retrieve relevant LiDAR points in the 3D space. The mean Average Precision (mAP) metric is used for evaluation, similar to the conventional retrieval problems. The benchmark provides annotations on 42/27/36 scenes in the nuScenes training, validation, and testing set [9]. We strictly follow the open-sourced POP-3D codes to evaluate our VEON-L, with mAP on all points (mAP-all) and mAP on visible points (mAP-vis) as metrics. Note that our VEON-L variant is trained with zero semantic annotations ($X/Y = 0/17$), and is never tuned on any retrieval labels. Tab. 3 gives the performance comparison between MaskCLIP+ [54], POP-3D [45] and our VEON-L. Our VEON-L surpasses POP-3D by a significant margin, with 22.4%, 11.2%, 18.3% mAP-all boost, and 22.9%, 11.4%, 17.7% mAP-vis boost on the training, validation and testing set, respectively. This suggests that the 3D representation output from VEON aligns well with language embeddings of CLIP, with powerful capability of handling open-vocabulary tasks.

Visualization. In Fig. 4, we qualitatively show the open-vocabulary capability of our VEON. Here, we choose the VEON-L variant with $X/Y = 0/17$, meaning that all the visualized results are obtained without any manual semantic labels. We collect three scenes from the Occ3D-nuScenes [9, 43] validation set, and visualize one on each row. As in Fig. 4, column 1 shows the surrounding images, and columns 2-3 compare the occupancy results of ground truth and our VEON predicted ones. We see that our VEON shows promising results, keeping a great alignment with the ground truth. Columns 4-5 are illustrations of open-vocabulary retrieval tests. Specifically, we utilize language embeddings of unseen classes in the vocabulary to find out which voxels in 3D space belong to the class. Notice that the classes for retrieval tests are fine-grained "subclasses" instead of "superclasses" defined by nuScenes (see the supplementary material for details). From Fig. 4, we observe that our VEON succeeds in recognizing open-vocabulary classes such as stairs, gravel, and road sign. This proves the efficacy of our model in recognizing open-world objects on the road.

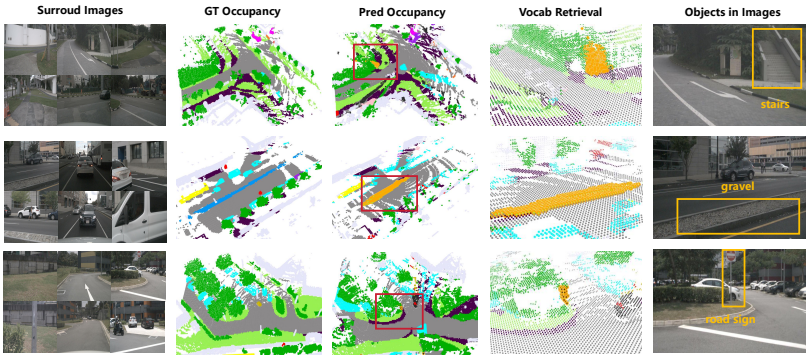

Fig. 4: Visualization of occupancy prediction (VEON-L) on the Occ3D-nuScenes occupancy benchmark [9,43] (validation set). We visualize the surrounding images (column 1), ground truth and predicted occupancy (column 2-3), and the retrieval results of certain open-vocabulary classes (column 4-5). Our VEON-L demonstrates the capability of recognizing unseen objects (in orange), such as stairs, gravel, and road signs.

Table 4: Ablation study on training the depth module ϕ_{dp} via low rank adaptation.

Variant	LoRA	bar.	bic.	ped.	tru.	veg.	mIoU
VEON-B	✗	4.7	3.6	5.1	7.3	23.2	11.57
VEON-B	✓	4.8	2.7	4.7	9.6	23.7	12.38
VEON-L	✗	10.2	4.9	3.0	12.4	22.8	14.57
VEON-L	✓	10.4	6.2	6.5	11.8	25.4	15.14

Table 5: Ablation study on the class reweighting strategy for the tail class trap.

Variant	RW	bar.	bic.	ped.	tru.	veg.	mIoU
VEON-B	✗	4.4	0.0	0.0	8.3	25.5	10.39
VEON-B	✓	4.8	2.7	4.7	9.6	23.7	12.38
VEON-L	✗	8.9	4.9	3.5	13.6	25.5	14.18
VEON-L	✓	10.4	6.2	6.5	11.8	25.4	15.14

4.3 Ablation Study

Depth estimation. VEON requires a depth estimation module ϕ_{dp}, which is trained with low-rank adaptation (LoRA) for robust domain transfer. We conduct a thorough study on it in Tab. 4. We follow the $C_s = \varnothing$ setting, with ViT-B and ViT-L as the CLIP backbone. We only report the IoUs of five representative classes (barrier, bicycle, pedestrian, truck, and vegetable) and the overall mIoU of all 17 classes. From Tab. 4, we discover that for the two variants of VEON, the mIoU rises moderately by 0.81 and 0.57. We conclude that preciser depth estimation module ϕ_{dp} leads to preciser 3D occupancy prediction results.

Class reweighting. We propose the class reweighting strategy (see Eq. 6) to escape from the tail class trap in open-vocabulary occupancy prediction. Experiments are conducted in Tab. 5 on both variants of VEON. We observe that adding the class reweighting strategy brings an increase of 1.99 and 0.96 in terms of mIoU. The reason can be found in the class-wise IoUs. Take the VEON-B variant as an example. After integrating the strategy, the IoU of "bicycle" rises from 0.0 to 2.7, and the IoU of "pedestrian" rises from 0.0 to 4.7. This indicates that the class reweighting strategy enables the network to recognize tail classes.

Table 6: Ablation study on the High-resolution Side Adaptor (HSA) module. We remove (−) / add (+) features from/to the VEON-B to check component efficacy.

Variant	bar.	bic.	ped.	tru.	veg.	mIoU
VEON-B	4.8	2.7	4.7	9.6	23.7	12.38
– Whole HSA	4.1	4.0	4.0	9.0	24.7	11.68
– Attention bias **A**	4.1	3.0	4.7	9.3	23.7	11.75
– Supplementary **S**	4.0	3.8	4.0	7.7	24.4	11.61
+ Token Offsets	4.2	3.7	5.1	9.3	24.4	12.06

Table 7: Statistical results on the parameters (M), trainable parameters (M) and the trainable fraction (%) within each component in VEON-L.

Models	Param	Param-Tr.	Frac.
MiDaS ϕ_{rel}	328.7	0.9	0.3%
D-Adaptor ϕ_{r2m}	16.8	16.8	100.0%
D-Model ϕ_{dp}	345.5	17.7	5.1%
CLIP ViT-L	304.3	0	0%
HSA	13.5	13.5	100.0%
3D Layers	14.8	14.8	100.0%
VEON-L	678.1	46.0	6.8%

High-resolution side adaptor (HSA). We test the indispensability of the HSA module in Tab. 6. Given our baseline as VEON-B, we first try a trivial solution of directly lifting the CLIP feature as in row 2. The 0.70 mIoU decrease proves the efficacy of adapting the CLIP foundation model. Then, we remove the attention bias matrix **A** and the supplementary matrix **S** respectively. As on rows 3-4 in Tab. 6, we suffer from 0.63 and 0.77 mIoU decrease. This indicates that our HSA succeeds in refining the CLIP features and provides a high-resolution supplement for semantic extraction. In row 5, we try another solution of linearly predicting and adding offsets to CLIP visual tokens. The 0.32 decrease infers that our attention bias solution is more feasible.

Parameter statistics. From Tab. 7, we do statistics for the parameters of each component within our VEON-L model. While our model has a tremendous parameter number of 678.1M due to the integration of two foundation models, the trainable parts within our VEON only occupy a small fraction of 6.8%, with 17.7M in the depth module (stage 1) and 28.3M in the occupancy predictor (stage 2). This affirms that our VEON remains lightweight.

5 Concluding Remarks

In this paper, we design a VEON framework for **V**ocabulary-**E**nhanced **O**ccupancy predictio**N**. We adopt a decoupled structure for 3D occupancy prediction, which assembles a depth foundation model MiDaS and a semantic foundation model CLIP. As directly integrating these two models meets challenges, we adapt MiDaS with a relative-metric-bin adaptor and low-rank adaptation (LoRA) for domain transfer, and equip CLIP with a high-resolution side adaptor (HSA) for enhanced feature extraction. We also design a class reweighting loss to escape from the tail class trap. Our VEON method shows competitive performance on the Occ3D-nuScenes dataset and strong capability of recognizing unseen and fine-grained classes. We hope our work could herald a rethinking of the construction pipeline of open-vocabulary 3D occupancy prediction models.

Acknowledgements. This work was supported by NSFC (62322113, 62376156), Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102), and the Fundamental Research Funds for the Central Universities.

References

1. The bevdet codebase. `https://github.com/HuangJunJie2017/BEVDet`, accessed on 2023-10-28
2. Cvpr 2023 3d occupancy prediction challenge. `https://github.com/CVPR2023-3D-Occupancy-Prediction/CVPR2023-3D-Occupancy-Prediction`, accessed on 2023-10-28
3. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. In: ICLR (2021)
4. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: ICCV. pp. 9297–9307 (2019)
5. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: CVPR. pp. 4009–4018 (2021)
6. Bhat, S.F., Alhashim, I., Wonka, P.: Localbins: Improving depth estimation by learning local distributions. In: ECCV. pp. 480–496 (2022)
7. Bhat, S.F., Birkl, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023)
8. Birkl, R., Wofk, D., Müller, M.: Midas v3. 1–a model zoo for robust monocular relative depth estimation. arXiv preprint arXiv:2307.14460 (2023)
9. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR. pp. 11621–11631 (2020)
10. Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: CVPR. pp. 3991–4001 (2022)
11. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021)
12. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: CVPR. pp. 7010–7019 (2023)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Eigen, D., Puhrsch, C., Fergus, R.: Depth map prediction from a single image using a multi-scale deep network. In: NeurIPS. pp. 2366–2374 (2014)
15. Fong, W.K., Mohan, R., Hurtado, J.V., Zhou, L., Caesar, H., Beijbom, O., Valada, A.: Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. RA-L **7**(2), 3795–3802 (2022)
16. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
17. Hu, E.J., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: ICLR (2021)
18. Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: CVPR. pp. 17853–17862 (2023)

19. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
20. Huang, J., Huang, G.: Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. arXiv preprint arXiv:2211.17111 (2022)
21. Huang, J., Huang, G., Zhu, Z., Yun, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021)
22. Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q., Yang, R.: The apolloscape open dataset for autonomous driving and its application. TPAMI **42**(10), 2702–2719 (2019)
23. Huang, Y., Zheng, W., Zhang, B., Zhou, J., Lu, J.: Selfocc: Self-supervised vision-based 3d occupancy prediction. arXiv preprint arXiv:2311.12754 (2023)
24. Huang, Y., Zheng, W., Zhang, Y., Zhou, J., Lu, J.: Tri-perspective view for vision-based 3d semantic occupancy prediction. In: CVPR. pp. 9223–9232 (2023)
25. Huang, Z., Wu, X., Chen, X., Zhao, H., Zhu, L., Lasenby, J.: Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. arXiv preprint arXiv:2309.00616 (2023)
26. Jiang, B., Chen, S., Xu, Q., Liao, B., Chen, J., Zhou, H., Zhang, Q., Liu, W., Huang, C., Wang, X.: Vad: Vectorized scene representation for efficient autonomous driving. In: ICCV. pp. 8340–8350 (2023)
27. Li, Y., Yu, Z., Choy, C., Xiao, C., Alvarez, J.M., Fidler, S., Feng, C., Anandkumar, A.: Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In: CVPR. pp. 9087–9098 (2023)
28. Li, Z., Snavely, N.: Megadepth: Learning single-view depth prediction from internet photos. In: CVPR. pp. 2041–2050 (2018)
29. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV. pp. 1–18 (2022)
30. Liu, K., Zhan, F., Zhang, J., Xu, M., Yu, Y., Saddik, A.E., Theobalt, C., Xing, E., Lu, S.: Weakly supervised 3d open-vocabulary segmentation. arXiv preprint arXiv:2305.14093 (2023)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2018)
32. Lu, S., Chang, H., Jing, E.P., Boularias, A., Bekris, K.: Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In: CoRL (2023)
33. Miao, R., Liu, W., Chen, M., Gong, Z., Xu, W., Hu, C., Zhou, S.: Occdepth: A depth-aware method for 3d semantic scene completion. arXiv preprint arXiv:2302.13540 (2023)
34. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
35. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR. pp. 815–824 (2023)
36. Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: ECCV. pp. 194–210 (2020)
37. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763 (2021)
38. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. TPAMI **44**(3), 1623–1637 (2020)

39. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV. pp. 746–760 (2012)
40. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR. pp. 2446–2454 (2020)
41. Tan, Z., Dong, Z., Zhang, C., Zhang, W., Ji, H., Li, H.: Ovo: Open-vocabulary occupancy. arXiv preprint arXiv:2305.16133 (2023)
42. Tang, P., Wang, Z., Wang, G., Zheng, J., Ren, X., Feng, B., Ma, C.: Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction. In: CVPR. pp. 15035–15044 (2024)
43. Tian, X., Jiang, T., Yun, L., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. arXiv preprint arXiv:2304.14365 (2023)
44. Tong, W., Sima, C., Wang, T., Chen, L., Wu, S., Deng, H., Gu, Y., Lu, L., Luo, P., Lin, D., et al.: Scene as occupancy. In: ICCV. pp. 8406–8415 (2023)
45. Vobecky, A., Siméoni, O., Hurych, D., Gidaris, S., Bursuc, A., Pérez, P., Sivic, J.: Pop-3d: Open-vocabulary 3d occupancy prediction from images. In: NeurIPS. pp. 50545–50557 (2023)
46. Wang, G., Wang, Z., Tang, P., Zheng, J., Ren, X., Feng, B., Ma, C.: Occgen: Generative multi-modal 3d occupancy prediction for autonomous driving. arXiv preprint arXiv:2404.15014 (2024)
47. Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. In: ICCV. pp. 17850–17859 (2023)
48. Wei, Y., Zhao, L., Zheng, W., Zhu, Z., Zhou, J., Lu, J.: Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In: ICCV. pp. 21729–21740 (2023)
49. Xian, K., Zhang, J., Wang, O., Mai, L., Lin, Z., Cao, Z.: Structure-guided ranking loss for single image depth prediction. In: CVPR. pp. 611–620 (2020)
50. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: CVPR. pp. 2945–2954 (2023)
51. Yao, Y., Luo, Z., Li, S., Zhang, J., Ren, Y., Zhou, L., Fang, T., Quan, L.: Blendedmvs: A large-scale dataset for generalized multi-view stereo networks. In: CVPR. pp. 1790–1799 (2020)
52. Zhang, C., Yan, J., Wei, Y., Li, J., Liu, L., Tang, Y., Duan, Y., Lu, J.: Occnerf: Self-supervised multi-camera occupancy prediction with neural radiance fields. arXiv preprint arXiv:2312.09243 (2023)
53. Zhang, Y., Zhu, Z., Du, D.: Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. arXiv preprint arXiv:2304.05316 (2023)
54. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV. pp. 696–712 (2022)