# VEON: Vocabulary-Enhanced Occupancy Prediction — Supplementary Material —

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

In the supplementary material, we first present some details of our VEON framework, including class embedding generation, subclass division, depth loss, feature alignment, and attention bias. Then, we provide more quantitative results and visualization on the nuScenes [4] dataset to demonstrate the open-vocabulary capability of our VEON. Finally, we discuss the potential negative societal impact and limitations of our work.

## A Framework Details

### A.1 Class Embedding Generation

In our VEON framework, we align the voxel-wise semantic-aware occupancy map $\mathbf{O}^{\mathbf{sa}}$ with the CLIP [10] language embeddings of specific classes, formulated as Eq. 6 in the manuscript. To generate class embeddings suitable for open-vocabulary recognition, we combine multiple natural language templates to jointly describe each single class. We then average the corresponding embeddings output from the CLIP language encoder to obtain the required embedding for each class [8, 14]. In practice, 14 templates are collected following SAN [13]. An example is "This is a photo of a {}", where {} represents the class name text. Tab. A shows the detailed list of the prompt templates.

### A.2 Subclass Division

In VEON, we need to define an overall class set $C_{all}$ for open-vocabulary recognition. The selection of $C_{all}$ seems to be trivial at first glance, as Occ3D-nuScenes [4, 12] natively classifies all voxels into 17 non-free classes [7] and 1 free class. However, we find such coarse-grained class division unsuitable for open-vocabulary tasks. For example, the first non-free class in Occ3D-nuScenes is termed as "others", obviously a meaningless class description. Voxels labeled as "others" may

⋆ Corresponding author.

| |
|---|
| “a photo of a {}.", |
| “This is a photo of a {}", |
| “There is a {} in the scene", |
| “There is the {} in the scene", |
| “a photo of a {} in the scene", |
| “a photo of a small {}.", |
| “a photo of a medium {}.", |
| “a photo of a large {}.", |
| “This is a photo of a small {}.", |
| “This is a photo of a medium {}.", |
| “This is a photo of a large {}.", |
| “There is a small {} in the scene.", |
| “There is a medium {} in the scene.", |
| “There is a large {} in the scene.", |

**Table A:** List of prompt templates used in VEON. We keep the same templates as those utilized in SAN [13].

be occupied by various subclasses of objects, including animal, trash can, skateboard, personal mobility, and ego vehicle, etc. Therefore, using the coarse-grained class terms provided by Occ3D-nuScenes is improper.

To better suit the class embeddings to the open-vocabulary task, we adopt a *subclass division strategy* that divides the original superclasses collected from Occ3D-nuScenes into separate subclasses. This enlarges the overall (non-free) class set $C_{all}$ from the original 17 superclasses to $\sim 60$ subclasses. The detailed list of subclasses, summarized from the official nuScenes description of these coarse superclasses, is shown in Tab. B.

With the subclass division strategy, we achieve a fine-grained understanding of the surrounding 3D space during inference. For instance, tree, bushes and other plants could be distinguished into different subclasses, despite that they all belong to the superclass “vegetation”. Note that for quantitative evaluation on the Occ3D-nuScenes benchmark, we project the subclasses back to the superclasses according to Tab. B, and calculate the class-wise IoUs and overall mIoU metrics.

### A.3 Depth Loss

In the first stage of VEON, we supervise the metric depth map $\mathbf{D}$ with a pixel-wise scale-invariant depth loss $L_{pix}$. Suppose $d_i$ is the $i$-th pixel of $\mathbf{D}$, and $\hat{d}_i$ is the $i$-th pixel of the corresponding ground truth $\hat{\mathbf{D}}$. Here $\hat{\mathbf{D}}$ is obtained by projecting the point cloud $\mathbf{P}$ onto the camera plane. Then, we strictly follow [2, 3, 6] to calculate the pixel-wise scale-invariant depth loss $L_{pix}$ as:

$$L_{pix} = \sqrt{\frac{1}{N_{pix}} \sum_i g_i^2 - \frac{\alpha}{N_{pix}^2} \left( \sum_i g_i \right)^2}, \tag{A}$$

| Superclass | List of subclasses |
|---|---|
| others | debris, animal, personal mobility, skateboard, segway, scooter, stroller, wheelchair, trash bag, trash can, wheelbarrow, bicycle rack, ambulance, police vehicle. |
| barrier | traffic barrier. |
| bicycle | bicycle. |
| bus | bus. |
| car | car, sedan, hatch-back, wagon, van, SUV, jeep. |
| const. veh. | construction vehicle. |
| motorcycle | motorcycle. |
| pedestrian | pedestrian, construction worker, police officer. |
| traffic cone | traffic cone. |
| trailer | trailer. |
| truck | truck. |
| driv. surf. | road. |
| other flat | traffic island, traffic delimiter, rail track, lake, river. |
| sidewalk | sidewalk, pedestrian walkway, bike path. |
| terrain | grass, rolling hill, soil, sand, gravel. |
| manmade | building, wall, guard rail, fence, drainage, hydrant, banner, street sign, traffic light, parking meter, stairs. |
| vegetation | vegetation, plants, bushes, tree. |

**Table B:** The subclass list used in VEON. The superclasses are kept the same as the predefined classes in nuScenes [4, 7], and the subclasses are summarized from the official class description from the nuScenes LiDAR segmentation [7] benchmark.

where $N_{pix}$ is the total number of pixels on $\mathbf{D}$, $\alpha$ is a constant, and $g_i$ is the log-difference between each depth $d_i$ and its corresponding ground truth $\hat{d}_i$ on $\hat{\mathbf{D}}$, namely $g_i = \log d_i - \log \hat{d}_i$. As is explained in the manuscript, $L_{pix}$ ensures the shape and smoothness of the output metric depth map $\mathbf{D}$. This design helps retain knowledge from the depth foundation model MiDaS [11], and is also beneficial to the subsequent bin depth transformation. As an implementation detail, $L_{pix}$ is calculated on the 8×-downsampled depth maps compared with the input surrounding images. Also, for those pixels without pseudo depth projected from the point cloud $\mathbf{P}$, they will never be involved in loss calculation.

### A.4 Feature Alignment

In VEON, we align the semantic-aware occupancy map $\mathbf{O}^{\mathbf{sa}}$ with existing 2D pixel-wise CLIP-aligned embeddings, as Eq. 6 in the manuscript. We design to utilize an off-the-shelf 2D open-vocabulary segmentor SAN [13] to generate the 2D pixel-wise CLIP-aligned embeddings. Then, $\mathbf{O}^{\mathbf{sa}}$ is supervised via 3D-to-2D projection and feature alignment. We will dive into detail in the sequel.

First, we introduce how to generate the 2D CLIP-aligned embeddings with SAN [13]. SAN is an open-vocabulary 2D segmentor composed of a CLIP image

encoder and a side adaptor network. It utilizes a query-based methodology to generate (1) class-agnostic object mask proposals and (2) proposal-wise embeddings by manipulating the CLIP attention layers. The final output of SAN is a pixel-wise classification map for the input 2D surrounding images. On each pixel, $|C_{all}|$ probabilities are given, indicating the likelihood that the pixel belongs to each particular class. For the detailed architecture of SAN, we refer readers to [13].

Second, we present details of the feature alignment process. For each voxel in the 3D space, we first project the center of the voxel onto the surrounding images based on the intrinsic and extrinsic camera parameters. The following procedure shifts according to the availability of semantic label on the voxel. If there exists no superclass label on the voxel, we select the subclass in $C_{all}$ with the highest classification probability on the projected pixel, and pick the corresponding CLIP language embedding as the (pseudo) ground truth for that voxel. If there exists a superclass label on the voxel (typically when $C_s \neq \varnothing$), we select the subclass restricted by the superclass annotation, and other procedures are kept the same. For example, consider a 3D voxel labeled as the superclass "vegetation". We refer to the projected 2D pixel on surrounding images and fetch the output of SAN on that pixel. In this case, only 4 subclasses, including "vegetation", "plants", "bushes" and "tree" will be regarded as candidate subclasses (see Tab. B), and the single subclass with the highest classification probability will be selected as pseudo ground truth class for supervising $\mathbf{O}^{\mathbf{sa}}$. The class embedding to align is then fetched from the CLIP language encoder.

### A.5 Attention Bias

We design a High-resolution Side Adaptor (HSA) to make the pretrained CLIP better suited to the open-vocabulary occupancy prediction task. The key idea is to maintain a side adaptor that absorbs early layers of visual tokens from CLIP and then outputs an attention bias matrix $\mathbf{A}$ to manipulate the attention layers in the later layers of CLIP. The HSA module has a higher resolution than the CLIP backbone, contributing to fine-grained scene understanding by providing high-resolution supplementary information.

Here, we focus on how the attention bias matrix $\mathbf{A}$ affects the forward pipeline of CLIP transformer layers. The CLIP backbone follows the ViT [5] architecture. Images are sliced into patches of $16 \times 16$, encoded into initial visual tokens $\mathbf{X}_0^{[v]}$, and concatenated with an initial global [cls] token $\mathbf{X}_0^{[cls]}$. The tokens $\mathbf{X}_0 = [\mathbf{X}_0^{[v]}, \mathbf{X}_0^{[cls]}]$ go through multiple transformer layers (12/24 layers for ViT-B/ViT-L), where the operation $[\cdot, \cdot]$ means token concatenation. Each transformer layer comprises multi-head attention, feed-forward network, and layer normalization [5]. Our attention bias matrix $\mathbf{A}$ operates solely in the multi-head attention. In the manuscript, we simplify the process as follows (copied from Eq. 4 in the manuscript):

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

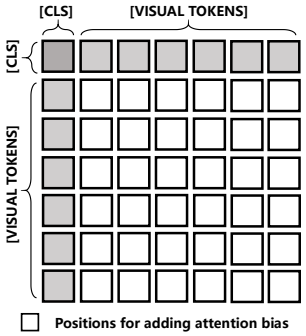


**Fig. A:** Positions for adding attention bias (blank squares).

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.

In fact, we ignore three details in the above formulation. *First*, in Eq. B, we omit the feed-forward network and layer normalization in each transformer layer. In other words, the output in Eq. B should additionally pass through the feed-forward network and layer normalization to become the input tokens $\mathbf{X}_{i+1}$ of the next transformer layer. *Second*, the global [cls] token $\mathbf{X}_i^{[cls]}$ is ignored in Eq. B. As is shown in Fig. A, each attention operation involves the feature interaction between $\mathbf{X}_i^{[v]}$ and $\mathbf{X}_i^{[cls]}$. Our attention bias $\mathbf{A}_i$ for layer $i$ is added only to the attention parts of visual tokens, *i.e.*, the blank positions in Fig. A. Also, the scale constant $\frac{1}{\sqrt{d}}$ is also omitted in Eq. B ($d$ is the dimension). *Third*, multiple attention heads are calculated separately in each transformer layer. In our VEON, the attention biases are also separate for each head. This means that the HSA head needs to output the attention bias for all the attention heads in all the later layers of CLIP. For example, in the ViT-L CLIP, the attention bias matrix $\mathbf{A}$ has a size of $(\frac{H}{16} \times \frac{W}{16}) \times 6 \times 8 \times 32$. Here $H$ and $W$ are the height and width of the input image, 6 is the number of layers being manipulated by $\mathbf{A}$, and 8 is the number of heads in each multi-head attention. Then, the inner production within $\mathbf{A}\mathbf{A}^T$ in Eq. B is performed on the last dimension of $\mathbf{A}$, with the head dimension as 32. In other words, $\mathbf{A}\mathbf{A}^T$ has the size of $(\frac{H}{16} \times \frac{W}{16}) \times (\frac{H}{16} \times \frac{W}{16}) \times 6 \times 8$, indicating the layer-wise and head-wise attention biases in the transformer layers.

## B More Experimental Results

### B.1 Occupancy Prediction with $C_s \neq \varnothing$

In Tab. 2 in the manuscript, we investigate the occupancy prediction performance of our VEON-L in the $C_s \neq \varnothing$ setting. Here we repeat the experiment on another variant, namely VEON-B, in Tab. C. Remember that with $C_s \neq \varnothing$, we have $X$

**Table C:** Performance of our VEON-B on the Occ3D-nuScenes occupancy benchmark [4, 12] in the $C_s \neq \varnothing$ setting.

| 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-B | 0 | 17 | 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-B | 9 | 8 | 1.0 | 9.5 | 3.5 | 23.8 | 16.3 | 9.3 | 5.47 | 3.5 | 4.7 | 5.1 | 6.7 | 45.0 | 0.6 | 21.1 | 21.8 | 24.0 | 24.2 | 13.26 |
| VEON-B | 13 | 4 | 0.9 | 9.5 | 4.8 | 26.8 | 25.7 | 10.4 | 7.9 | 5.2 | 9.4 | 10.1 | 16.4 | 62.0 | 14.7 | 23.4 | 19.3 | 24.6 | 24.5 | 17.38 |

seen classes with semantic annotations and $Y$ unseen classes without semantic annotations. Similar to Tab. 2, we pick two different $X/Y$ divisions ($X/Y = 9/8$ and $X/Y = 13/4$), and the $X/Y = 0/17$ variant is also provided for comparison. Note that the left $X$ and the right $Y$ classes in Tab. C are seen and unseen classes [7], respectively. In other words, in the $X/Y = 9/8$ case, classes from "others" to "traffic cone" are seen classes, while the classes from "trailer" to "vegetation" are unseen classes.

From Tab. C, we observe three phenomena. *First*, similar to the results of VEON-L, the VEON-B variant also benefits from the increase in seen classes $X$. When $X$ rises from $0 \rightarrow 9 \rightarrow 13$, the mIoU also increases from $12.38 \rightarrow 13.26 \rightarrow 17.38$. This overall mIoU increase primarily comes from the additional seen classes, such as the $14.7 \rightarrow 23.8 \rightarrow 26.8$ IoU increase in the class "bus", while the performance on unseen classes remains stable. *Second*, comparing Tab. 2 with Tab. C, we discover that with all three types of $X/Y$ settings, the VEON-L variants surpass the VEON-B variants respectively by 2.76, 1.90, and 2.56 mIoU. This affirms that 2D data prior originating from large-scale vision language pre-training is critical for 3D open-vocabulary tasks such as occupancy prediction. *Third*, the VEON variants do not perform well on certain classes when they are not explicitly annotated, *e.g.*, "other flats". This can attributed to the failure of the open-vocabulary segmentor SAN [13] in recognizing superclass "other flats", which includes stuff subclasses such as traffic island, traffic delimiter, river, etc.

### B.2 More Visualization

In Fig. B, we qualitatively show the open-vocabulary capability of our VEON, as a supplement to Fig. 4 in the manuscript. All settings are kept the same as Fig. 4, with VEON-L as our model and the Occ3D-nuScenes [4, 12] dataset as the benchmark. Remember that the selected VEON-L is trained without any semantic labels. In Fig. B, column 1 shows the surrounding images, and columns 2-3 compare the ground truth occupancy and our VEON predicted ones. Columns 4-5 visualize the open-vocabulary voxel retrieval results. Specifically, we utilize language embedding of any unseen subclass in $C_{all}$ to search for which voxels in 3D space belong to that subclass. Each occupancy snapshot in column 4 is an enlarged view of the local occupancy in the red box in column 3, and the camera image in column 5 has the same viewing angle as the occupancy snapshot

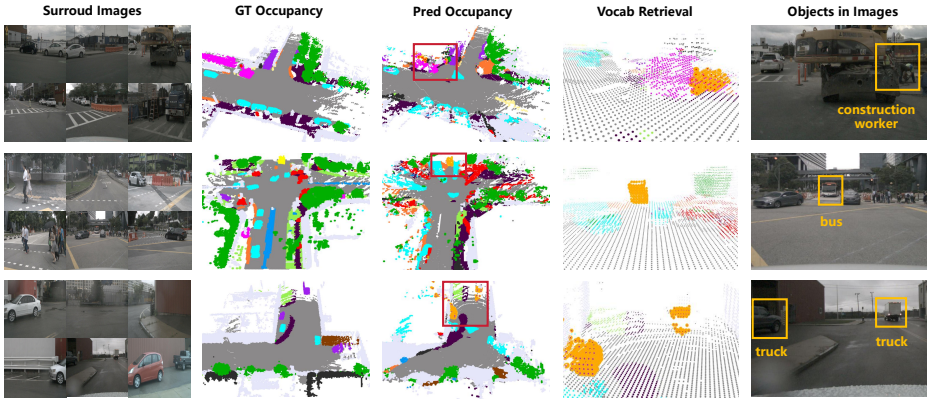


**Fig. B:** More visualization of occupancy prediction (VEON-L) on the Occ3D-nuScenes occupancy benchmark [4,12] (validation set). We visualize the surrounding images (column 1), ground truth and predicted occupancy (column 2-3), and the open-vocabulary retrieval results of certain classes (column 4-5). We see that our VEON-L not only shows competitive occupancy prediction results but also succeeds in recognizing unseen objects (colored in orange), such as construction worker, bus, truck, etc. Remember that the above results are obtained without any semantic labels.

in column 4. The target objects retrieved by natural language are highlighted with orange in columns 4-5. From Fig. B, we observe that our VEON succeeds in recognizing open-vocabulary classes such as construction worker, bus, and truck. This proves the efficacy of our model in open-vocabulary 3D occupancy prediction in the wild.

## C Potential Societal Impact and Limitations

### C.1 Potential Societal Impact

Our VEON aims to predict open-vocabulary 3D occupancy, which is a central task in autonomous driving. Such perception around the ego car is not related to privacy-related issues. However, imperfect occupancy prediction results may lead to failure in subsequent planning and control, causing traffic accidents and casualties. We believe that our work makes a solid step towards robust and practical open-vocabulary 3D occupancy prediction, and can inspire further advancements in this essential module for autonomous driving.

### C.2 Limitations

One major limitation of VEON is that its performance is hindered by the frozen foundation models. For instance, VEON does not perform well on superclasses such as "other flat" (see Tab. 1 in the manuscript). This can attributed to the failure of the open-vocabulary segmentor SAN [13] in recognizing stuff within

"other flats", including subclasses such as traffic island, river, etc. And the performance of SAN relies on the pretrained CLIP backbone [13]. Since transferring knowledge from pretrained foundation models is a prevailing trend, we may consider leveraging more powerful Vision-Language Models (VLMs) in the future. These VLMs, *e.g.* MiniGPT-4 [15], LLaVa [9], and Qwen-VL [1], possess strong vision-language comprehension and reasoning capabilities, which may benefit open-vocabulary 3D occupancy prediction.

## References

1. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966 (2023)
2. Bhat, S.F., Alhashim, I., Wonka, P.: Adabins: Depth estimation using adaptive bins. In: CVPR. pp. 4009–4018 (2021)
3. 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)
4. 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)
5. 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)
6. 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)
7. 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)
8. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: CVPR. pp. 7061–7070 (2023)
9. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS. pp. 49250–49267 (2023)
10. 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)
11. 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)
12. 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)
13. Xu, M., Zhang, Z., Wei, F., Hu, H., Bai, X.: Side adapter network for open-vocabulary semantic segmentation. In: CVPR. pp. 2945–2954 (2023)
14. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: ECCV. pp. 736–753 (2022)

15. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)