3D Occupancy Prediction with Low-Resolution Queries via Prototype-aware View Transformation

Appendix

A. Dataset Details

In this section, we provide further details of the evaluation datasets Occ3D-nuScenes [14] and SemanticKITTI [2]. Occ3D-nuScenes offers six surrounding multi-view camera images and a densely labeled semantic occupancy ground truth in the size of $200 \times 200 \times 16$. Each voxel represents one of 17 class labels in the $0.4m \times 0.4m \times 0.4m$ sized space of the real world. 700 and 150 scenes are used for training and validation, respectively. SemanticKITTI provides 22 outdoor driving scenarios, which includes 10 scenarios for training, 1 for validation, and 11 for testing. The ground truth scenes are voxelized in the dimension of $256 \times 256 \times 32$. Each voxel are in the size of $0.2m \times 0.2m \times 0.2m$ and are assigned with one of 20 classes (19 semantic and 1 free space). To facilitate a broad understanding, we provide the configurations of Occ3D-nuScenes and SemanticKITTI datasets in the table below (see Table A.1).

Table A.1. Comparison of dataset configurations.

Dataset	Output Res.	Voxel Size	# Classes	# Images
Occ3d-nuScenes	$200 \times 200 \times 16$	$0.4m \times 0.4m \times 0.4m$	12	Multi-view
SemanticKITTI	$256 \times 256 \times 32$	$0.2m \times 0.2m \times 0.2m$	20	Single

B. Experimental Details

B.1. Implementation Details

Occ3D-nuScenes. We fix the weights for the occupancy loss and Lovasz loss as $\lambda_1 = 10.0$ and $\lambda_2 = 1.0$, respectively. Then we search the best-performing values of λ_3 and λ_4 within $\{1.0, 0.1, 0.01\}$, where we empirically find the optimal combination of $\lambda_3 = 0.1$ and $\lambda_4 = 0.01$ for `Tiny`, and $\lambda_3 = 1.0$ and $\lambda_4 = 1.0$ for `Small` and `Base`. Similarly, we search the values of τ_{cls} and τ_{cons} in between $\{0.3, 0.5\}$ and find that $\tau_{\text{cls}} = 0.3$ and $\tau_{\text{cons}} = 0.5$ produces the best result in the `Tiny` setting, and $\tau_{\text{cls}} = 0.3$ and $\tau_{\text{cons}} = 0.3$ for the `Small` and `Base` setting. All models are trained with 4 NVIDIA A100 GPUs. The experiment configurations of Occ3D-nuScenes are summarized in Table A.2.

SemanticKITTI. For this task, we integrate our proposed strategies into two baseline models, VoxFormer [9] and Symphonies [7]. We empirically determine the optimal hyperparameters of $\{\lambda_3, \lambda_4, \tau_{\text{cls}}, \tau_{\text{cons}}\}$: $\{1.0, 1.0, 0.3, 0.3\}$ for VoxFormer-`Small`, $\{1.0, 1.0, 0.5, 0.3\}$ for VoxFormer-`Base`, $\{1.0, 1.0, 0.3, 0.3\}$ for Symphonies-`Small` and $\{1.0, 1.0, 0.3, 0.5\}$ for Symphonies-`Base`. Similar to Occ3D-nuScenes, the models are trained with 4 NVIDIA A100 GPUs. The experimental configurations are summarized in Table A.3.

Table A.2. Experimental Configurations of Occ3D-nuScenes.

	Specification	`Tiny`	`Small`	`Base`
Model	Image resolution	432×800	432×800	432×800
	Image backbone	ResNet-101	ResNet-101	ResNet-101
	FPN level	5	5	5
	Channel	256	256	256
	Encoder layer	3	3	3
	Decoder layer	2	2	2
	Query size	$50 \times 50 \times 4$	$50 \times 50 \times 16$	$100 \times 100 \times 16$
Training	Optimizer	AdamW	AdamW	AdamW
	Learning rate	2e-4	2e-4	2e-4
	Scheduler	Cosine Annealing	Cosine Annealing	Cosine Annealing
	Weight decay	0.01	0.01	0.01
	Epoch	12	12	12

Table A.3. Experimental Configurations on SemanticKITTI.

		VoxFormer		Symphonies	
	Specification	`Small`	`Base`	`Small`	`Base`
Model	Image resolution	370×1200	370×1200	370×1206	370×1206
	Image backbone	ResNet-50	ResNet-50	ResNet-50	ResNet-50
	FPN level	5	5	3	3
	Channel	128	128	128	128
	Encoder layer	3	3	1	1
	Decoder layer	-	-	3	3
	Query size	$64 \times 64 \times 8$	$128 \times 128 \times 16$	$64 \times 64 \times 8$	$128 \times 128 \times 16$
Training	Optimizer	AdamW	AdamW	AdamW	AdamW
	Learning rate	2e-4	2e-4	2e-4	2e-4
	Scheduler	Cosine Annealing	Cosine Annealing	MultiStep	MultiStep
	Weight decay	0.01	0.01	0.0001	0.0001
	Epoch	20	20	30	30

B.2. Baseline Details

In Table 1 of the main manuscript, we report the results of several strong baseline models: CTF-Occ (NeurIPS'23) [14], BEVFormer (ECCV'22) [10], TPVFormer (CVPR'23) [5], SurroundOcc (ICCV'23) [17], OccFormer (ICCV'23) [20] and PanoOcc (CVPR'24) [16]. However, since there are no official reports on the baseline results for `Small` and `Tiny` query settings, we select BEVFormer and PanoOcc as the main comparison targets and reproduce their results using the official code base. Also, since PanoOcc uses additional 3D object detection ground truth, we retrain the model without the detection loss for a fair comparison with our proposed ProtoOcc, which only trains with the occupancy ground truth. We additionally report the performance including the detection loss on PanoOcc in Table A.4. For SemanticKITTI, we additionally compare with MonoScene (CVPR'22) [3], HASSC (CVPR'24) [15], VoxFormer (CVPR'23) [9] and Symphonies (CVPR'24) [7]. Here, we select VoxFormer

Algorithm 1 PyTorch Pseudo Code for Linear Mapping Function $\mathcal{H}$ in Eq.2

```
# Mapping affinity matrix A in 2D grid-cell (h'x
↪  w').
# N, M, K, : number of view, number of prototype,
↪  number of hit query
# h', w':  spatial dimension

# input:
# A : affinity matrix, [N, M, K]
# q_c: each query coordinate, [N, K, 1, 2]

# output:
# prototype_aware_pixel_feature: [N, M, h', w']

def map_affinity_to_grid(A, q_c):
    # get implicit 2D grid-cell
    prototype_aware_pixel_feature =
    ↪  torch.zeros(N, M, h'* w')

    # get query coordinate
    q_c = q_c[..., 0] * w'
    q_c = q_c[..., 1] * h'
    q_c = q_c.floor()

    # filtering the value beyond the spatial
    ↪  shape
    c_mask_x = torch.logical_and(q_c[..., 0] >=
    ↪  0, q_c[..., 0] < w')
    c_mask_y = torch.logical_and(q_c[..., 1] >=
    ↪  0, q_c[..., 1] < h')
    c_mask = torch.logical_and(c_mask_x,
    ↪  c_mask_y)

    # get index and value
    c_idx = q_c[..., 1] * w' + q_c[..., 0]

    c_idx = (c_idx *
    ↪  c_mask).squeeze().unsqueeze(1).expand(N,
    ↪  M, -1)

    c_val = A *
    ↪  c_mask.squeeze().unsqueeze(1).expand(N,
    ↪  M, -1)

    # scatter similarity into the 2d gird-cell
    prototype_aware_pixel_feature.scatter_add_(2,
    ↪  c_idx, c_val)

    return
    ↪  prototype_aware_pixel_feature.reshape(N,
    ↪  M, h', w')
```

and Symphonies as our main experimental baseline and plug our modules to observe their effectiveness.

C. Method Details

C.1. Prototype-aware View Transformation

The core components of this process is prototype mapping and a contrastive learning based on pseudo ground truth masks. To achieve prototype mapping, 2D prototype feature $\mathbf{P}_{\text{img}}$ should be first obtained from the given input images. We adopt the grouping technique of [4] to generate $\mathbf{P}_{\text{img}}$, which is a widely used algorithm for clustering [1, 6, 19]. Specifically, we set the downsampling ratio r as 4, and the number of iterations for clustering as 6. Also, the contrastive learning with pseudo masks is necessary for enhancing the distinctiveness among the prototype features, thereby enriching the high-level contexts in voxel queries. In this stage, we develop a linear mapping function $\mathcal{H}$ for mapping 3D voxel prototype feature $\mathbf{P}_{\text{vox}}$ onto 2D grid-cell. To enhance clarity and facilitate understanding, we provide a pseudo-code of linear mapping function $\mathcal{H}$ in Algorithm 1. For this algorithm, we utilize an additional argument `q_c` indicating the coordinate of individual queries, which are obtained through the camera parameters during view transformation process.

C.2. Multi-perspective Occupancy Decoding

Since feature-level voxel augmentations (e.g., Random Dropout and Gaussian Noise) explicitly modify the feature values within the voxel queries, it is intuitive that they produce different contextual results when decoded. However, how do spatial-level voxel augmentations (e.g., Transpose and Flips) create different contexts without any changes in the feature values? The answer to this question can be found in the shared-weight design of the transposed convolution network.

Consider a simple 2D case with $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$ as the input and $\begin{bmatrix} w & x \\ y & z \end{bmatrix}$ as the kernel. Just as in the voxel queries, assume that each element of the input represents a specific region and is aware of its connectivity with the neighboring elements. Then, performing the 2D transposed convolution $\mathcal{C}_{2D}$ with strides as (1,1), we derive the following output:

$$\mathcal{C}_{2D}^{(1,1)}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}, \begin{bmatrix} w & x \\ y & z \end{bmatrix}\right) = \begin{bmatrix} aw & ax+bw & bx \\ ay+cw & az+by+cx+dw & bz+dx \\ cy & cz+dy & dz \end{bmatrix}.$$

The output elements are the linear combinations between the neighboring features, indicating a specific spatial context. Now we make spatial augmentations and apply $\mathcal{C}_{2D}$ with the same kernel. The result for the transposed input is formulated as:

$$\mathcal{C}_{2D}^{(1,1)}\left(\begin{bmatrix} a & c \\ b & d \end{bmatrix}, \begin{bmatrix} w & x \\ y & z \end{bmatrix}\right) = \begin{bmatrix} aw & ax+cw & cx \\ ay+bw & az+cy+bx+dw & cz+dx \\ by & bz+dy & dz \end{bmatrix}.$$

Since the transposed input changes the spatial configuration of the original input, we re-do the spatial augmentation on

the output to match the spatial locations between the augmentations:

$$\begin{bmatrix} aw & ax+cw & cx \\ ay+bw & az+cy+bx+dw & cz+dx \\ by & bz+dy & dz \end{bmatrix}$$
$$\xrightarrow{\text{Re-do}} \begin{bmatrix} aw & ay+bw & by \\ ax+cw & az+cy+bx+dw & bz+dy \\ cx & cz+dx & dz \end{bmatrix}.$$

Then, comparing this with the output from the first equation, we observe that each element, representing the feature of a specific region in the occupancy space, is made up of different linear combinations, indicating a distinct spatial context. For example, $ax+bw$ and $ay+bw$ in the identical position of each output indicates that they understand the context in that region uniquely by assigning different weights to features a and b. Similarly, the 2D transposed convolution result of flipped input yields:

$$\mathcal{C}_{2D}^{(1,1)}\left(\begin{bmatrix} d & c \\ b & a \end{bmatrix}, \begin{bmatrix} w & x \\ y & z \end{bmatrix}\right) = \begin{bmatrix} dw & dx+cw & cx \\ dy+bw & dz+cy+bx+aw & cz+ax \\ by & bz+ay & az \end{bmatrix},$$

and re-doing the same augmentation on the output produces:

$$\begin{bmatrix} dw & cw+dx & cx \\ bw+dy & aw+bx+cy+dz & ax+cz \\ by & ay+bz & az \end{bmatrix}$$
$$\xrightarrow{\text{Re-do}} \begin{bmatrix} az & ay+bz & by \\ ax+cz & aw+bx+cy+dz & bw+dy \\ cx & cw+dx & dw \end{bmatrix}.$$

Note that we apply both horizontal and vertical flips for this augmentation. As a result, this creates another totally new context in each region. The same principle holds true for 3D transposed convolution used in our network. Therefore, we can conclude that the spatial augmentation with a transposed convolution is a valid strategy for contextual diversity.

D. Additional Experimental Results

D.1. Quantitative Results

Occ3D-nuScenes. We present an additional comparative result for the `Base` and `Small`-sized voxel queries. Note that † follows the commonly suggested configuration [10, 16] for `Base`-sized voxel queries with larger image size and more training epochs. Furthermore, ‡ indicates that the model utilizes 3D bounding box ground truth. As shown in Table A.4, our method achieves an overall performance improvement across every variant († and ‡), especially surpassing the baselines in several important classes (e.g., cars and pedestrians). Notably, compared to ‡, we observe that ProtoOcc exhibits superior performance, despite the absence of explicit geometric and semantic cues derived from the bounding box ground truth of each object. This observation demonstrates that the proposed ProtoOcc method effectively captures visual geometric details from the given RGB images by leveraging high-level image prototypes and multi-perspective decoding strategies regardless of the voxel query sizes.

SemanticKITTI. We further present additional qualitative results on the validation and hidden test set with detailed scores over each class. As clearly indicated in Table A.6 and A.7, we observe that ProtoOcc shows significant improvements across various sizes of voxel query, which is consistent with the previous results in the main manuscript. Specifically, we show that ProtoOcc is also beneficial in test set, indicating the generalizability of our proposed methods.

Efficiency Comparison. The experimental comparisons of the computational efficiencies are reported in table A.5. According to the results, due to the reduced voxel size, we notice performance degradations from `Base` setting to `Small` setting. However, applying our ProtoOcc on the two baseline (VoxFormer [9] and Symphonies [7]) offsets the loss in performances, even outperforming `Base` performance in mIoU with `Small` query setting. Note that ProtoOcc achieves this performance with 83.32%, 54.58%, and 82.29% less parameters, inference time and FLOPs, respectively, for VoxFormer. For Symphonies, ProtoOcc saves 53.17%, 46.46%, and 67.90% of parameters, inference time and FLOPs, respectively.

D.2. Qualitative Results

In this section, we provide additional qualitative results of 3D occupancy prediction and 3D semantic scene completion. As clearly shown in Figure A.2 and A.3, the high-level 2D semantics enable ProtoOcc to interpret the scene in a more sectionalized manner, producing more detailed predictions over the baseline. In contrast, the baseline, which does not utilize the explicit guidance of high-level 2D contextual information, confuses the occupied states in empty space (See 2$^{\text{nd}}$ row example of Figure A.3). Furthermore, the baseline detects ghost objects, causing serious safety concerns due to misunderstanding the irregular driving surface (See 1$^{\text{st}}$ row example of Figure A.2). However, our context-diversified decoding strategy enables precise semantic understanding, by considering the variously augmented context features. Also, from A.4 and A.5, we observe that applying ProtoOcc can effectively capture the boundaries of the objects in 2D images, which affects depth predictions of the occupancies.

Table A.4. **Comparison of the quantitative results on Occ3D-nuScenes [14].** † demonstrates the variant of the `Base`, which utilizes the larger image resolution and more training duration. ‡ utilizes external ground truth, particularly bounding boxes for 3D object detection.

Query Size	Model	mIoU	others	barrier	bicycle	bus	car	const. veh.	motorcycle	pedestrian	traffic cone	trailer	truck	drive. suf.	other flat	sidewalk	terrain	manmade	vegetation
`Base`	BEVFormer [10]	34.97	7.53	41.77	16.39	44.06	48.48	17.27	20.01	23.36	21.16	28.88	35.59	80.12	35.35	47.65	51.89	40.68	34.28
	PanoOcc [16]	38.11	9.75	45.31	22.45	43.13	50.19	22.25	27.35	24.49	25.17	31.74	37.95	81.74	42.29	50.82	54.80	40.81	37.14
	BEVFormer†	38.91	9.13	48.22	22.80	47.30	54.15	21.86	27.76	27.48	26.34	32.69	38.93	81.81	39.89	50.24	53.03	43.87	35.97
	PanoOcc†	41.33	11.60	**50.74**	27.45	**48.03**	53.50	23.41	31.39	28.97	29.07	34.78	39.86	83.10	45.44	54.00	56.63	44.53	**40.08**
	PanoOcc‡	41.60	**11.99**	49.82	28.92	45.46	54.78	**25.2**	**32.93**	28.86	**30.71**	33.87	41.32	83.18	45.0	53.80	56.10	45.11	**40.1**
	ProtoOcc† (Ours)	**41.87**	11.47	49.97	**29.91**	47.75	**55.07**	23.08	32.22	**29.07**	29.38	**35.17**	**42.02**	**83.66**	**45.82**	**54.7**	**57.51**	**44.95**	40.04
`Small`	BEVFormer [10]	33.98	6.75	41.67	13.91	41.97	48.49	17.83	18.01	22.19	19.08	29.64	33.23	79.42	36.48	46.82	49.26	39.04	33.91
	PanoOcc [16]	35.78	8.18	41.60	20.79	41.25	47.78	21.87	23.42	21.03	19.29	29.71	36.10	81.20	40.00	49.22	53.94	38.09	34.83
	PanoOcc‡	36.63	8.64	**43.71**	21.69	42.55	49.91	21.32	**25.35**	**22.92**	**20.19**	29.78	37.19	80.97	40.36	49.65	52.84	39.81	35.82
	ProtoOcc (Ours)	**37.80**	**9.28**	43.64	**22.30**	**44.72**	**50.07**	**23.68**	25.23	22.77	19.66	**30.43**	**38.73**	**82.05**	**42.61**	**51.68**	**55.84**	**41.91**	**38.05**

Table A.5. Comparison of Computational Efficiency.

Query Res.	Model	Inf. Time	FLOPs (G)	Params	mIoU
128×128×16 (`Base`)	VoxFormer-B [9]	49.1 ms	288	34.61M	12.35
	Symphonies-B [7]	76.2 ms	162	21.59M	14.38
64 × 64 × 8 (`Small`)	VoxFormer-S	13.4 ms	37	5.31M	11.57
	+ ProtoOcc–S (Ours)	22.3 ms	51	5.77M	12.40
	Symphonies-S	23.5 ms	46	9.85M	13.64
	+ ProtoOcc–S (Ours)	40.8 ms	52	10.11M	14.50

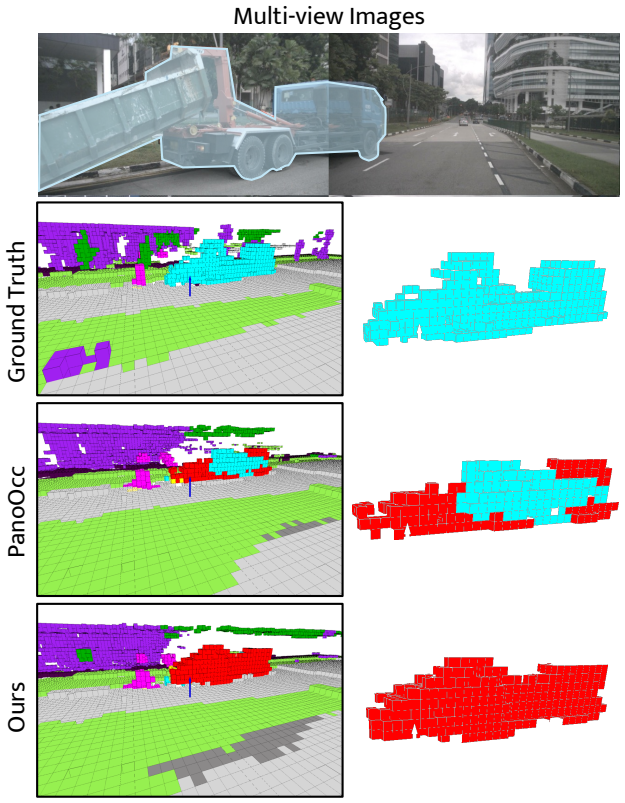


Figure A.1. **Limitation Case.**

E. Limitation and Societal Impact

As shown in Figure A.1, our proposed ProtoOcc misses the class label while accurately predicting the comprehensive geometry of the object. This phenomenon mainly occurs in the objects with rare classes, which is a long-standing class imbalance problem for the classification tasks. Moreover, the prediction of classes bounded in pre-defined labels is unsuitable for the comprehensive understanding of 3D scene. Thus, we recommend further research to extend the task to an open vocabulary setting, by incorporating the powerful external knowledge of foundation models (e.g., CLIP [12] and Grounding DINO [11]). Specifically, our prototype-aware pixel features can be aligned within the text-image embedding space to provide a fine-grained and unbounded understanding of the surrounding scene. We are open to more discussions and suggestions.

Table A.6. **Quantitative results on SemanticKITTI `val`.**

Method	IoU	mIoU	road (%)	sidewalk (%)	parking (%)	other-grnd. (%)	building (%)	car (%)	truck (%)	bicycle (%)	motorcycle (%)	other-veh. (%)	vegetation (%)	trunk (%)	terrain (%)	person (%)	bicyclist (%)	motorcyclist (%)	fence (%)	pole (%)	traf.-sign (%)
LMSCNet [13]	28.61	6.70	40.68	18.22	4.38	0.00	10.31	18.33	0.00	0.00	0.00	0.00	13.66	0.02	20.54	0.00	0.00	0.00	1.21	0.00	0.00
AICNet [8]	29.59	8.31	43.55	20.55	11.97	0.07	12.94	14.71	4.53	0.00	0.00	0.00	15.37	2.90	28.71	0.00	0.00	0.00	2.52	0.06	0.00
JS3C-Net [18]	38.98	10.31	50.49	23.74	11.94	0.07	15.03	24.65	4.41	0.00	0.00	6.15	18.11	4.33	26.86	0.67	0.27	0.20	3.94	3.77	1.45
MonoScene [3]	36.86	11.08	56.52	26.72	14.27	0.46	14.09	23.26	6.98	0.61	0.45	1.48	17.89	2.81	29.64	1.86	1.20	0.00	5.84	4.14	2.25
TPVFormer [5]	35.61	11.36	56.50	25.87	20.60	0.85	13.88	23.81	8.08	0.36	0.05	4.35	16.92	2.26	30.38	0.51	0.89	0.00	5.94	3.14	1.52
OccFormer [20]	36.50	13.46	58.85	26.88	19.61	0.31	14.40	25.09	25.53	0.81	1.19	8.52	19.63	3.93	32.62	2.78	2.82	0.00	5.61	4.26	2.86
HASSC [15]	44.82	13.48	57.05	28.25	15.90	1.04	19.05	27.23	9.91	0.92	0.86	5.61	25.48	6.15	32.94	2.80	4.71	0.00	6.58	7.68	4.05
VoxFormer-S [9]	43.10	11.51	53.71	26.04	13.18	0.09	18.91	24.39	1.80	1.14	0.13	1.46	25.22	5.71	31.93	0.67	1.14	0.00	5.80	5.51	2.75
+ ProtoOcc (Ours)	43.55	12.39	56.06	26.96	13.28	0.29	19.00	24.69	9.85	0.18	0.23	3.23	25.74	5.57	33.89	1.03	1.14	0.00	6.32	5.33	2.65
VoxFormer-B [9]	44.02	12.35	54.76	26.35	15.50	0.70	17.65	25.79	5.63	0.59	0.51	3.77	24.39	5.08	29.96	1.78	3.32	0.00	7.64	7.11	4.18
+ ProtoOcc (Ours)	**44.90**	13.57	58.11	28.83	17.52	0.54	19.72	27.43	12.14	0.24	0.16	3.21	26.42	6.54	34.31	1.26	2.89	0.00	7.87	7.43	3.31
Symphonies-S [7]	41.67	13.64	56.86	27.73	18.23	0.33	21.37	26.24	9.23	1.98	0.71	8.16	24.47	5.87	30.68	3.29	3.32	0.00	7.27	8.03	5.32
+ ProtoOcc (Ours)	43.02	14.50	57.84	28.27	17.94	0.09	22.81	27.67	15.96	1.69	1.63	11.46	26.13	5.29	32.48	3.22	2.54	0.00	7.70	7.54	5.23
Symphonies-B [7]	41.85	14.38	56.52	26.29	18.08	0.02	22.22	29.19	10.75	2.86	2.87	10.60	25.62	6.53	30.29	3.92	3.13	0.00	8.12	9.85	6.09
+ ProtoOcc (Ours)	42.12	**14.83**	56.95	27.46	18.29	0.02	22.07	29.32	14.46	3.24	2.54	13.66	25.87	6.19	30.76	3.79	3.11	0.00	8.93	9.20	5.76

Table A.7. **Quantitative results on SemanticKITTI `test`.**

Method	IoU	mIoU	road (%)	sidewalk (%)	parking (%)	other-grnd. (%)	building (%)	car (%)	truck (%)	bicycle (%)	motorcycle (%)	other-veh. (%)	vegetation (%)	trunk (%)	terrain (%)	person (%)	bicyclist (%)	motorcyclist (%)	fence (%)	pole (%)	traf.-sign (%)
LMSCNet [13]	31.38	7.07	46.70	19.50	13.50	3.10	10.30	14.30	0.30	0.00	0.00	0.00	10.80	0.00	10.40	0.00	0.00	0.00	5.40	0.00	0.00
AICNet [8]	23.93	7.09	39.30	18.30	19.80	1.60	9.60	15.30	0.70	0.00	0.00	0.00	9.60	1.90	13.50	0.00	0.00	0.00	5.00	0.10	0.00
JS3C-Net [18]	34.00	8.97	47.30	21.70	19.90	2.80	12.70	20.10	0.80	0.00	0.00	4.10	14.20	3.10	12.40	0.00	0.20	0.20	8.70	1.90	0.30
MonoScene [3]	34.16	11.08	54.70	27.10	24.80	5.70	14.40	18.80	3.30	0.50	0.70	4.40	14.90	2.40	19.50	1.00	1.40	0.40	11.10	3.30	2.10
TPVFormer [5]	34.25	11.26	55.10	27.20	27.40	6.50	14.80	19.20	3.70	1.00	0.50	2.30	13.90	2.60	20.40	1.10	2.40	0.30	11.00	2.90	1.50
OccFormer [20]	34.53	12.32	55.90	30.30	31.50	6.50	15.70	21.60	1.20	1.50	1.70	3.20	16.80	3.90	21.30	2.20	1.10	0.20	11.90	3.80	3.70
HASSC [15]	43.40	13.34	54.60	27.70	23.80	6.20	21.10	22.80	4.70	1.60	1.00	3.90	23.80	8.50	23.30	1.60	4.00	0.30	13.01	5.80	5.50
VoxFormer-S [9]	42.56	11.28	51.30	24.80	18.50	2.70	21.40	19.7	1.00	0.40	0.20	1.10	24.70	5.90	21.10	0.70	0.60	0.20	11.60	4.00	4.30
+ ProtoOcc (Ours)	42.20	11.60	54.50	26.30	19.40	3.90	20.80	19.80	1.70	0.80	0.30	1.80	23.80	6.20	21.30	0.70	0.50	0.10	10.20	4.00	4.30
VoxFormer-B [9]	42.95	12.20	53.90	25.30	21.10	5.60	19.80	20.80	3.50	1.00	0.70	3.70	22.40	7.50	21.30	1.40	2.60	0.20	11.10	5.10	4.90
+ ProtoOcc (Ours)	**43.87**	12.81	57.10	28.20	22.80	4.30	21.70	22.40	2.70	0.80	0.40	1.60	24.90	7.90	22.60	0.90	0.90	0.30	12.30	5.80	5.70
Symphonies-S [7]	42.04	14.07	57.60	29.40	29.00	10.50	24.20	22.00	2.70	2.00	1.80	4.70	23.80	7.50	22.80	2.10	1.40	0.70	13.80	5.60	5.70
+ ProtoOcc (Ours)	42.21	14.36	57.20	30.40	30.00	10.10	24.50	22.30	2.80	2.00	1.70	4.30	24.50	8.10	23.20	2.30	2.20	0.40	14.60	6.20	6.30
Symphonies-B [7]	41.82	14.50	57.00	28.00	27.50	9.10	24.10	23.70	3.70	3.60	2.40	4.40	24.60	9.90	22.50	2.90	1.70	0.50	15.70	7.50	7.00
+ ProtoOcc (Ours)	42.17	**14.77**	57.90	28.60	27.60	10.20	24.30	24.10	3.10	3.30	3.20	4.70	25.00	9.20	22.60	3.40	2.00	1.50	15.60	7.30	6.90

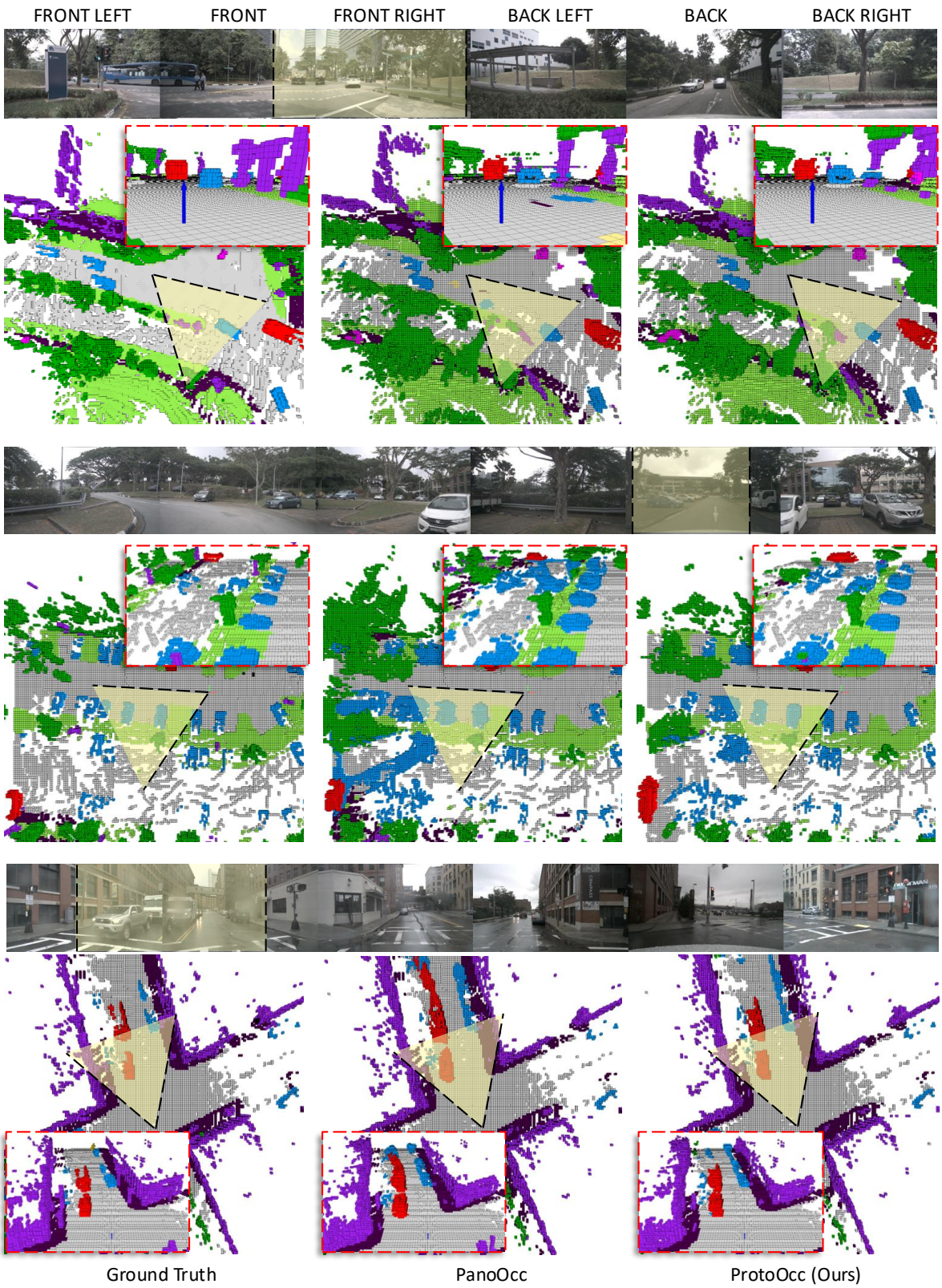


Figure A.2. **Predictions in challenging scenarios.** We visualize the prediction results in both aerial and zoomed views, where the zoomed view point is highlighted in yellow. Best viewed in color.

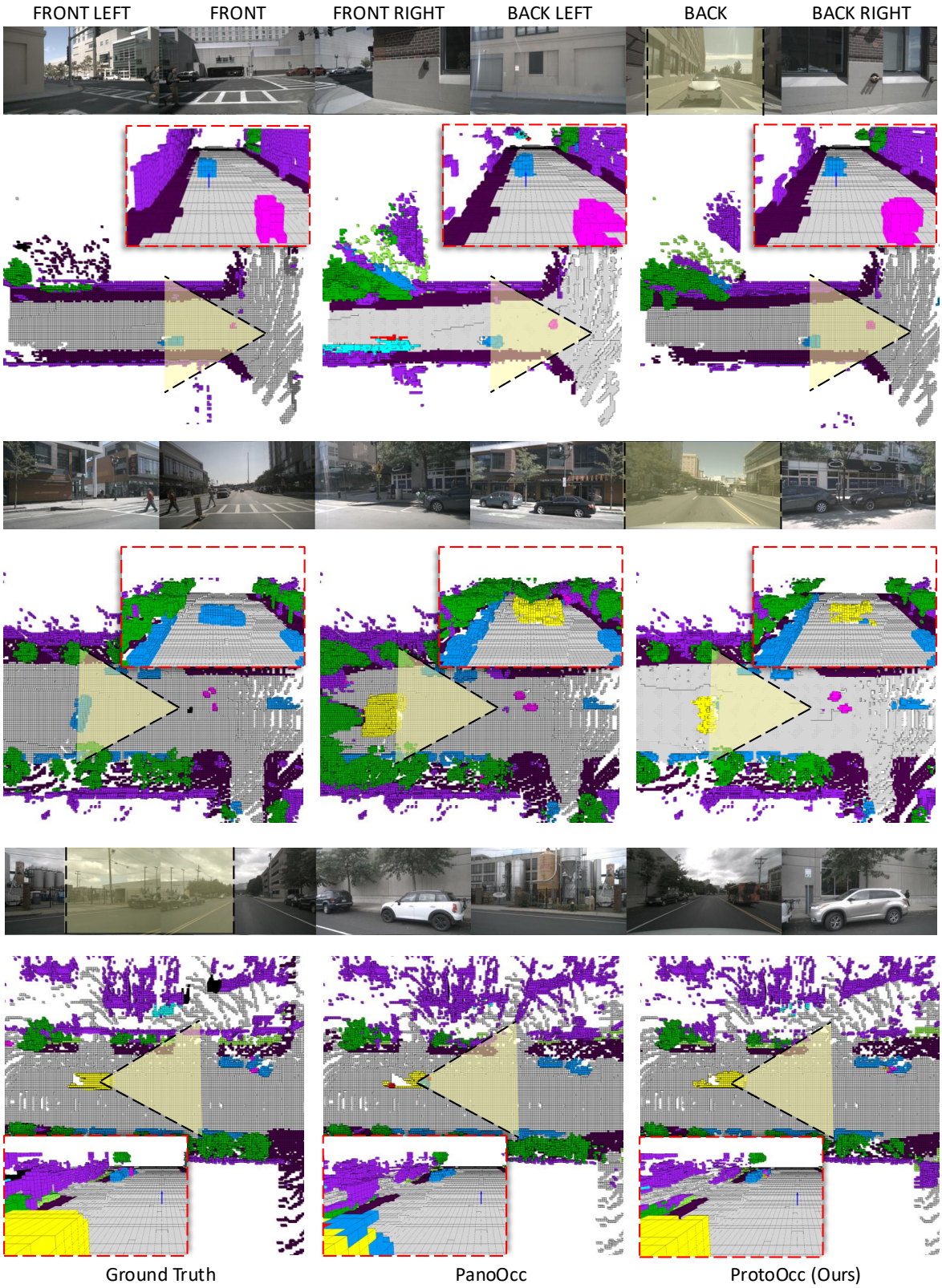
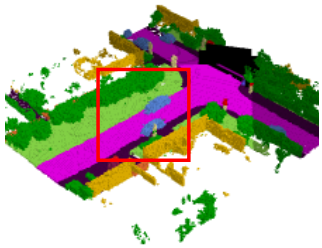
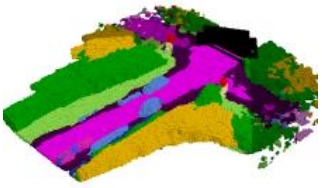


Figure A.3. **Predictions in challenging scenarios.** We visualize the prediction results in both aerial and zoomed views, where the zoomed view point is highlighted in yellow. Best viewed in color.

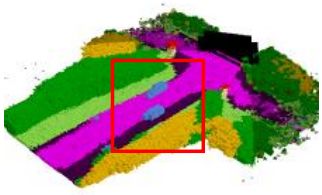
RGB Image



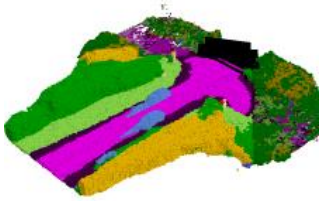
Ground Truth



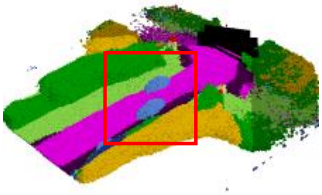
VoxFormer-S



+ ProtoOcc (Ours)

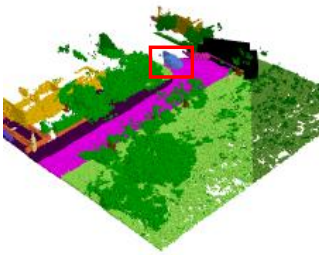


VoxFormer-B



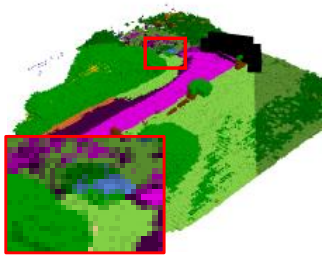
+ ProtoOcc (Ours)

RGB Image

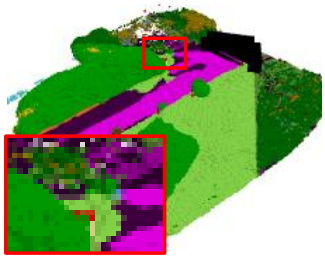


Ground Truth

VoxFormer-S



+ ProtoOcc (Ours)



VoxFormer-B

+ ProtoOcc (Ours)

Figure A.4. **Qualitative results on SemanticKITTI.**

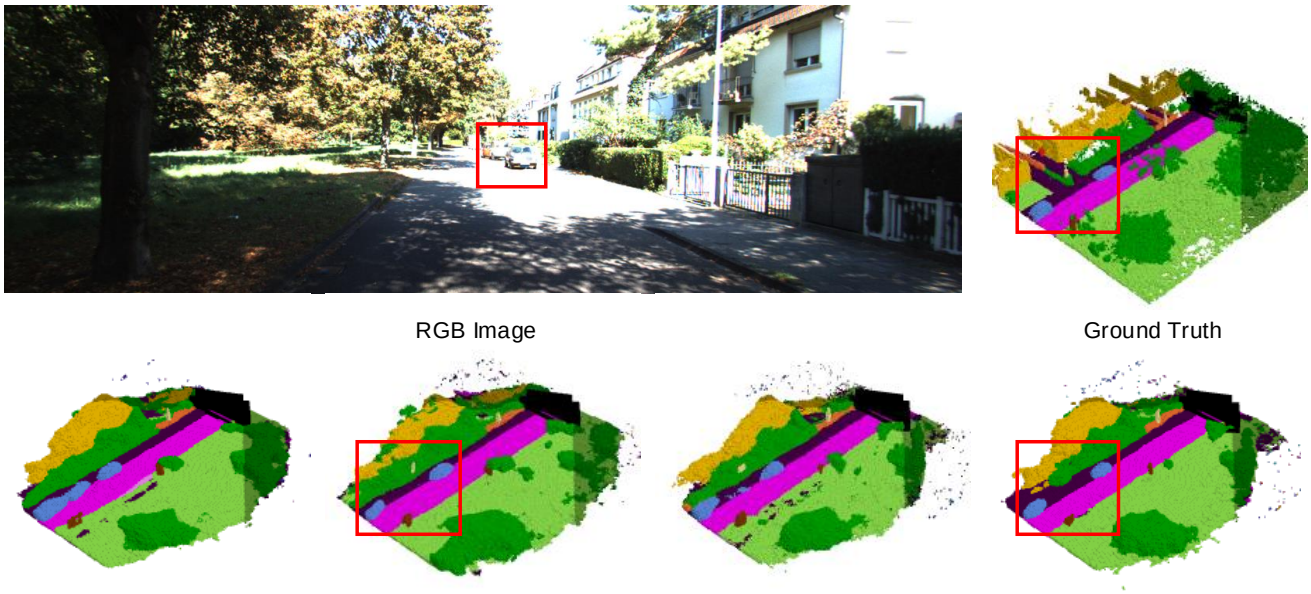


Figure A.5. **Qualitative results on SemanticKITTI.**

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11): 2274–2282, 2012. 2

[2] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019. 1

[3] 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. 1, 5

[4] Jian Ding, Nan Xue, Gui-Song Xia, Bernt Schiele, and Dengxin Dai. Hgformer: Hierarchical grouping transformer for domain generalized semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15413–15423, 2023. 2

[5] 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, 5

[6] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018. 2

[7] Haoyi Jiang, Tianheng Cheng, Naiyu Gao, Haoyang Zhang, Tianwei Lin, Wenyu Liu, and Xinggang Wang. Symphonize 3d semantic scene completion with contextual instance queries. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20258–20267, 2024. 1, 3, 4, 5

[8] Jie Li, Kai Han, Peng Wang, Yu Liu, and Xia Yuan. Anisotropic convolutional networks for 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3351–3359, 2020. 5

[9] 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, 3, 4, 5

[10] 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. 1, 3, 4

[11] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4

[12] 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. 4

[13] Luis Roldao, Raoul de Charette, and Anne Verroust-Blondet. Lmscnet: Lightweight multiscale 3d semantic completion. In *2020 International Conference on 3D Vision (3DV)*, pages 111–119. IEEE, 2020. 5

[14] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 4

[15] Song Wang, Jiawei Yu, Wentong Li, Wenyu Liu, Xiaolu Liu, Junbo Chen, and Jianke Zhu. Not all voxels are equal: Hardness-aware semantic scene completion with self-distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14792–14801, 2024. 1, 5

[16] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. *arXiv preprint arXiv:2306.10013*, 2023. 1, 3, 4

[17] Yi Wei, Linqing Zhao, Wenzhao Zheng, Zheng Zhu, Jie Zhou, and Jiwen Lu. Surroundocc: Multi-camera 3d occupancy prediction for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21729–21740, 2023. 1

[18] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3101–3109, 2021. 5

[19] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13964–13973, 2020. 2

[20] Yunpeng Zhang, Zheng Zhu, and Dalong Du. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9433–9443, 2023. 1, 5