

Supplementary Material

Our supplementary material includes dataset details, implementation details, evaluation metrics, and additional visualization results. We also provide a demo video of our work. To enable our work to contribute to the community, we promise to open-source codes and checkpoints upon acceptance.

A Dataset Details

The SemanticKITTI [1] dataset is a large-scale dataset based on the KITTI Vision Benchmark [2], which includes automotive LiDAR scans voxelized into $256 \times 256 \times 32$ grids with 0.2m resolution for 22 sequences. The dataset contains 28 classes with non-moving and moving objects. In our experiments, we only use RGB images captured by cam2 as input from the KITTI odometry benchmark and follow the official split of the dataset, with sequence 08 for validation and other sequences from 0 to 10 for training.

The KITTI-MOT [2] dataset is based on the KITTI Multi-Object Tracking Evaluation 2012, which consists of 21 training sequences and 29 test sequences. It holds stereo images from two forward-facing cameras and LiDAR point clouds. We selected all the training sequences except sequences where the ego-car is stationary or contains only a small number of moving objects(sequence 0012, 0013, and 0017) for training, and select testing sequences 0000-0014 for validation. We also apply LiDAR data and pseudo-optical flow to coarsely select dynamic frames for a higher sampling rate in our training process.

The nuScenes [3] dataset consists of 1000 sequences of various driving scenes of 20 seconds duration each and the keyframes are annotated at 2Hz. Each sample includes RGB images collected by 6 surround cameras with 360° horizontal FOV and LiDAR point clouds from 32-beam LiDAR sensors. The 1000 scenes are officially divided into training, validation, and test splits with 700, 150, and 150 scenes, respectively.

B Implementation Details

B.1 Model Architecture

We adopted ConvNeXt-Base with SparK [4] pretraining as the 2D image backbone and a four-level FPN [5] to extract image features. We utilize TPV [6] uniformly divided to represent a cuboid area for multi-view feature integration, i.e. [80, 80, 6.4] meters around the ego car for nuScenes [3] and [51.2, 51.2, 6.4] meters in front of the ego car for SemanticKITTI [1] and KITTI-MOT [2]. Considering the computational cost of the subsequent temporal fusion module, we use half of the output occupancy resolution for TPV grid cell, i.e. 0.8 meters for nuScenes and 0.4 meters for SemanticKITTI and KITTI-MOT, respectively.

For the input temporal sequence, we utilize 2 previous frames for our temporal fusion module. Our 3D refine module includes a three-layer residual 3D convolution block and a 3D FPN block for spatial feature integration. To upsample the volume into the output occupancy resolution, we use a deconvolution module as FBOCC [7]. We adopt separate two-layer MLPs $\{\Theta_s, \Theta_f\}$ as decoders to construct volumetric fields for SDF and flow.

B.2 2D Flow Estimation

For occupancy flow prediction in KITTI-MOT [2] dataset, we leverage the flow estimation model of Unimatch [8] trained on FlyingChairs [9], FlyingThings3D [10] and fine-tuned on KITTI [2] to predict optical flow maps for supervision directly. Note that the Unimatch model is trained in a supervised manner, we provide an unsupervised flow fine-tuning strategy following [11] when adapted to different driving scenes. Specifically, we fine-tune the Unimatch model utilizing unsupervised flow

techniques including stopping the gradient at the occlusion mask, encouraging smoothness before upsampling the flow field, and continual self-supervision with image resizing.

As for the nuScenes [3] dataset, the 3D occupancy flow ground-truth data is only available at 2Hz keyframes so it is challenging for optical flow estimation. Thus we employed a tracking-based flow estimation strategy using CoTrackerV2 [12]. We aim to obtain accurate optical flow between two keyframes by including the non-keyframe sequences to capture the long-term track. Specifically, we first use open-vocabulary 2D segmentor GroundedSAM[13] to predict dynamic foreground semantic segmentation. Then we take the adjacent keyframes and the non-keyframe sequences between the two keyframes as video input and conduct dense track to capture the long-term motion dependency in the masked regions. Once we have the initial pixel coordinates in the first keyframe and the corresponding pixel coordinates in the next, we subtract the two and get the final optical flow map in the masked regions. We take this optical flow map with the dynamic foreground mask as pseudo-flow labels for occupancy flow prediction.

B.3 Training Settings

The resolution of the input image is 512x1408 for nuScenes [3], 352x1216 for SemanticKITTI [1], and 352x1216 for KITTI-MOT [2]. For loss weight, we set $\lambda_{flow} = 5 \times 10^{-3}$, $\lambda_{eik} = 0.1$, $\lambda_{dreg} = 0.1$, if present, and the weights for the edge λ_{edge} and the LiDAR λ_{lidar} losses are 0.02 and 0.2 respectively. During training, we adopt the AdamW optimizer with an initial learning rate 1e-4 and weight decay of 0.01. We use the multi-step learning rate scheduler with linear warming up in the first 1k iterations. We train our models with a total batch size of 8 on 8 A100 GPUs for 16 epochs. Experiments on SemanticKITTI [1] and KITTI-MOT [2] take less than one day, while experiments on nuScenes [3] finish within two days.

B.4 Temporal Fusion Module

Algorithm 1 provides the pseudo-code of our proposed temporal fusion module. In the BEV fusion process, we employ a Backward Forward Attention Module (BFAM) with deformable attention (DAttn) to fuse the BEV features from two adjacent frames. To illustrate our algorithmic process clearly, we visually demonstrate our temporal fusion module in the demo video.

Algorithm 1 Pseudo-code for Temporal Fusion Module

```

1: Input: A temporal sequence of Voxel features  $\{V_{-(n-1)}, \dots, V_{-1}, V_0\}$ .  $V_0$  represents Voxel
   feature of the current frame and  $V_{-i}$  corresponds to the  $i$ -th frame before  $V_0$ . A sequence of
   transformation matrix of ego coordinates  $\{T_{-(n-1)}, \dots, T_{-1}, T_0\}$ .
2: for  $i = 0$  to  $n - 1$  do
3:    $\hat{V}_{-i} \leftarrow \text{EgoMotionAlignment}(V_{-i}, T_{-i}, T_0)$ 
4:    $B_{-i}^g \leftarrow \text{MeanPooling}(\hat{V}_{-i})$ 
5:    $B_{-i} \leftarrow \text{Concatenate}(\text{VolumeToSlices}(\hat{V}_{-i}), B_{-i}^g)$  ▷ temporal volumes to BEV slices
6: end for
7: for  $i = 0$  to  $n - 2$  do
8:    $B_{-(i+1)} \leftarrow B_{-(i+1)} + \beta \cdot \text{DAttn}(B_{-(i+1)}, \{B_{-i}, B_{-(i+1)}\})$  ▷ backward process
9: end for
10: for  $i = n - 2$  to  $0$  do
11:    $B_{-i} \leftarrow B_{-i} + \beta \cdot \text{DAttn}(B_{-i}, \{B_{-(i+1)}, B_{-i}\})$  ▷ forward process
12: end for
13:  $\tilde{V}, \tilde{B}_g \leftarrow \text{SlicesToVolume}(B_0)$ 
14:  $\tilde{V}' \leftarrow \text{BEVVolumeFusion}(\tilde{V}, \tilde{B}_g)$ 
15: return  $\tilde{V}'$ 

```

B.5 Differentiable Rendering

In the rendering stage, we conduct a uniform sampling of N points $P = \{p_i | i = 1, \dots, N\}$ along the ray and apply a tri-linear interpolation operation to efficiently compute the SDF values for each point

74 from the volumetric SDF field [14]. Furthermore, the unbiased rendering weights can be calculated
 75 by $w_i = \alpha_i \prod_{j=1}^{i-1} (1 - \alpha_j)$, with α_i denoting the opacity value proposed by NeuS [15]:

$$\alpha_i = \max \left(\frac{\Phi(s_i) - \Phi(s_{i+1})}{\Phi(s_i)}, 0 \right) \quad (1)$$

76 where $\Phi(x)$ is sigmoid function $\Phi(x) = (1 + e^{-\xi x})^{-1}$ with a temperature coefficient ξ .

77 Let d_i and f_i denote the depth and flow of the i -th point, we can calculate the rendered depth d and
 78 flow f of the ray by:

$$d = \sum_{i=1}^N w_i d_i, \quad f = \sum_{i=1}^N w_i f_i \quad (2)$$

79 C Evaluation Metrics

80 C.1 Depth Estimation Evaluation Metrics

81 Following [16, 17, 18], we use evaluation metrics for self-supervised depth estimation as follows:

$$\text{Abs Rel: } \frac{1}{|T|} \sum_{d \in T} |d - d^*| / d^*, \quad \text{Sq Rel: } \frac{1}{|T|} \sum_{d \in T} |d - d^*|^2 / d^* \quad (3)$$

$$\text{RMSE: } \sqrt{\frac{1}{|T|} \sum_{d \in T} |d - d^*|^2}, \quad \text{RMSE log: } \sqrt{\frac{1}{|T|} \sum_{d \in T} |\log d - \log d^*|^2} \quad (4)$$

82 where d and d^* indicate predicted and ground truth depths respectively, and T indicates all pixels
 83 on the depth map. We calculate metrics for depth values in the range of $[0.1, 80]$ meters using 1:2
 84 resolution against the raw image.

85 C.2 3D Occupancy Prediction Evaluation Metrics

86 Previous approaches [19, 20, 21] utilize the intersection over union (IoU) as the evaluation metrics
 87 of 3D occupancy prediction on SemanticKITTI [1] dataset. However, according to our experiment
 88 results, the penalty is overly strict in evaluating the reconstruction quality effectively. As illustrated
 89 in Figure 1, we observed that rendering-based methods [19, 20] often generate true positive (TP)
 90 predictions (marked in green in the figure) distributed on the ground or in invisible regions below
 91 the ground. This prevents effective evaluation of reconstruction details, as a deviation of one voxel
 92 will lead to an IoU of zero. Furthermore, rendering-based methods tend to predict a thick surface
 93 due to the absence of supervision in invisible regions, causing a large number of false positive (FP)
 94 predictions (marked in red in the figure) and a significant reduction of the precision metric.

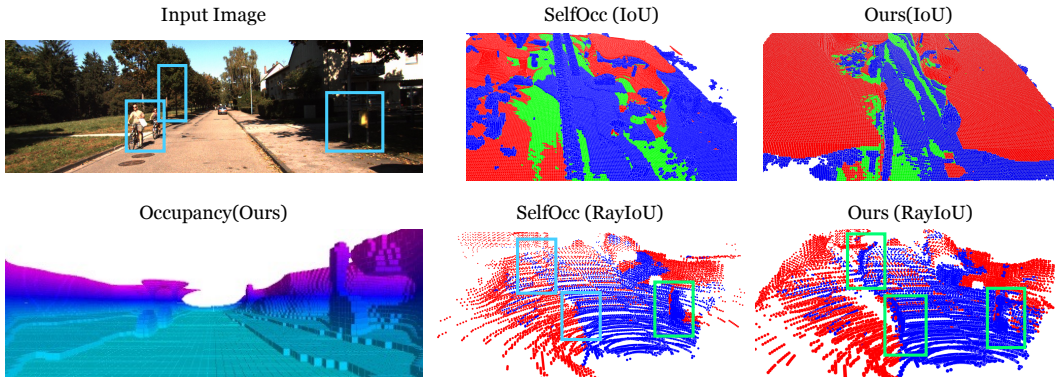


Figure 1: The comparison of IoU and RayIoU measurement results in 3D occupancy prediction.

To demonstrate the advanced occupancy prediction quality of our approach, we employ Ray-based IoU (**RayIoU**) [22] as our evaluation metric and re-evaluate other comparative methods using their open-source checkpoint models. Initially, we generated LiDAR rays with origins sampled from the sequence, and then compute the travel distances of each LiDAR ray before intersecting with the occupied voxels. A query ray is classified as true positive (TP) when the L1 error between the predicted depth and ground-truth depth is less than a pre-defined threshold (1m, 2m, and 4m).

As is shown in Figure 1, the RayIoU metric reflected the superiority of our method on the detailed prediction of the tree and bicycles compared to SelfOcc.

C.3 Occupancy Flow Prediction Evaluation Metrics

On the KITTI-MOT [2] data, due to the absence of ground-truth occupancy flow labels, we conduct the evaluation following the scene flow benchmark of KITTI dataset [2]. Specifically, we use LiDAR-projected depth and pseudo 2D optical flow cues to generate disparity and optical flow labels. And we computed end-point error of rendered and ground-truth disparity (DE) and optical flow (EPE). D1_5% represent the percentage of disparity outliers with end-point error smaller than 4 pixels or 5% of the ground-truth disparity. F1_10% is the percentage of optical flow outliers with end-point error smaller than 8 pixels or 10% of the ground-truth flow labels. And SF_10% is the percentage of scene flow outliers (outliers in either D1_10% or F1_10%). To better evaluate the scene flow in foreground regions, we also reported the foreground disparity error (DE_FG) and optical flow error (EPE_FG) by leveraging the semantic mask obtained from GroundedSAM [13].

On the nuScenes [3] dataset, since we have the ground-truth occupancy flow labels, we use Ray-based IoU (**RayIoU**), and absolute velocity error (**mAVE**) for occupancy flow, following the evaluation metrics of *Occupancy and Flow track for CVPR 2024 Autonomous Grand Challenge*.

For **RayIoU** measurement, we perform evaluation on the OpenOcc benchmark [23] as introduced in subsection C.2. We use **mAVE** to indicate the velocity errors for a set of true positives (TP) with a threshold of 2m distance. The absolute velocity error (AVE) is defined for 8 classes ('car', 'truck', 'trailer', 'bus', 'construction_vehicle', 'bicycle', 'motorcycle', 'pedestrian') in m/s.

D Visualization Results

D.1 Depth Estimation

Figure 2 shows the qualitative comparison of depth estimation on the SemanticKITTI [1] validation set. The visualization results illustrate the successful prediction of detailed and accurate depth by our method compared to other rendering-based approaches.

D.2 3D Occupancy Prediction

Figure 3 shows the qualitative comparison of 3D occupancy prediction on the SemanticKITTI [1] validation set. Our method achieves precise occupancy prediction for thin structures such as poles, trees, and cyclists. Also, it provides smooth predictions for cars and road surfaces compared to other supervised [21] and self-supervised [20, 24, 19] methods.

We provided more visualization results of depth estimation, novel view depth synthesis, and 3D occupancy prediction to illustrate the superiority of our method. As is shown in Figure 4, our method exhibited strong performance across these three tasks.

D.3 Occupancy Flow Prediction

Figure 5 and Figure 6 shows the visualization results of the depth estimation and occupancy flow prediction tasks on the KITTI-MOT [2] and nuScenes [3] validation set. Our proposed method can simultaneously provide accurate 3D occupancy and occupancy flow prediction.

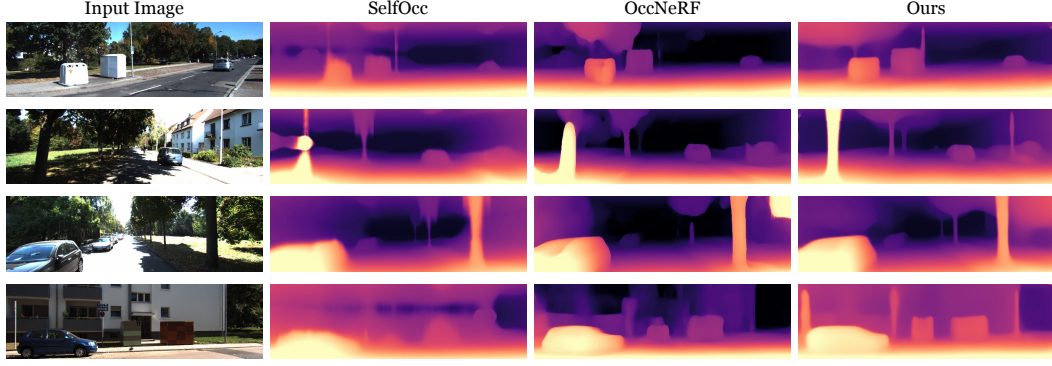


Figure 2: **Qualitative comparison for self-supervised depth estimation with other baselines on the SemanticKITTI [1] validation set.**

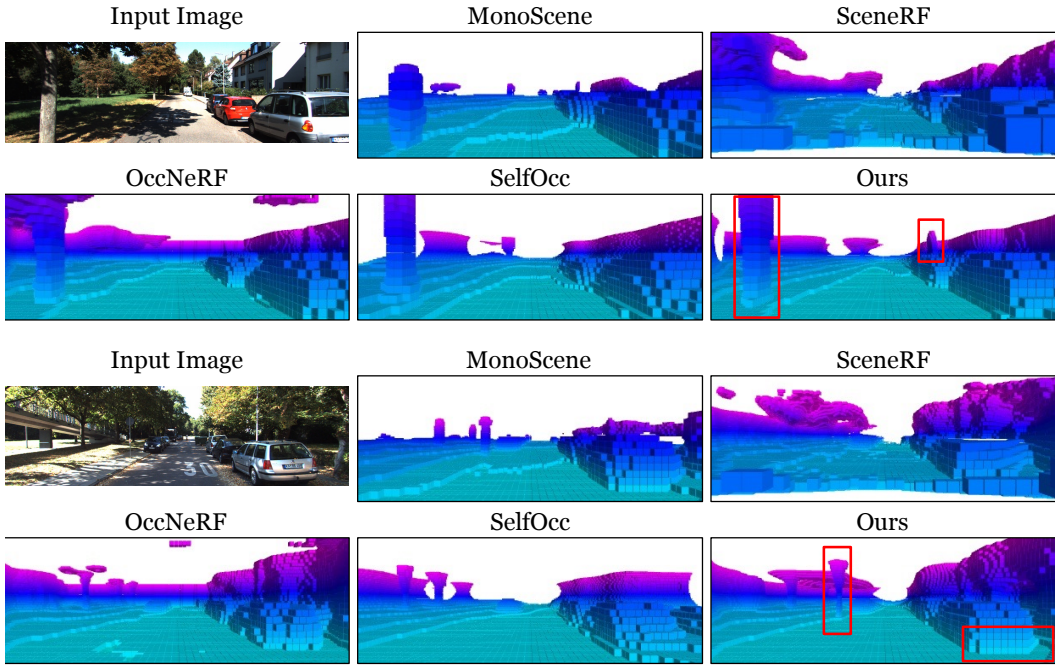


Figure 3: **Qualitative comparison for 3D occupancy prediction on the SemanticKITTI [1] validation set.** The red bounding box shows the most noticeable part.

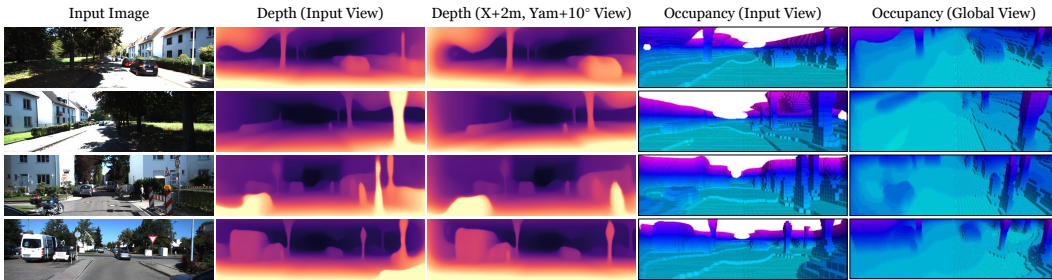


Figure 4: **Visualizations of depth estimation, novel view depth synthesis, 3D occupancy prediction on the SemanticKITTI [1] validation set.** (X+2m) means moving +2 meters along the x-axis of the LiDAR coordinate, and (Yaw+10°) means turning left for 10°.

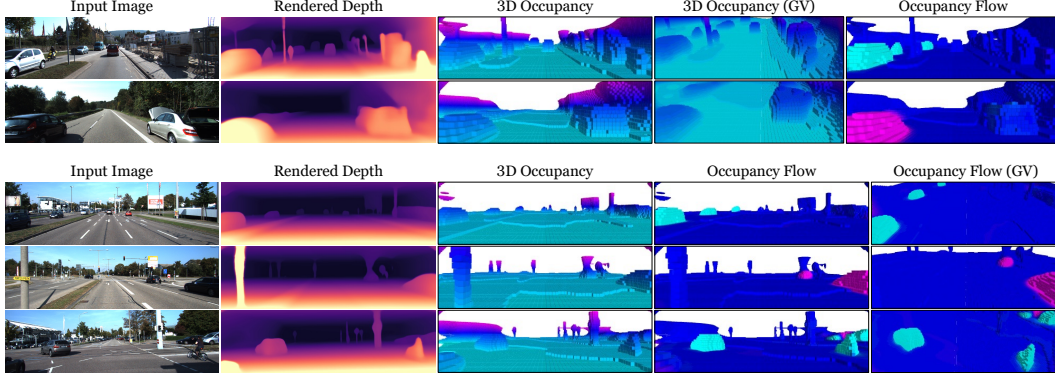


Figure 5: Visualizations of depth estimation, 3D occupancy, and occupancy flow prediction on the KITTI-MOT [2] validation set. GV indicates the global view.

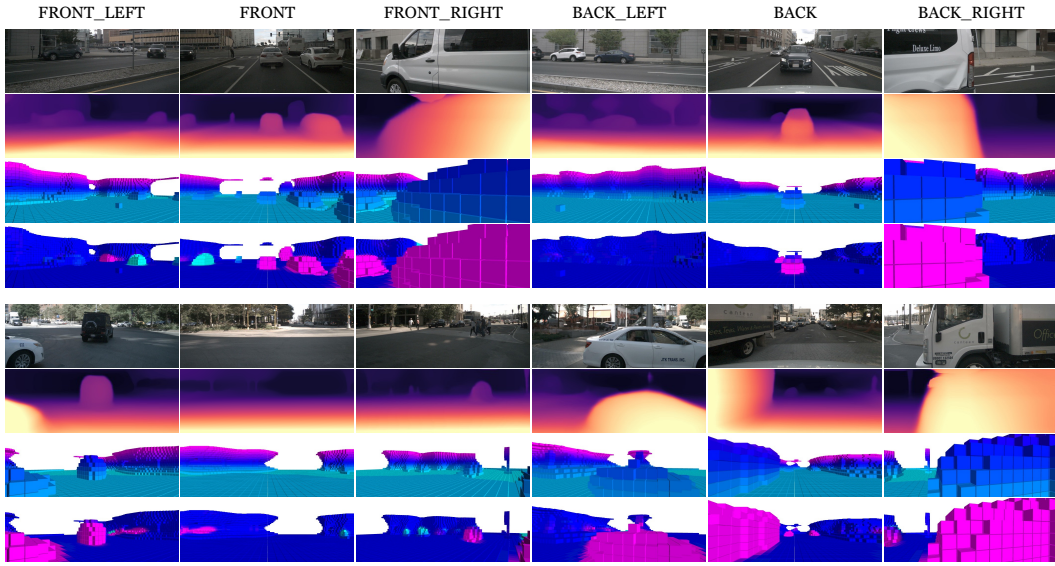


Figure 6: Visualizations of depth estimation, 3D occupancy, and occupancy flow prediction on the nuScenes [3] validation set. We show the six input surrounding images in the first row and the estimated depth from the corresponding views in the second row. The third and fourth rows demonstrate the predicted 3D occupancy and occupancy flow results separately.

138 E Limitations and Future Work

139 Although we use temporal sequence input to better exploit the historical information, our model
 140 cannot completely handle the occlusion problem due to the inherent rendering-based limitation.
 141 Subsequent research could investigate long-term occupancy flow modeling and solutions to leverage
 142 the temporal sequence supervision to scale up the visible range of perspective. In addition, although
 143 our Let Occ Flow can offer accurate 3D occupancy and occupancy flow prediction, the prediction
 144 quality highly relies on the quality of optical flow cues, which serves as the supervision for the
 145 flow field. To improve the quality of 2D optical flow labels, we proposed a tracking-based flow
 146 estimation method and an unsupervised flow fine-tuning strategy. We hope the future work can pay
 147 more attention on the improvement of the flow supervision quality. Finally, our occupancy flow
 148 prediction does not explicitly enforce consistency within instances, and future work may explore to
 149 integrate instance perception into occupancy flow prediction.

References

- [1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9297–9307, 2019.
- [2] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.
- [3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.
- [4] K. Tian, Y. Jiang, Q. Diao, C. Lin, L. Wang, and Z. Yuan. Designing bert for convolutional networks: Sparse and hierarchical masked modeling. *arXiv preprint arXiv:2301.03580*, 2023.
- [5] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection, 2017.
- [6] Y. Huang, W. Zheng, Y. Zhang, J. Zhou, and J. Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2302.07817*, 2023.
- [7] Z. Li, Z. Yu, D. Austin, M. Fang, S. Lan, J. Kautz, and J. M. Alvarez. FB-OCC: 3D occupancy prediction based on forward-backward view transformation. *arXiv:2307.01492*, 2023.
- [8] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [9] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.
- [10] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.
- [11] R. Jonschkowski, A. Stone, J. T. Barron, A. Gordon, K. Konolige, and A. Angelova. What matters in unsupervised optical flow. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 557–572. Springer, 2020.
- [12] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. Cotracker: It is better to track together. *arXiv:2307.07635*, 2023.
- [13] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.
- [14] C. Sun, M. Sun, and H. Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022.
- [15] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689*, 2021.
- [16] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

- 195 [17] Y. Wei, L. Zhao, W. Zheng, Z. Zhu, Y. Rao, G. Huang, J. Lu, and J. Zhou. Surrounddepth: En-
196 tangling surrounding views for self-supervised multi-camera depth estimation. In *Conference*
197 *on Robot Learning*, pages 539–549. PMLR, 2023.
- 198 [18] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe. Unsupervised learning of depth and ego-
199 motion from video. In *Proceedings of the IEEE conference on computer vision and pattern*
200 *recognition*, pages 1851–1858, 2017.
- 201 [19] Y. Huang, W. Zheng, B. Zhang, J. Zhou, and J. Lu. Selfocc: Self-supervised vision-based 3d
202 occupancy prediction. *arXiv preprint arXiv:2311.12754*, 2023.
- 203 [20] A.-Q. Cao and R. de Charette. Scenerf: Self-supervised monocular 3d scene reconstruction
204 with radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer*
205 *Vision*, pages 9387–9398, 2023.
- 206 [21] A.-Q. Cao and R. De Charette. Monoscene: Monocular 3d semantic scene completion. In
207 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages
208 3991–4001, 2022.
- 209 [22] P. Tang, Z. Wang, G. Wang, J. Zheng, X. Ren, B. Feng, and C. Ma. Sparseocc: Rethinking
210 sparse latent representation for vision-based semantic occupancy prediction. *arXiv preprint*
211 *arXiv:2404.09502*, 2024.
- 212 [23] W. Tong, C. Sima, T. Wang, L. Chen, S. Wu, H. Deng, Y. Gu, L. Lu, P. Luo, D. Lin, et al.
213 Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer*
214 *Vision*, pages 8406–8415, 2023.
- 215 [24] C. Zhang, J. Yan, Y. Wei, J. Li, L. Liu, Y. Tang, Y. Duan, and J. Lu. Occnerf: Advancing 3d
216 occupancy prediction in lidar-free environments. *arXiv preprint arXiv:2312.09243*, 2023.