OccMamba: Semantic Occupancy Prediction with State Space Models

Supplementary Material

6. Appendix section

6.1. More training details

In our experiment, we utilize the AdamW optimizer with a base learning rate of $5e{-4}$ and a weight decay of 0.01 to ensure effective optimization while maintaining regularization. As to the image backbone ResNet-50, which is pre-trained by torchvision, we scale its learning rate by a factor of 0.1. The learning rate schedule follows a Cosine Annealing policy, combined with a linear warmup over the first 500 iterations, starting from one-third of the base learning rate, and gradually decreasing to a minimum ratio of $1e{-3}$. For training, we set the number of epochs to 20. This experimental setup reflects our focus on balancing optimization efficiency, model stability, and rigorous evaluation to achieve reliable and reproducible results.

6.2. More reordering schemes

In addition to the Hilbert curve, other space-filling curves, such as the Z-order, are also widely used. Therefore, we conduct comparative experiments, following the procedures outlined in Sec. 4.1 and Sec. 4.3 on the OpenOccupancy validation set with v0.0 annotations to evaluate the performance of the Z-order. As presented in Table 8, it is evident that our OccMamba-128 with height-prioritized 2D Hilbert expansion outperforms the Z-curve variant in semantic occupancy prediction. In theory, the Hilbert curve effectively preserves the spatial proximity when mapped to a 1D sequences due to its recursive, space-filling path. In contrast, the Z-order curve employs a simple interleaving of bits, which makes it more likely that adjacent points are separated by greater distances in the 1D sequences. Consequently, the Hilbert curve generally offers superior locality preservation in multiple dimensions.

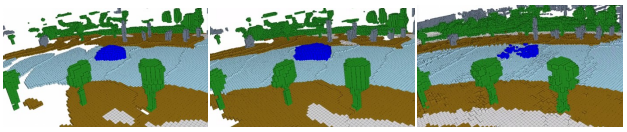
Reordering Schemes	mIoU
3D Z-order	24.6
3D Hilbert	24.8
Height-prioritized 2D Z-order expansion	25.0
Height-prioritized 2D Hilbert expansion	**25.2**

Table 8. Performance on more reordering schemes.

6.3. More ablations on local context processor (LCP)

Metric Specificity. OpenOccupancy's mIoU does not use techniques like visual masks, causing ambiguous evaluation of occluded regions. In Table. 9, LCP improves IoU (denser occupancy) and RayIoU (from SparseOcc [39], surface accuracy, excluding occlusions) by **1.0%** and **0.6%** in v0.0 labels, respectively, validating its effectiveness in refining geometric coherence (Fig. 5(a,b)).

Label Quality Impact. The old OpenOccupancy labels (v0.0), derived from static LiDAR, suffer from incomplete annotations for dynamic objects (Fig. 5(c)) and occluded areas LiDAR never seen. By using new labels (v0.1), our LCP improves mIoU by **0.4%**(Table. 9), demonstrating more gains as label noise reduces.



(a) w/o LCP (b) w LCP (c) label

Figure 5. Reconstruction results of distant occupancy (about 40m).

Method	Label	IoU	mIoU	RayIoU@0.2m
w/o LCP	v0.0	33.7	25.0	24.2
w LCP	v0.0	34.7	25.2	24.7
w/o LCP	v0.1	34.2	25.8	26.5
w LCP	v0.1	34.9	26.2	27.0

Table 9. More results of OccMamba-128.

6.4. Ablation on training loss

We conduct an ablation study on the training objectives mentioned in Sec. 3.4. Specifically, we carry out experiments on the OpenOccupancy dataset, following the procedures outlined in Sec. 4.1. To facilitate training, we use only 20% of the training set, with the model configured as OccMamba-128. The results, as shown in Table 10, indicate that all the training objectives contribute significantly to the ultimate performance. In particular, the inclusion of $\mathcal{L}_{\text{iou}}$ and $\mathcal{L}_{\text{CE}}$ yield significant performance enhancements, as evidenced by the increase in mIoU, highlighting their critical role in our OccMamba.

$\mathcal{L}_{\text{CE}}$	$\mathcal{L}_{\text{iou}}$	$\mathcal{L}_{\text{depth}}$	$\mathcal{L}_{\text{geo}}$	$\mathcal{L}_{\text{sem}}$	mIoU
				✓	19.2
			✓	✓	19.5
		✓	✓	✓	19.9
	✓	✓	✓	✓	21.7
✓	✓	✓	✓	✓	22.9

Table 10. Ablation study on the effect of each training loss.

6.5. More experimental results

Due to space constraints, we put the detailed class-wise performance on the SemanticKITTI dataset in this section. As shown in Table 11, our OccMamba achieves state-of-the-art results on SemanticKITTI test set.

Method	Input Modality	mIoU	road	sidewalk	parking	other ground	building	car	truck	bicycle	motorcycle	other vehicle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	traffic sign
MonoScene [4]	C	11.1	54.7	27.1	24.8	5.7	14.4	18.8	3.3	0.5	0.7	4.4	14.9	2.4	19.5	1.0	1.4	0.4	11.1	3.3	2.1
SurroundOcc [46]	C	11.9	56.9	28.3	30.2	6.8	15.2	20.6	1.4	1.6	1.2	4.4	14.9	3.4	19.3	1.4	2.0	0.1	11.3	3.9	2.4
OccFormer [52]	C	12.3	55.9	30.3	31.5	6.5	15.7	21.6	1.2	1.5	1.7	3.2	16.8	3.9	21.3	2.2	1.1	0.2	11.9	3.8	3.7
RenderOcc [32]	C	12.8	57.2	28.4	16.1	0.9	18.2	24.9	6.0	0.4	0.3	3.7	26.2	4.9	3.6	1.9	3.1	0.0	9.1	6.2	3.4
LMSCNet [35]	L	17.0	64.0	33.1	24.9	3.2	38.7	29.5	2.5	0.0	0.0	0.1	40.5	19.0	30.8	0.0	0.0	0.0	20.5	15.7	0.5
JS3C-Net [49]	L	23.8	64.0	39.0	34.2	14.7	39.4	33.2	7.2	**14.0**	**8.1**	**12.2**	43.5	19.3	39.8	**7.9**	**5.2**	0.0	30.1	17.9	15.1
SSC-RS [29]	L	24.2	**73.1**	**44.4**	38.6	**17.4**	**44.6**	36.4	5.3	10.1	5.1	11.2	44.1	26.0	**41.9**	4.7	2.4	**0.9**	30.8	15.0	7.2
Co-Occ [31]	C&L	24.4	72.0	43.5	**42.5**	10.2	35.1	**40.0**	6.4	4.4	3.3	8.8	41.2	**30.8**	40.8	1.6	3.3	0.4	**32.7**	26.6	20.7
M-CONet [45]	C&L	20.4	60.6	36.1	29.0	13.0	38.4	33.8	4.7	3.0	2.2	5.9	41.5	20.5	35.1	0.8	2.3	0.6	26.0	18.7	15.7
OccMamba-128 (ours)	C&L	**24.6**	68.7	41.0	35.9	9.1	40.8	34.8	**8.8**	8.8	6.5	8.9	**44.9**	28.7	40.6	4.2	2.6	0.6	32.0	**27.0**	**23.3**

Table 11. Performance on SemanticKITTI test set. The best and second-best are in bold and underlined, respectively.