SkyEye: Self-Supervised Bird's-Eye-View Semantic Mapping Using Monocular Frontal View Images

- Supplementary Material -

Nikhil Gosala*, Kürsat Petek*, Paulo L.J. Drews-Jr, Wolfram Burgard, and Abhinav Valada

In this supplementary material, we present additional experiments to study the performance of SkyEye. Specifically, we analyze the computational efficiency of our model in Sec. S.1 and perform an additional ablative experiment to study the characteristics of our contribution in Sec. S.2. We also present further qualitative results to demonstrate the performance of SkyEye on the KITTI-360 dataset in Sec. S.5.

S.1. Evaluation of Model Efficiency

In this section, we compare the efficiency of SkyEye with that of the other baselines on the KITTI-360 dataset. Tab. S.1 presents the results of this comparison on an NVIDIA RTX 3090 GPU. We observe that SkyEye is the most efficient of all baselines in terms of the number of learnable parameters, using 7.73 million fewer parameters as compared to the state-of-the-art fully supervised approach, PoBEV [6]. We also note that SkyEye uses significantly fewer Multiply-Accumulate (MAC) operations as compared to PoBEV. A large part of this efficiency can be attributed to the use of a 3D voxel grid representation to jointly represent features of both FV and BEV, thus alleviating the need for separate representations in FV and BEV. Further, we observe that our network design translates into a nearly 3-fold reduction in runtime as compared to PoBEV, requiring only $77.84\,\mathrm{ms}$ per sample and allowing SkyEye to be used in many real-time applications.

Table S.1. Comparison of model efficiency on the KITTI-360 dataset.

Method	# Params (M)	MAC (G)	Runtime (ms)
IPM [25]	27.83	**32.51**	71.65
VED [24]	169.20	391.49	25.42
VPN [30]	23.01	55.45	**8.68**
PON [32]	91.06	657.63	164.08
TIIM [33]	40.72	1290.16	193.82
PoBEV [6]	22.33	272.68	213.92
SkyEye (Ours)	**14.60**	219.94	77.84

S.2. Additional Ablation Experiments

In this section, we perform an ablation experiment to analyze the impact of different FV window sizes (WS) used during implicit supervision on the overall performance of the model. To this end, we perform implicit supervision using windows of size 0, 6, 10, 14, and 20, and subsequently refine these pretrained models using 1% of BEV pseudo labels. We choose 1% of BEV pseudo labels to better highlight the impact of window sizes on the overall performance of the model. Tab. S.2 presents the results of this ablation study. We observe that a window size of 10 generates the best mIoU score across all the tested window sizes, marginally outperforming a window size of 6 by $0.12\,\mathrm{pp}$. We also observe that larger window sizes report worse performance across almost all classes. Thus, we use WS $= 10$ in our SkyEye framework and perform all experiments with this window size.

S.3. Additional Quantitative Results

In this section, we provide further quantitative results to assess the influence of FV and BEV label quality on the overall performance of SkyEye. To this end, we conduct two further experiments following the same experimental setup as presented in Tab. 2, but replace the source of supervision in either FV or BEV.

S.3.1. Impact of BEV Label Quality

We quantify the impact of BEV label quality by replacing the BEV pseudolabels during finetuning with the corresponding BEV ground truth labels. Our results, shown in Tab. S.3, indicate that the use of BEV ground truth labels does not significantly impact the lower percentage splits of 0.1% and 1%. However, it provides a significant performance boost for higher BEV percentage splits. We attribute this to the increased quality of supervision for dynamic objects such as car and 2-wheeler, for which the pseudolabel generation pipeline often introduces imperfections. This observation is supported by the significantly higher IoU scores obtained for these two object classes when BEV pseudolabels are replaced with BEV ground truth labels.

S.3.2. Impact of FV Label Quality

We investigate the dependency of SkyEye on the quality of FV labels during the pretraining phase by replacing the FV ground truth labels with FV semantic predictions. To this end, we first train the Panoptic-DeepLab [1] model for se-

Table S.2. Ablation study on the impact of Window Size on the overall performance of the model. This experiment uses only 1% of BEV pseudo labels to highlight the impact of window size. All scores are reported on the KITTI-360 dataset.

Window Size	Road	Sidewalk	Building	Terrain	Person	2-Wheeler	Car	Truck	mIoU
0	71.82	33.27	36.00	41.23	2.80	0.00	24.97	5.42	26.94
6	71.97	35.63	36.86	39.07	2.93	0.00	28.23	9.55	28.03
10	72.00	33.76	37.59	38.75	3.77	1.81	28.04	9.53	**28.15**
14	71.81	32.87	36.93	38.47	3.02	2.87	28.58	7.93	27.81
20	71.21	35.16	35.77	38.26	3.59	2.35	28.91	5.16	27.43

Table S.3. Ablation study on the impact of Implicit Supervision on the overall network performance. All scores are reported on the KITTI-360 dataset.

BEV (%)	BEV GT	FV GT	Epochs	Road	Sidewalk	Building	Terrain	Person	2-Wheeler	Car	Truck	mIoU
0.1	✓	✓	300	68.78	28.20	35.56	26.08	0.00	0.00	21.61	0.00	22.53
	✓	✗		57.35	19.36	22.13	12.54	0.00	0.00	10.56	0.00	15.24
1	✓	✓	100	72.56	34.33	36.70	41.66	0.00	0.16	33.85	10.39	28.71
	✓	✗		70.69	31.13	32.38	40.08	0.00	0.00	29.08	3.95	25.91
10	✓	✓	50	76.07	40.30	40.30	45.33	3.75	8.15	42.64	10.73	33.41
	✓	✗		73.16	37.08	38.41	45.45	3.66	6.69	40.60	7.94	31.62
50	✓	✓	30	76.43	39.89	45.22	46.64	5.10	7.93	42.43	12.30	34.49
	✓	✗		72.50	36.92	39.41	45.12	3.63	7.46	41.21	9.73	32.00
100	✓	✓	20	75.99	41.35	44.26	45.91	4.08	9.53	44.13	12.68	34.74
	✓	✗		72.82	38.27	40.86	45.86	3.59	7.74	41.37	9.74	32.53

mantic segmentation on Cityscapes [2] and use the resulting model to generate FV semantic predictions on KITTI-360. We then pretrain SkyEye using these FV predictions and subsequently finetune it on different percentages of BEV ground truth labels. Tab. S.3 presents the results of this study. We observe that the use of FV predictions in the pretraining stage only results in a marginal decrease in performance for all percentage splits except 0.1%. The drop in the 0.1% split can be attributed to the degradation of FV label quality in the pretraining phase, which cannot be sufficiently compensated for by the small amount for BEV labels available in this split.

S.4. Pseudolabel Generation

Hyperparameters: For pseudolabel generation, we use 10 future frames with a step size of 2 resulting in $\mathcal{W}$=6 for the accumulation step. For DBSCAN, we set *eps=0.2*, *min_pts=20* for person and 2-wheeler, and *eps=0.5*, *min_pts=50* for car and truck. For the RANSAC algorithm used to generate cluster ellipses, we set *min_samples=20*, *residual_thresh=3*, and *max_iters=10*.

Pseudolabel Quality: We quantify the quality of our BEV pseudolabels by comparing them with the BEV ground truth labels and achieve an mIoU score of 48.62%.

S.5. Additional Qualitative Results

In this section, we present additional qualitative results on the KITTI-360 dataset and also present the BEV semantic maps obtained when our SkyEye model is trained using different percentages of BEV pseudo labels.

S.5.1. BEV Semantic Mapping

We qualitatively demonstrate the performance of our model by comparing the output from our network to those obtained from the state-of-the-art fully supervised approach, PoBEV [6]. We also show an Error/Improvement map in the rightmost column to highlight the difference in predictions between both approaches. Fig. S.1 presents the semantic BEV map predictions obtained from both networks. We observe from the figures that our approach is mostly on par with PoBEV across a wide range of scenarios from straight roads with multiple parked cars, to curved roads, and to complex intersections. As already noted in the main paper, we observe from Fig. S.1(d, e, f) that SkyEye performs significantly better on static regions as compared to PoBEV which can be confirmed by looking at the large swathes of green in the last column. Our model is able to estimate the extent of roads and sidewalks better than PoBEV, and this improvement in performance can be attributed to training using implicit supervision which encourages the model to learn spatially coherent representations. We also observe from Fig. S.1(a, b, c) that our model is able to accurately estimate the locations of multiple cars in the scene and these images qualitatively look very similar to that of PoBEV. However, we note that when cars are extremely close to each other, our model sometimes stretches the extent of cars and merges multiple cars into one big blob. This is a limitation of our model and is largely a consequence of using only a forward-facing camera during implicit supervision which

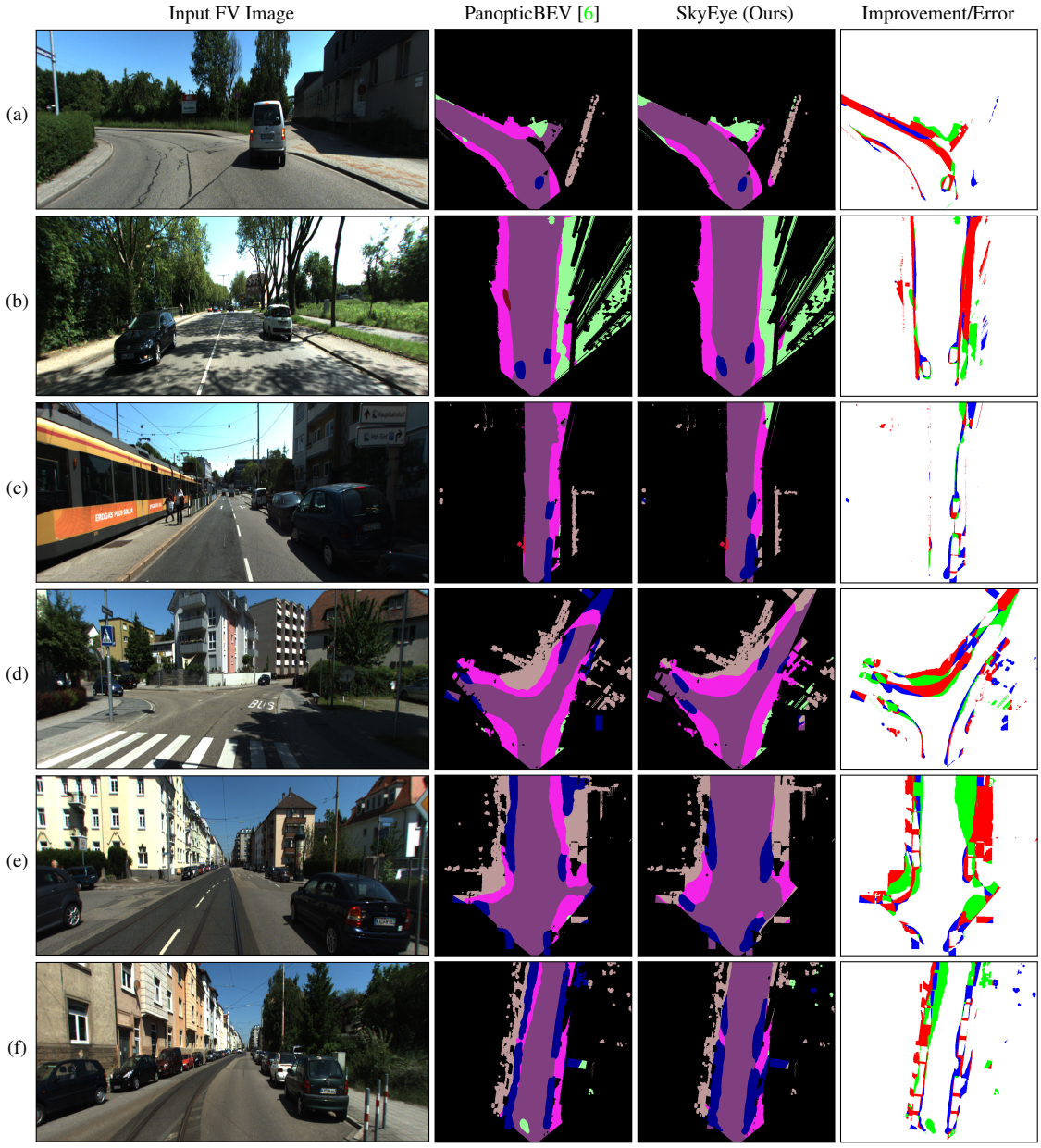

Figure S.1. Additional Qualitative Results on the KITTI-360 dataset. The rightmost column shows the Improvement/Error map which depicts pixels misclassified by PoBEV and correctly predicted by SkyEye in green, pixels misclassified by SkyEye and correctly by PoBEV in blue, and pixels misclassified by both models in red.

inhibits the model from extracting the extents of all four sides of vehicles. Furthermore, our method does not have access to the ground truth BEV map obtained using LiDAR data that is able to perceive both close and distant vehicles with higher precision. Nevertheless, our model performs on par with PoBEV without using any ground truth supervision in BEV, thus highlighting the impact of our self-supervised BEV semantic mapping framework.

S.5.2. Different Percentages of BEV Pseudolabels

In this section, we qualitatively evaluate the impact of using different percentages of BEV pseudo labels on the overall performance of our model. Fig. S.2 presents the results of this evaluation. We observe a very interesting trend across samples when training our model with different percentages of BEV pseudo labels. Across all images in Fig. S.2, we observe that our model gradually improves its reasoning about dynamic cars in the scene with an increase in the percentage

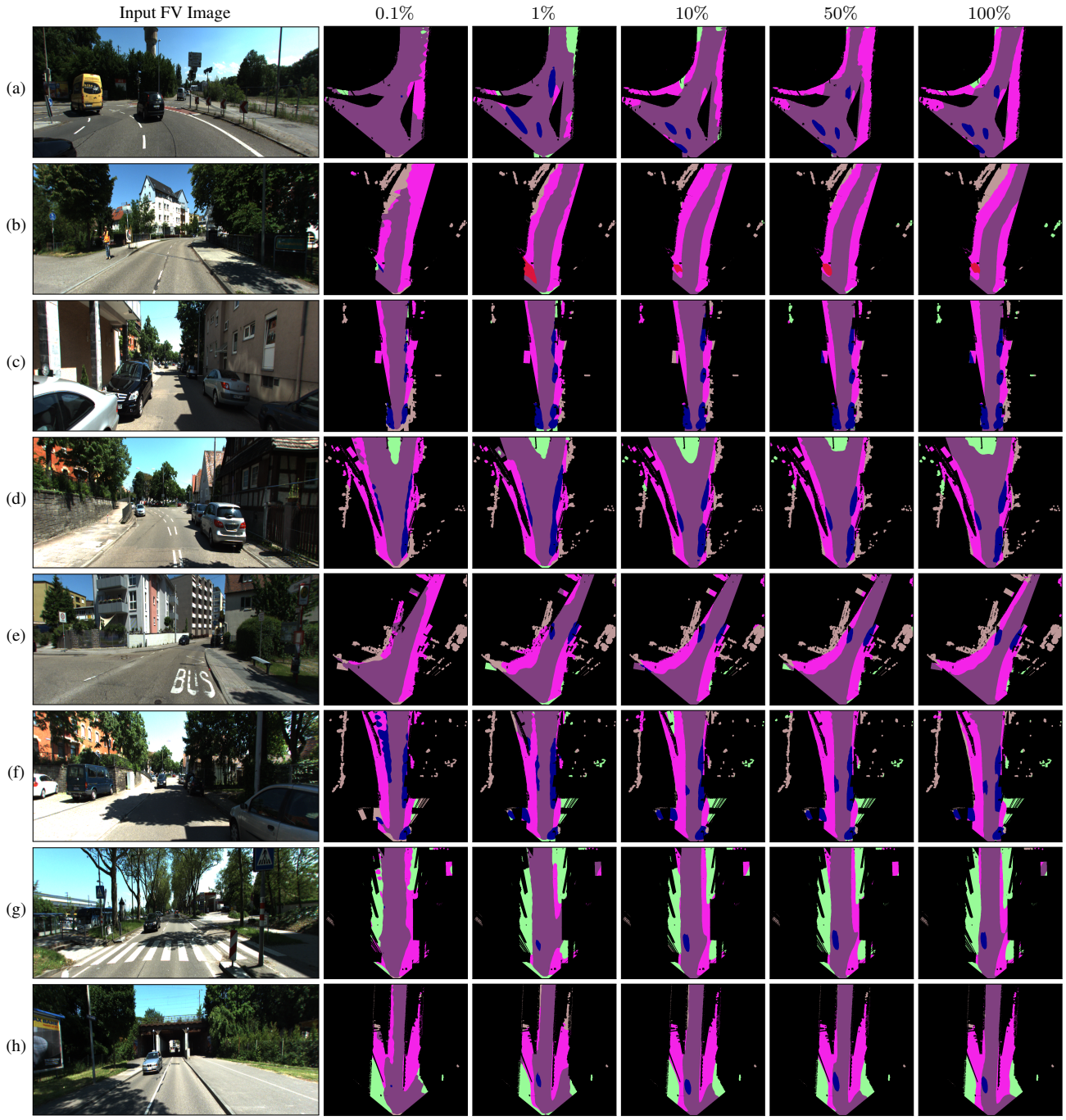

Figure S.2. Qualitative results obtained when SkyEye is trained using 0.1%, 1%, 10%, 50% and 100% of BEV pseudolabels.

of BEV pseudo labels. When using only 0.1% of pseudo labels, our model fails to identify any dynamic cars in the scene. Upon increasing the number of pseudo labels to 1%, we see that our model starts reasoning about the locations of dynamic objects but the cars look stretched and artificial. However, upon further increasing the percentage of BEV pseudo labels to 10%, our model predicts the locations as well as the extent of cars accurately. Further increase in the percentage of pseudo labels refines the predictions and better constrains the extent of vehicles, but no significant improvement can be observed between the BEV maps obtained using 10%, 50% and 100% of pseudo labels. This supports our findings in Tab. 2 of the main paper where we observe no large changes in mIoU scores when using 10%, 50%, and 100% of BEV pseudo labels to train the model.

Further, a special observation can be made when the scene is fully static. Fig. S.2(c, d, f) depict static scenes with multiple parked cars wherein we observe that our model with 0.1% of pseudo labels is already able to predict the locations of cars with high accuracy. This accurate estimation of the location of static cars with very few pseudo labels can be attributed to the structure infused into the 3D voxel grid representation by implicit supervision. This strong structural signal thus enables the model to reason about the world in BEV even when the model is exposed to extremely sparse samples in BEV.

References

[1] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12475–12485, 2020. 11

[2] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 12

[3] Nikhil Gosala and Abhinav Valada. Bird's-eye-view panoptic segmentation using monocular frontal view images. *IEEE Robotics & Automation Letters*, 7(2):1968–1975, 2022. 1, 2, 5, 6, 8, 11, 12, 13

[4] Chenyang Lu, Marinus Jacobus Gerardus van de Molengraft, and Gijs Dubbelman. Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks. *IEEE Robotics and Automation Letters*, 4(2):445–452, 2019. 2, 5, 6, 11

[5] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991. 5, 6, 11

[6] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. 2, 5, 6, 11

[7] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2020. 2, 5, 6, 11

[8] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9200–9206. IEEE, 2022. 2, 5, 6, 11