ShelfOcc: Native 3D Supervision beyond LiDAR for Vision-Based Occupancy Estimation

Supplementary Material

A. Additional Qualitative Results

We provide further qualitative comparisons to complement the results in the main paper. First, we compare STCOcc [35] trained on our pseudo-labels against the previous state-of-the-art method GaussianFlowOcc [2], as well as the Occ3D-nuScenes ground truth [57] in Fig. A.1. For each scene, we show the three front-facing camera images alongside 3D predictions rendered from an elevated third-person viewpoint behind the ego vehicle for a single frame. STCOcc produces clean, dense, and well-regularized occupancy predictions with minimal noise or depth bleeding, whereas GaussianFlowOcc exhibits pronounced artifacts stemming from its 2D supervision pipeline. It is clearly visible that the model trained with our framework can correctly estimate the 3D shape of objects, while previous work suffers from depth bleeding.

We also visualize the effect of the different *versions* of our proposed pipeline introduced in Fig. 1, rendered from a top-down viewpoint in Fig. A.2. The naïve single-frame variant (version 1) yields sparse and incomplete geometry, missing large portions of the scene. Aggregating all points across the sequence without distinguishing motion (version 2) introduces object trails and leads to missing objects when low-confidence points are filtered out, both of which degrade the supervision quality. In contrast, our final design (version 3), which explicitly separates static and dynamic content and applies confidence filtering only to the static scene, produces a dense, coherent scene without trails or missing objects. Finally, videos accompanying all qualitative comparisons, including comparison shown in the main paper, are available in the official GitHub repository.

B. Additional Quantitative Results

B.1. Comparison to LiDAR-Supervised Methods

Table B.1 provides an extended comparison between our approach and methods that rely on LiDAR for supervision, including weakly supervised approaches AGO [31] and VEON [78] using LiDAR data, as well as fully supervised methods trained with semantic 3D ground truth. Interestingly, both COTR and CVT-Occ, when trained solely on our pseudo-labels, achieve competitive performance relative to AGO and even surpass VEON, despite both of the latter relying on LiDAR for geometric supervision. STCOcc surpasses them even more clearly in terms of mIoU. Unfortunately, the authors of these methods do not report geometric IoU, where we would expect them to perform more strongly due to their access to LiDAR depth. These findings highlight that our LiDAR-free, shelf-supervised framework can match or even outperform prior methods that depend on LiDAR supervision for geometry. At the same time, there remains a performance gap compared to fully supervised methods trained directly on densely annotated LiDAR voxel labels. While LiDAR-based occupancy estimation is not the focus of this work, we provide these numbers to contextualize the remaining room for improvement relative to full 3D supervision. We omit AutoOcc [81] from this comparison, as we were unable to retrace their differing evaluation protocol.

B.2. Per-Class Semantic Segmentation Ablation

We additionally report per-class performance for the semantic segmentation ablation (cf. Tab. 3) in Tab. B.2. The results confirm that the proposed sky grounding technique substantially improves the quality of the pseudo-labels across almost all classes. By reducing spurious false positives from the open-vocabulary detector the resulting labels become significantly cleaner and more stable. Notably, improvements are pronounced for low-frequency classes such as *bus*, *traffic cone*, and *truck*. For these categories, sky grounding prevents the detector from erroneously predicting object boxes in every frame, enabling more accurate class assignments and reducing confusion with the background.

B.3. Ablation on Resolution and Backbone

We further investigate the impact of input image resolution and backbone capacity on models trained with our ShelfOcc pseudo-labels in Tab. B.3. For this study, we use CVT-Occ as the representative architecture. Our results show that CVT-Occ benefits noticeably from scaling both the backbone and the input resolution. Doubling the resolution to 512×1408 and replacing the ResNet-50 encoder with a larger ResNet-101 already yields a clear improvement in semantic mIoU. Increasing the resolution further to the full nuScenes input size leads to additional gains, improving not only mIoU but also geometric IoU. We also experimented with VoVNet-99, which has a parameter count comparable to ResNet-101. While its semantic mIoU is similar to that of ResNet-101, it achieves slightly lower geometric IoU, suggesting that encoder architecture plays a nontrivial role in exploiting the pseudo-label supervision. Overall, these findings indicate that our 3D supervision pipeline can effectively leverage higher-resolution inputs and stronger back-

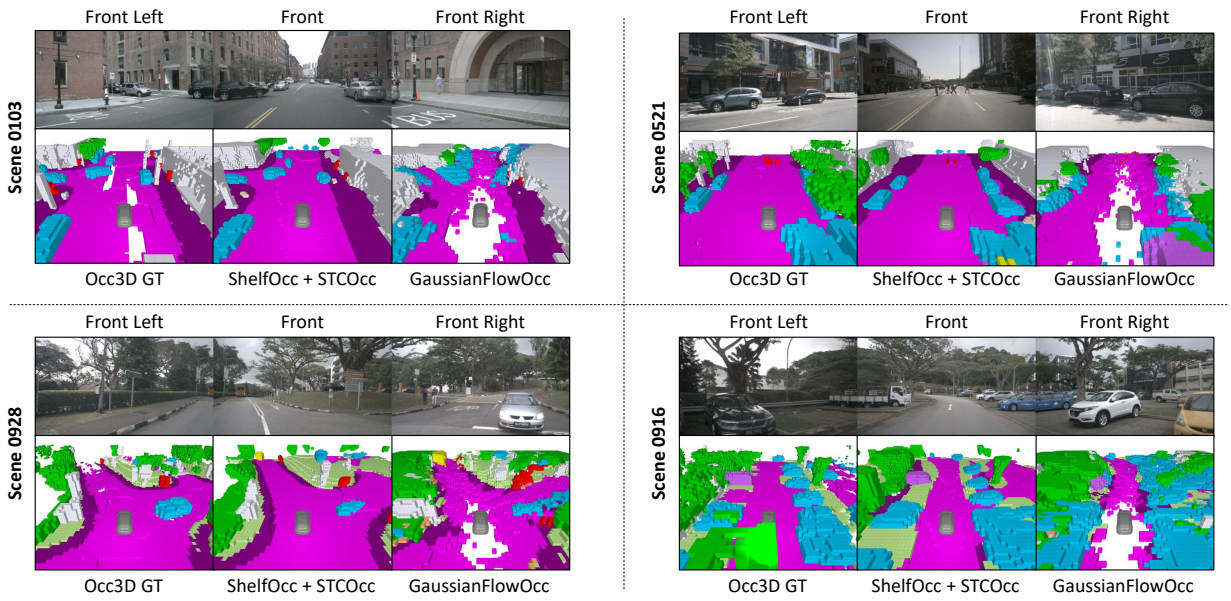

Figure A.1. **Qualitative comparison with previous state-of-the-art.** We show predictions from STCOcc [35] trained on our ShelfOcc pseudo-labels, compared against GaussianFlowOcc [2] and the Occ3D-nuScenes ground truth. STCOcc produces cleaner and more geometrically consistent occupancy predictions, demonstrating the benefits of our 3D supervision.

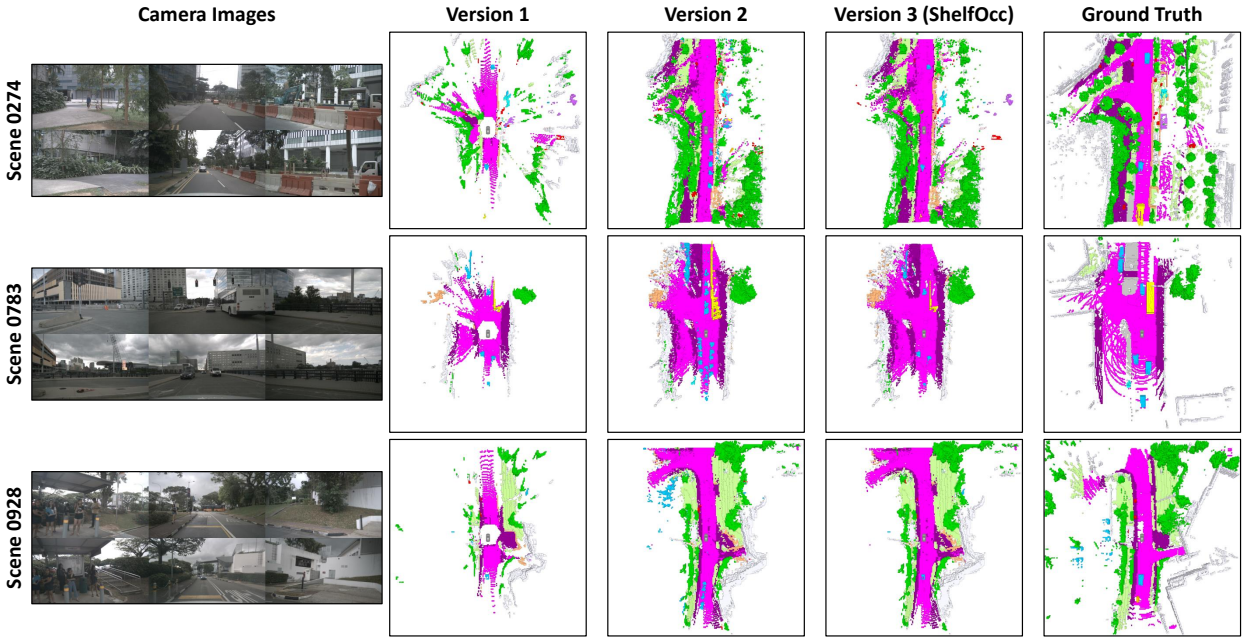

Figure A.2. **Qualitative comparison of the different *versions* of our proposed pipeline.** We visualize pseudo-labels produced by the three pipeline variants introduced in the main paper: (1) the naïve single-frame approach, (2) full temporal aggregation without handling motion, and (3) our final design, which aggregates static geometry while treating dynamic objects separately. The comparison highlights how version 3 avoids sparsity, object trails, and missing objects, resulting in clean and coherent 3D supervision.

bones, offering additional room for performance scaling.

B.4. Comparison to SfmOcc

As noted in the main paper, the recent work SfmOcc [42] adopts a similar strategy of generating 3D occupancy

Table B.1. **Performance on the Occ3D-nuScenes validation set compared to methods trained with LiDAR data.** The *LiDAR* column indicates whether a method uses raw 3D LiDAR points during training, while the *Annotations* column denotes methods that rely on semantically annotated LiDAR ground truth (e.g., voxel labels from Occ3D-nuScenes). Despite using only camera images for supervision, our shelf-supervised pipeline outperforms prior methods that depend on LiDAR-based geometric supervision.

Method	LiDAR	Annotations	mIoU	IoU	barrier	bicycle	bus	car	cons. vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	sidewalk	terrain	manmade	vegetation
AGO [31]	✓	✗	21.39	-	6.75	6.43	14.00	22.82	5.57	16.66	13.20	6.80	10.53	15.89	71.48	34.48	41.37	29.33	25.66
VEON [31]	✓	✗	17.07	-	10.40	6.20	17.70	12.70	8.50	7.60	6.50	5.50	8.20	11.80	54.50	25.50	30.20	25.40	25.40
CVT-Occ [70]	✓	✓	42.36	-	49.46	23.57	49.18	55.63	23.10	27.85	28.88	29.07	34.97	40.98	81.44	51.37	54.25	45.94	39.71
COTR [41]	✓	✓	46.41	75.01	52.11	31.95	46.03	55.63	32.57	32.78	30.35	34.09	37.72	41.84	84.48	57.55	60.67	51.99	46.33
STCOcc [35]	✓	✓	46.83	-	52.3	32.2	50.5	56.5	31.7	33.9	33.4	33.8	38.9	44.9	83.9	57.1	60.1	50.6	42.7
Ours: ShelfOcc + COTR [41]	✗	✗	18.65	53.71	9.10	6.20	22.92	22.08	1.66	5.94	9.92	8.55	0.0	15.32	67.93	31.13	38.76	23.11	17.15
Ours: ShelfOcc + CVT-Occ [70]	✗	✗	19.21	52.72	11.53	6.38	20.39	21.92	4.20	10.18	9.02	10.67	0.89	13.08	68.42	31.23	41.42	22.74	16.15
Ours: ShelfOcc + STCOcc [35]	✗	✗	22.87	56.14	13.98	11.36	25.27	25.80	7.25	16.61	12.91	13.42	5.37	17.15	68.01	34.66	42.73	25.63	22.89

Table B.2. **Effect of improved semantic segmentation.** We train STCOcc [35] on our pseudo-labels with and without using the sky grounding technique. Using the improved semantic segmentation also improves downstream occupancy estimation performance.

Method	Sky Grounding	mIoU	IoU	barrier	bicycle	bus	car	cons. vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driv. surf.	sidewalk	terrain	manmade	vegetation
Ours: *ShelfOcc*	✗	6.21	17.21	3.7	1.93	3.81	5.57	1.55	3.27	3.95	5.39	0.10	2.87	20.17	9.27	10.46	8.21	12.91
Ours: *ShelfOcc*	✓	9.62	26.00	6.58	3.28	7.02	8.81	2.57	4.74	4.97	8.1	0.12	5.41	34.59	14.58	18.16	10.91	14.45
Ours: *ShelfOcc + STCOcc [35]*	✗	22.48	54.26	11.96	9.89	23.64	25.88	7.89	15.24	13.31	14.43	7.73	16.84	67.64	35.51	40.13	24.43	22.67
Ours: *ShelfOcc + STCOcc [35]*	✓	22.87	56.14	13.98	11.36	25.27	25.80	7.25	16.61	12.91	13.42	5.37	17.15	68.01	34.66	42.73	25.63	22.89

Table B.3. **Ablation of Image Backbones for CVT-Occ.** Comparison of ResNet variants and VoVNet while simultaneously scaling image resolution along with backbone size. We observe further performance increases when scaling up the model size and image resolution.

Backbone	Image Size	mIoU	IoU
ResNet-50	256 x 704	19.21	52.72
ResNet-101	512 x 1408	19.78	52.77
ResNet-101	928 x 1600	20.24	53.02
VoVNet-99	928 x 1600	20.23	51.84

pseudo-labels to EasyOcc [15] and our work by employing structure-from-motion techniques to generate the depth maps. The authors utilize the complete set of training images for a given sequence, applying bundle adjustment to align the images and estimate camera poses. From these aligned poses, they subsequently derive depth maps. Semantic maps are computed using a pre-trained, open-vocabulary image model similar to our work. These semantic and geometric maps are then fused to generate pseudo-labels, which provide explicit supervision for the 3D semantic occupancy prediction task.

A significant limitation of the SfmOcc pipeline, however, is its reliance on a static scene assumption. The method's effectiveness is contingent upon scenes that exhibit substantial ego-motion while containing minimal dynamic objects. Consequently, the authors generate pseudo-labels and train their network on a curated subset of the nuScenes dataset where these conditions are met. In Tab. B.4, we present a direct comparison between ShelfOcc and SfmOcc on this specific subset of static scenes.

Table B.4. **ShelfOcc vs. SfmOcc on nuScenes static scene subset.**

Method	mIoU/IoU **w/ mask**	mIoU/IoU **w/o mask**	RayIoU
SfmOcc [42]	9.89 / **49.5**	**7.19** / 5.48	5.99
ShelfOcc (Ours)	**9.9** / 27.51	6.7 / **17.87**	**16.14**

The results in Tab. B.4 indicate that SfmOcc achieves a higher geometric IoU within the visible regions (IoU w/ mask). This is likely attributable to its methodology of aggregating all scene geometry, which includes static instances of typically dynamic classes (e.g., parked cars). These static dynamic objects are considered dynamic in our method and consequently filtered out, as we just use the semantic segmentation to determine which pixels are dynamic.

However, the performance of SfmOcc degrades substantially in non-visible areas, resulting in a significantly sparser reconstruction, as evidenced by its low IoU of 5.48 when the mask is removed. This limitation is further underscored by the RayIoU metric, where SfmOcc scores only 5.99, in-

Table B.5. **Occupancy estimation performance on SurroundOcc [64].**

Method	LiDAR	mIoU	IoU
MonoScene [5]	✓	7.31	23.96
TPVFormer [20]	✓	11.66	11.51
BEVFormer [33]	✓	16.75	30.50
GaussianFormer [22]	✓	19.10	29.83
SurroundOcc [64]	✓	20.30	31.49
ShelfOcc + STCOcc [35]	✗	8.25	15.71

dicating a failure to infer geometry beyond the immediate field of view. ShelfOcc demonstrates markedly superior scene completion capabilities, achieving a RayIoU of 16.14.

Most critically, the static scene assumption inherent to SfmOcc constrains its applicability to a limited subset of scenarios. ShelfOcc, by contrast, is not bound by this limitation and can be universally applied to both static and dynamic scenes, highlighting its broader utility and robustness.

B.5. Performance on SurroundOcc

SurroundOcc [64] offers an additional method for generating 3D semantic occupancy ground truth for the nuScenes dataset using LiDAR data. Different from Occ3D-nuScenes [57] though, the evaluation protocol does not use any visibility mask, similar to the RayIoU metric used in the main paper. For completeness, we further provide results on the SurroundOcc dataset in Table B.5 when training STCOcc on our pseudo-labels, but *without* any retraining or grid adjustments (SurroundOcc has a different grid configuration). However, we did not find any other weakly supervised methods that provide results for this dataset, thus there is no fair comparison possible. For reference, we include some results of methods trained with the full 3D voxel ground truth based on semantically annotated LiDAR. The results show that even without any reconfiguration or additional training, our method generalizes reasonably well across different benchmarks.

C. Details on Semantic Segmentation

We provide the full vocabulary used to query the open-vocabulary detector Grounding DINO [37] in Tab. C.6. As described in the main paper, each category is queried individually by forwarding a prompt of the form *"QUERY . sky"* through the model. Including the background token *sky* encourages the detector to identify sky regions explicitly, which in turn reduces false positives for the target query. For each forward pass, we discard all predicted boxes corresponding to *sky* and retain only the boxes associated with the target query. To ensure high-quality detections, we further filter out any box whose predicted logit falls below 0.2. All remaining boxes across all query categories are aggregated and passed to the SAM segmentation model, which generates a mask for each box. The logit of the originating Grounding DINO detection is assigned to every pixel within the corresponding SAM mask. To construct the final semantic segmentation map, we overlay all predicted masks and perform per-pixel selection based on the highest associated logit. This produces a dense, open-vocabulary segmentation result that serves as the semantic input to our pseudo-label generation pipeline.

D. Details on Experimental Setup

All models were trained for 24 epochs using four NVIDIA A100 GPUs. The total training time was approximately 48 hours. The pseudo-label generation process, which is performed prior to training, consists of two main stages. The initial stage involves inference with the foundation model, which takes approximately 36 hours to complete on four GPUs. Subsequently, the occupancy voxelization pipeline processes the output to generate the final pseudo-labels, a step that requires an additional 2 hours.

Table C.6. **Vocabulary used for querying Grounding DINO [37] for open-vocabulary object detection.** The left column lists the class labels, and the right column contains the prompts used during mask generation.

Class	Prompts
'barrier'	'barricade', 'barrier'
'bicycle'	'bicycle'
'bus'	'bus'
'car'	'car', 'sedan', 'van'
'construction_vehicle'	'excavator', 'crane'
'motorcycle'	'motorcycle', 'scooter'
'pedestrian'	'person', 'pedestrian'
'traffic_cone'	'traffic-cone'
'trailer'	'trailer'
'truck'	'lorry', 'truck'
'driveable_surface'	'highway', 'street', 'roadmarking'
'sidewalk'	'sidewalk', 'walkway'
'terrain'	'turf', 'grass', 'sand'
'manmade'	'building', 'wall', 'fence', 'pole', 'sign', 'light', 'bridge', 'billboard'
'vegetation'	'bush', 'plants', 'tree'