MonoSOWA: Scalable monocular 3D Object detector Without human Annotations

Supplementary Material

6. Ablations

6.1. Distance based AP

To provide a deeper insight into our autolabelling pipeline, we evaluate pseudo labels on the KITTI-360 dataset [14] based on the distance of the target cars in Tab. 9. We calculate the mean average precision on cars that are within the specified range.

$AP_{BEV}/AP_{3D}@0.5$				
0-10m	10-20m	20-30m	30-40m	40-50m
50.81/29.27	52.33/35.72	41.85/26.51	12.60/7.59	0.75/0.21

Table 9. Auto-labelling mean average precision on KITTI-360 [14] training set depending on object distance

6.2. Runtime analysis

We provide a runtime analysis of auto-labelling on the KITTI-360 [14] dataset in Tab. 10, where our method takes approximately 1.3 seconds to autolabel each frame. We ran our experiments on a single node with 1x AMD EPYC 7534 CPU and 1x NVIDIA A100 GPU.

Stage	Run time / frame [ms]
Metric3DV2-G [8]	792 ± 2.2
MViTv2-H [13]	318 ± 2.0
Frames aggregation	37 ± 19.8
Optimization	164 ± 276
Total (ours)	$1\,311 \pm 291.6$
VSRD [18]	$\approx 900\,000$

Table 10. Auto-labelling runtime on KITTI-360 [14] dataset

6.3. Monocular Metric Depth Estimator

We provide an analysis of how different Monocular Metric Depth estimators affect our methods' performance in Tab. 11. Metric3Dv2 [8] performs the best, while Depth Anything [44] and ZoeDepth [2] lacks behind. It is worth noting that only ZoeDepth has been fine-tuned on the KITTI dataset. We have also tried methods such as UniDepth [28] or SPIDepth [12]; however, their performance is highly dependent on the fine-tuning on the target dataset. On the other hand, Metric3Dv2 does not depend on fine-tuning, and it hasn't seen the KITTI dataset during training at all. Showing its powerful abilities and enabling our method as a zero-shot.

We extend our evaluation to the larger KITTI-360 dataset [14], with results presented in Tab. 12. In this zero-shot setting, only Metric3Dv2 [8] performs effectively. In contrast, both Depth Anythingv2 [44] and ZoeDepth [2] struggle on this dataset, on which they were not fine-tuned. To account for camera differences, we apply depth scaling to compensate for the different focal length of KITTI-360 compared to KITTI [14] and VirtualKITTI2 [4].

Metric Monocular Depth Estimator	$AP_{BEV}/AP_{3D}@0.5$		
	Easy	Moderate	Hard
ZoeDepth [2]	28.34/19.88	19.52/16.02	21.31/14.12
Depth Anything [44]	28.16/19.77	18.67/15.43	20.58/13.68
Metric3Dv2 [8]	41.72/33.48	35.84/25.09	33.61/23.01

Table 11. Auto-labelling mean average precision on KITTI [6] training set depending on metric depth estimation model

Metric Monocular Depth Estimator	$AP_{BEV}/AP_{3D}@0.5$	
	Easy	Hard
ZoeDepth [2]	26.02/13.74	25.23/13.27
Depth Anything [44]	2.11/0.59	2.13/0.59
Metric3Dv2 [8]	61.17/47.07	51.92/45.51

Table 12. Auto-labelling mean average precision on KITTI-360 [14] training set depending on metric depth estimation model

6.4. Human-annotated masks

In order to make a fair comparison, we additionally provide an experiment where instead of MViTV2 [13] segmentation masks, we use segmentation masks created by humans to have the exact same setting as VSRD [18]. The experiment uses the KITTI-360 [14] dataset, where an instance segmentation mask is provided for each object in each frame. This helps in two ways. First, we employ masks for point extraction, and as they are ground-truth, no false positives or negatives are present. Second, as the segmentation also comes with instance id, tracking vehicles is trivial.

As shown in Table 13, using ground-truth masks signif-

Method	Masks	$AP_{BEV}/AP_{3D}@0.5$		$AP_{BEV}/AP_{3D}@0.3$	
		Easy	Hard	Easy	Hard
VSRD [18]	yes	29.07/21.77	22.83/16.46	58.40/**50.86**	50.61/43.45
Ours	no	38.41/29.98	**35.26/27.56**	50.84/42.72	49.22/**46.59**
Ours	yes	**38.90/34.44**	33.49/26.93	**59.41**/49.03	**54.11**/43.84

Table 13. Ablation study on the effect of using human-annotated masks instead of 2D detector for generating pseudo-labels to train MonoDETR [49] on KITTI-360 [14] training set and evaluating on KITTI-360 test set

LiDAR for training	Labels	$AP_{BEV}/AP_{3D}@0.5$ Easy	Moderate	Hard	$AP_{BEV}/AP_{3D}@0.3$ Easy	Moderate	Hard
no	pseudo	59.76/51.55	44.08/37.09	36.99/33.15	73.38/72.70	57.23/56.30	48.59/47.70
yes	pseudo	64.50/60.04	51.44/44.50	44.48/37.76	78.64/77.62	65.76/63.79	57.91/56.36
yes	human	**67.44/65.09**	**53.46/47.39**	**46.80/44.58**	**80.30/79.72**	**67.16/65.87**	**59.54/58.83**

Table 14. Ablation study on the effect of using LiDAR scans to generate pseudo-labels to train MonoDETR [49] on KITTI [6] training set and evaluate on KITTI validation set.

icantly increases the accuracy, especially on the 0.3 IoU both on BEV and 3D. It also demonstrates that when using the same inputs as VSRD, our method outperforms VSRD when using 0.3 IoU evaluation, while maintaining its superior performance at 0.5 IoU, both on BEV and 3D, therefore achieving *state-of-the-art results in both metrics*.

6.5. LiDAR and pseudo-LiDAR

We further provide an analysis of how using LiDAR would affect our method's performance. Although monocular metric depth estimation has advanced significantly, it is still far from perfect. In Table 14, we present a comparison of our method using either original LiDAR scans captured by Velodyne HDL-64E or using pseudo-LiDAR, which is created by lifting depth predictions from Metric3Dv2 [8] as in our method. Using original LiDAR scans significantly improves the average precision in both 0.3 and 0.5 IoU on Bird's Eye View (BEV) and 3D. LiDAR scans not only do not suffer from depth prediction errors but because of their consistency between frames, our method can use them to refine imprecise transformations between frames by employing the Iterative Closest Points (ICP) algorithm [32]. Unfortunately, for pseudo-LiDAR, the ICP algorithm diverges in some cases.

To decrease the effect of the inaccurate pseudo-lidar, we speculate that using low-quality LiDAR generating sparse point clouds could guide and refine the pseudo-lidar, which is, on the other hand, very dense. Using a combination of LiDAR and pseudo-lidar might also mitigate the disadvantages of both methods.

6.6. Ego-vehicle pose noise

To evaluate how the inaccuracy in the ego-vehicles poses (GPS and/or IMU noise) affects the performance of our method, we provide an analysis in Tab. 15. As it is a common practice, we model GPS noise as a first-order Gaussian Markov process, including linear drift and signal drops, and then measure the level of noise as the average difference in meters (RMSE) between the noisy and ground truth signal. As is also a common practice, we try to denoise the signal with a Kalman Filter to reduce its impact. Then we use those noisy ego-vehicle poses in our pipeline.

Our results show that, as the noise is relatively low, it has minimal effect on the performance. However, as the noise increases, the performance unfortunately decreases. It is worth noting that, as the auto-labelling is offline, more filtering and more sophisticated de-noising methods can likely be used to further mitigate the effect of noise.

Noise level Avg. RMSE	$AP_{BEV}/AP_{3D}@0.5$ Easy	Moderate	Hard
0m	41.72/33.48	35.84/25.09	33.61/23.01
0.1m	40.16/34.97	34.86/25.74	32.33/23.21
0.2m	38.01/31.05	32.38/23.40	30.09/21.08
0.5m	23.82/17.09	17.30/14.43	18.90/12.45
1.0m	15.83/11.33	10.84/9.14	11.81/7.38

Table 15. Auto-labelling mean average precision on KITTI [6] training set depending on GPS/IMU noise level

We extended our ablation study to the larger KITTI-360 dataset [14] to further validate our approach. The results demonstrate the model's robustness on the BEV metric, which remains largely unaffected by noise. In contrast, the 3D metric proved more sensitive to these perturbations. We attribute this to the different hyperparameters for car state classification and the increased number of frames used for aggregation. This observation is consistent with our findings on the original KITTI dataset [6], where higher noise levels also led to a predictable degradation in 3D performance.

Noise level Avg. RMSE	$AP_{BEV}/AP_{3D}@0.5$ Easy	Hard
0m	66.04/50.16	64.74/48.79
0.1m	66.26/13.70	64.30/7.56
0.2m	63.96/13.60	53.43/7.52
0.5m	62.13/13.10	50.33/7.26
1.0m	48.04/8.64	36.36/6.41

Table 16. Auto-labelling mean average precision on KITTI-360 [14] training set depending on GPS/IMU noise level

6.7. Number of frames used in aggregation

Tab. 17 shows the ablation study on the number of frames used in the aggregation process. Perhaps surprisingly, aggregating over longer sequences does not yield better performance. We speculate that the distant car poses a significant challenge for the detector itself; therefore, creating precise 3D labels for such objects does not translate to better 3D detection accuracy.

No. of frames	AP_{BEV}/AP_{3D}@0.5		
	Easy	Moderate	Hard
± 1	20.41/17.40	18.22/15.88	19.99/13.75
± 10	39.60/**34.37**	32.85/**24.72**	30.53/22.25
± 20	39.77/33.62	33.89/**24.72**	31.76/**22.59**
± 30	**40.21**/33.05	**34.24**/24.43	**32.10**/22.33
± 50	39.27/32.37	33.33/23.44	31.35/21.57
± 100	39.23/32.19	28.29/23.05	31.14/21.25

Table 17. Number of frames used in the aggregation process ablation evaluating pseudo-labels directly on KITTI [6] training set.

6.8. Steepness parameter in Saturated Closeness Criterion

In this ablation, we explore possible values for the steepness parameter α in the Saturated Closeness Criterion, which controls what pseudo-LIDAR points are discarded as outliers. As shown in Tab. 18, α equal to 10 achieves the best performance.

α	AP_{BEV}/AP_{3D}@0.5		
	Easy	Moderate	Hard
1	39.19/32.07	28.28/23.05	31.13/21.24
5	39.33/32.10	28.31/23.04	31.15/21.27
10	**39.38/32.26**	**33.10/23.16**	**31.20/21.36**
15	39.28/32.17	28.28/23.05	31.14/21.26
20	39.23/32.25	28.20/23.11	31.10/21.33

Table 18. Ablation of how steepness parameter α affects AP of pseudo-labels on KITTI [6] training set.

6.9. Canonical Focal Length

We also provide ablation on how the value of the canonical focal length affects the performance of MonoDETR [49] in Tab. 19. The results show that the choice of focal length is not crucial as the performance does not change significantly between the chosen values. However, it is advisable to keep the canonical focal length close to the focal length of the cameras, as 500 and 750 achieve the best results and are also the closest to the real focal length of the cameras.

Canonical Focal Length	AP_{BEV}/AP_{3D}@0.5		
	Easy	Moderate	Hard
250	**64.28**/55.49	49.10/41.98	42.35/35.64
500	63.43/53.46	49.22/42.24	42.50/35.91
750	63.71/**56.78**	**50.10/43.75**	**43.66/37.00**
1000	62.79/55.18	49.39/42.13	42.80/35.87

Table 19. Ablation study of how canonical focal length affects MonoDETR [49] AP trained on both KITTI [6] and KITTI-360 [14] training set and evaluated on KITTI validation set

7. Qualitative analysis

In Figure 7, we show multiple frames, in which we show how the MonoDETR [49] trained without human annotations performs on the KITTI-360 [14] dataset. Please note that we are using our best model trained on both KITTI [6] and KITTI-360 datasets. We show both good and bad predictions, for example, the trash bin being recognized as a car.

In Figure 8, we show how MonoDETR [49] trained by our method on Waymo Open Dataset [35] training set performs on the Waymo validation subset.

Figure 7. Qualitative analysis of MonoDETR [49] trained using our method without using human annotations on KITTI [6] and KITTI-360 [14] . White 3D bounding boxes are predictions, the number inside/near the bounding box is the confidence of each prediction.

Figure 8. Qualitative analysis of MonoDETR [49] trained using our method without using human annotations on Waymo Open Dataset [35]. Colourful 3D bounding boxes are predictions of our model.