GBlobs: Explicit Local Structure via Gaussian Blobs for Improved Cross-Domain LiDAR-based 3D Object Detection

Supplementary Material

This supplementary material details the experimental setup (Suppl. A), presents further findings on the influence of local and global features on LiDAR-based 3D object detection (Suppl. B), and provides a qualitative analysis of our results (Suppl. C).

A. Experimental Setup

To ensure reproducibility (besides the provided source code), we present detailed information about our experimental setup in Tab. 1. If not specified otherwise, we use default settings from OpenPCDet[1].

Our models were trained on the entire KITTI and nuScenes training sets, along with 20% of the Waymo dataset (a standard practice in the field). In all our experiments (except KITTI→Waymo in Tab. 3 of the main manuscript), we trained the models to simultaneously predict Cars/Vehicles, Pedestrians, and Cyclists. For fair comparison with 3D-VF [6], we trained the detector to predict only Cars/Vehicles in KITTI→Waymo. We employed standard data augmentation techniques, including random sampling, point cloud rotation, scaling, and flipping.

We use the KITTI metric [4] for evaluation (except in Tab. 6b of the main manuscript), reporting Average Precision (AP) on Bird's-eye View (BEV) / 3D views at 40 recall positions. For the in-domain Waymo evaluation in Tab. 6b (main manuscript), we report LEVEL_1 /LEVEL_2 AP (standard Waymo metric). We use Intersection over Union (IoU) thresholds of 0.7, 0.5, and 0.5 for Cars, Pedestrians, and Cyclists, respectively. KITTI→Waymo in Tab. 3 (main manuscript) uses an IoU threshold of 0.5 for Cars to ensure fair comparison. We utilize the complete validation sets of all datasets to assess the performance of our proposed method.

B. Height Bias

Autonomous driving datasets define different reference points for LiDAR point clouds, *e.g.* Waymo [7] aligns the height axis origin with the road, while KITTI [4] and nuScenes [1] use the vehicle's mounting point. This inherently introduces bias into the network. A common approach is to manually align source and target point clouds by shifting them to a shared origin [10, 11]. Otherwise the detectors fail catastrophically as demonstrated in Tab. 2. Although this is not a critical issue in our controlled setting, a detector trained with such bias could pose a significant risk in real-world applications. Our GBlobs are not affected by biases associated with global input features, as they encodes local point cloud geometry.

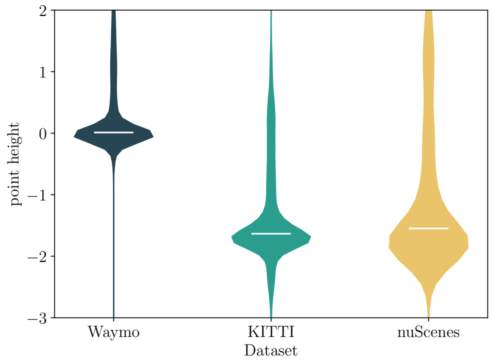


Figure 1. z distribution

C. Qualitative Results

In order to depict benefits of training a model with our GBlobs as input features, we conduct following qualitative analysis. We apply a nuScenes trained Voxel R-CNN detector to a challenging KITTI scene featuring a slightly curved road. Such detectors, trained with standard global input features, often predict false positives, even in areas without object indications.

A similar phenomenon can be observed with the SECOND [9] detector employed in the KITTI→Waymo benchmark in Fig. 3. It is noteworthy that SECOND, trained on KITTI, a dataset consisting primarily of small and mid-size European sedans, has never seen anything that resembles aerial work platforms during training. Nevertheless, when trained with global features and applied on Waymo (which has such object labeled as Vehicles), it manages to produce a detection at this location with high certainty (orange arrows in Fig. 3a). We hypothesize that the detector's prediction was influenced by specific points at specific heights. Given its training, such detections are unexpected. This raises the question of how many other detections, which are false positives, such detector produces. A model trained with our GBlobs did not make such uneducated guess (Fig. 3b).

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Gian-

[1] https://github.com/open-mmlab/OpenPCDet/

Table	Dataset	Detector						BS	E
		Name	Range	Voxel Size	Optimizer				
					Name	LR	WD		
Tab. 2	Waymo [7]	Voxel R-CNN [3]	$[-75.2\ ,-75.2\ ,-2, 75.2\ , 75.2\ , 4]$	$[0.1\ , 0.1\ , 0.15]$	Adam	1×10^{-2}	1×10^{-3}	32	30
	nuScenes [1]	Voxel R-CNN [3]	$[-75.2\ ,-75.2\ ,-2, 75.2\ , 75.2\ , 4]$	$[0.1\ , 0.1\ , 0.15]$	Adam	1×10^{-2}	1×10^{-3}	32	30
Tab. 3	KITTI [4]	PointPillars [5]	$[\ \ 0.0\ ,-39.68, -2, 69.12, 39.68, 4]$	$[0.16, 0.16, 6.0\ ]$	Adam	3×10^{-3}	1×10^{-2}	32	80
	KITTI [4]	SECOND [9]	$[\ \ 0.0\ ,-40.0\ ,-3, 70.4\ , 40.0\ , 1]$	$[0.16, 0.16, 6.0\ ]$	Adam	3×10^{-3}	1×10^{-2}	32	80
	KITTI [4]	Part-A^2 [2]	$[\ \ 0.0\ ,-40.0\ ,-3, 70.4\ , 40.0\ , 1]$	$[0.16, 0.16, 6.0\ ]$	Adam	1×10^{-2}	1×10^{-2}	32	80
Tab. 4	$*$	CenterPoint [12]	$[-75.2\ ,-75.2\ ,-3, 75.2\ , 75.2\ , 5]$	$[0.10, 0.10, 0.20]$	Adam	3×10^{-3}	1×10^{-2}	32	30
Tab. 5	nuScenes [1]	Voxel R-CNN [3]	$[-75.2\ ,-75.2\ ,-2, 75.2\ , 75.2\ , 4]$	$[0.1\ , 0.1\ , 0.15]$	Adam	1×10^{-2}	1×10^{-3}	32	30
Tab. 6	KITTI [4]	SECOND [9]	$[\ \ 0.0\ ,-40.0\ ,-3, 70.4\ , 40.0\ , 1]$	$[0.16, 0.16, 6.0\ ]$	Adam	3×10^{-3}	1×10^{-2}	32	80
Tab. 6a	Waymo [7]	DSVT [8]	$[-74.88, -74.88, -2, 74.88, 74.88, 4]$	$[0.32, 0.32, 6.0\ ]$	Adam	3×10^{-3}	5×10^{-2}	24	30
Tab. 6b	Waymo [7]	Voxel R-CNN [3]	$[-75.2\ ,-75.2\ ,-2, 75.2\ , 75.2\ , 4]$	$[0.1\ , 0.1\ , 0.15]$	Adam	1×10^{-2}	1×10^{-3}	32	30

Table 1. Complete experimental setup for each table from the main manuscript. We specify the source domain, where $*$ specifies all except the target dataset for our multi-source domain generalization. We report LiDAR point cloud range, voxel size, optimizer parameters (learning rate (LR), weight decay (WD)), batch size (BS) and the number of epochs (E) used for training.

z-alignment	**Method**	**Car**	**Pedestrian**	**Cyclist**	**mAP**
✓	Voxel R-CNN [3]	66.93/28.80	23.39/18.65	19.23/15.76	36.52/21.07
	Voxel R-CNN [3] w/ GBlobs	**80.95/53.98**	**38.33/33.22**	**29.18/25.68**	**49.48/37.62**
✗	Voxel R-CNN [3]	54.61/20.83	10.51/ 7.68	5.88/ 5.12	23.66/11.21
	Voxel R-CNN [3] w/ GBlobs	**80.84/55.05**	**37.93/33.24**	**28.62/24.60**	**49.13/37.63**

Table 2. Influence of z-alignment on detector trained with different input features. We trained Voxel R-CNN [3] on nuScenes [1] using all three classes (Car, Pedestrian, Cyclist) simultaneously and evaluated performance using Average Precision (AP) on Bird's-eye View (BEV) / 3D views at 40 recall positions. Intersection over Union (IoU) thresholds of 0.7, 0.5, and 0.5 were used for Car, Pedestrian, and Cyclist, respectively. We evaluate the performance on KITTI [4], where we report the average AP across all difficulty levels (Easy, Moderate, Hard). Additionally, we provide the mean AP over the three classes. The best value in each category is highlighted in bold.

carlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proc. CVPR*, 2020. 1, 2

[2] Jaeseok Choi, Yeji Song, and Nojun Kwak. Part-Aware Data Augmentation for 3D Object Detection in Point Cloud. In *Proc. IROS*, 2021. 2

[3] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. In *Proc. AAAI*, 2021. 2, 3

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Proc. CVPR*, 2012. 1, 2

[5] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast Encoders for Object Detection from Point Clouds. In *Proc. CVPR*, 2019. 2

[6] Alexander Lehner, Stefano Gasperini, Alvaro Marcos-Ramiro, Michael Schmidt, Mohammad-Ali Nikouei Mahani, Nassir Navab, Benjamin Busam, and Federico Tombari. 3D-VField: Adversarial Augmentation of Point Clouds for Domain Generalization in 3D Object Detection. In *Proc. CVPR*, 2022. 1

[7] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. In *Proc. CVPR*, 2020. 1, 2

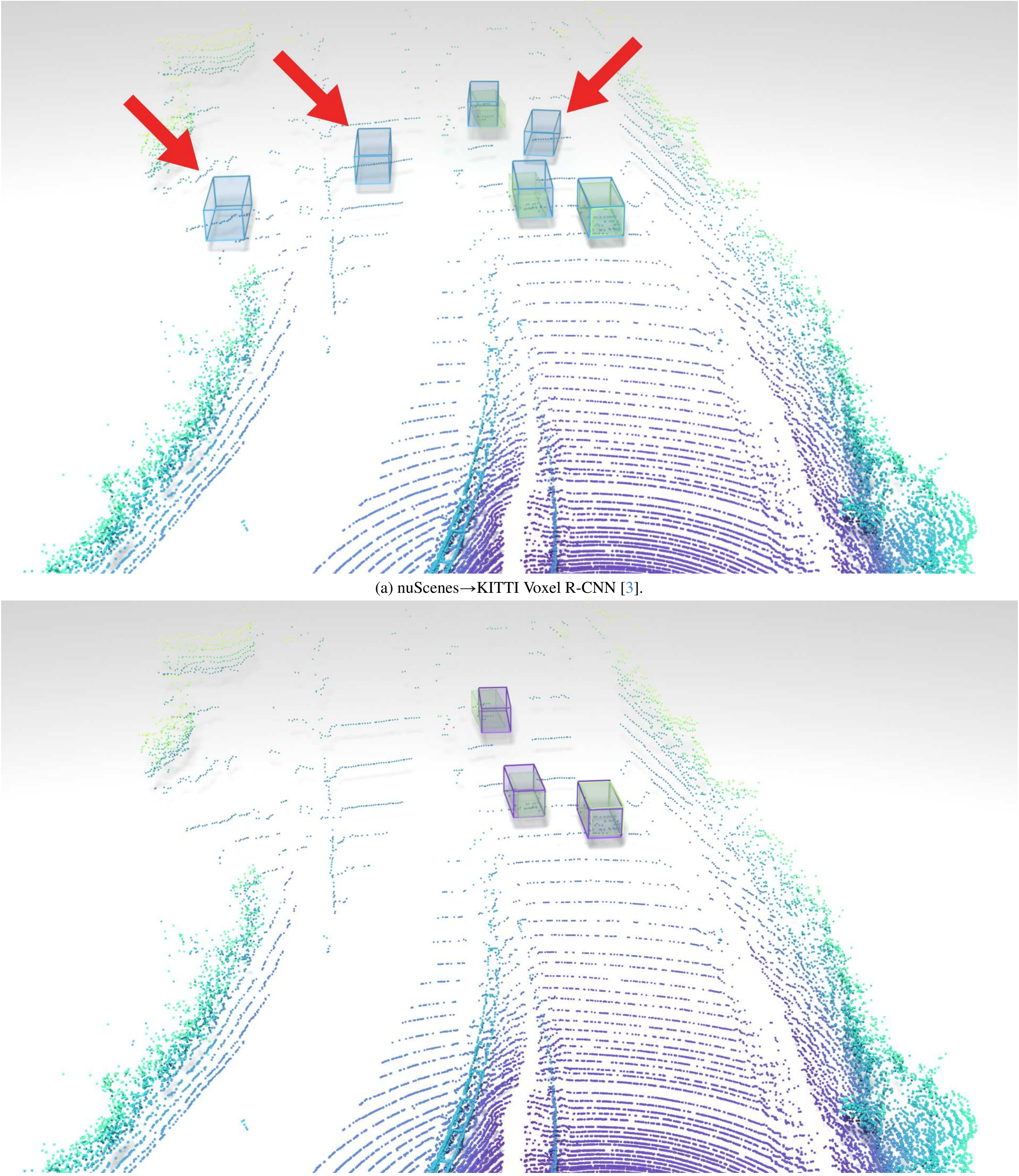
[8] Haiyang Wang, Chen Shi, Shaoshuai Shi, Meng Lei, Sen Wang, Di He, Bernt Schiele, and Liwei Wang. DSVT: Dynamic Sparse Voxel Transformer with Rotated Sets. In *Proc. CVPR*, 2023. 2

[9] Yan Yan, Yuxing Mao, and Bo Li. SECOND: Sparsely Embedded Convolutional Detection. *Sensors*, 18(10):3337, 2018. 1, 2, 4

[10] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D: Self-training for Unsupervised Domain Adaptation on 3D Object Detection. In *Proc. CVPR*, 2021. 1

[11] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. ST3D++: Denoised Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. *TPAMI*, 45(5):6354–6371, 2022. 1

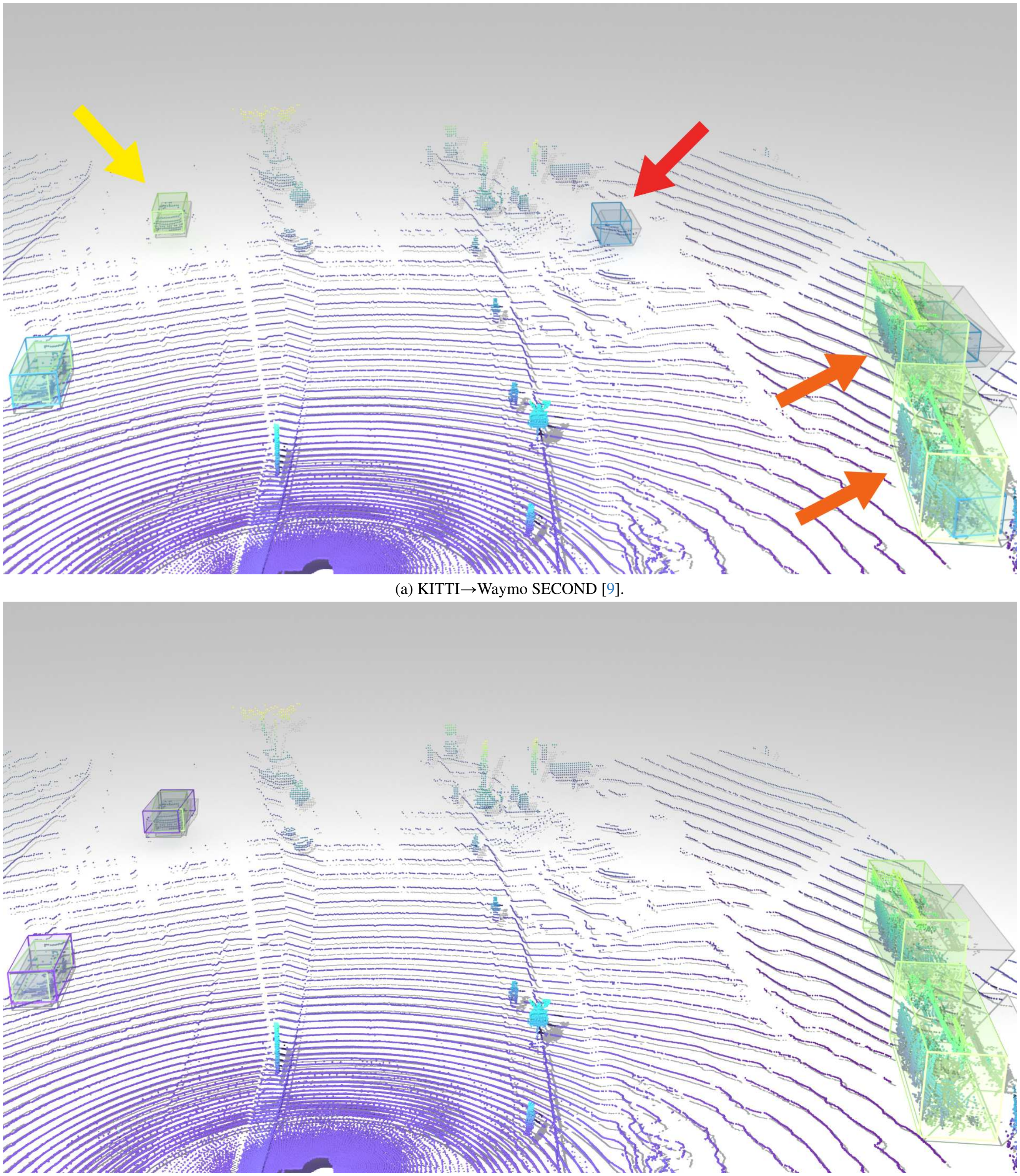
[12] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3D Object Detection and Tracking. In *Proc. CVPR*, 2021. 2



(a) nuScenes→KITTI Voxel R-CNN [3].

(b) nuScenes→KITTI Voxel R-CNN [3] w/ GBlobs.

Figure 2. Qualitative evaluation of Voxel R-CNN [3] on a nuScenes→KITTI benchmark thresholded at 0.5. Ground truth detections are shown in green. Detections from a model trained on standard global input features and our GBlobs are depicted in blue (a) and purple (b), respectively. False positive detections are marked with red arrows. The color of the point cloud represents the height.



(a) KITTI→Waymo SECOND [9].

(b) KITTI→Waymo SECOND [9] w/ GBlobs.

Figure 3. Qualitative evaluation of SECOND [9] on a KITTI→Waymo benchmark thresholded at 0.5. Ground truth detections are shown in green. Detections from a model trained on standard global input features and our GBlobs are depicted in blue (a) and purple (b), respectively. False positive and false negative detections are marked with red and yellow arrow, respetively. Detections which are dubious are markes with orange arrow. The color of the point cloud represents the height.