SAMFusion: Sensor-Adaptive Multimodal Fusion for 3D Object Detection in Adverse Weather (Supplementary Information)

Edoardo Palladin*[1], Roland Dietze*[2], Praveen Narayanan[1], Mario Bijelic[1,3], Felix Heide[1,3]

[1]Torc Robotics [2]University of Stuttgart [3]Princeton University

In this supplemental document, we provide additional information in support of the findings in the main manuscript. Specifically, Sec. 1 provides additional training details, and Sec. 2 lists further information about the SAMFusion network architecture. In Sec. 3, we present additional evaluations on the NuScenes dataset [4] in good weather conditions to compare against state-of-the-art methods. We provide a runtime evaluation in Sec. 4 and include additional qualitative results of SAMFusion in adverse weather in Sec. 5.

Table of Contents

1 Additional Training Details

We train our method SAMFusion in an end-to-end manner for 12 epochs, starting from a pretrained Cascade Mask R-CNN [6] checkpoint trained on the COCO [17] and NuScenes [4] benchmarks. We initialize both camera backbones equally with Cascade Mask R-CNN [6] checkpoints for the camera branch while learning the LiDAR and radar weights from scratch. We train all four modality-specific backbones, the transformer encoder layers, the multi-modal object queries and the transformer decoder layers without freezing any model components.

During training we augment samples and apply random rotations with a range of $r \in [-0.4, 0.4]$, random scaling with a factor of $s \in [0.9, 1.1]$ and random translation with standard deviation 0.5 in x, y and z direction. Additionally, we

* These authors contributed equally to this work.

apply random horizontal flipping following [1]. We train with a class-balanced resampling strategy presented in CBGS [25] to balance the class distribution for the SeeingThroughFog dataset [2]. Following [25] our training uses the Adam optimizer with cyclic learning rate policy, with max learning rate 4×10^{-4}, weight decay 0.0001 and momentum 0.85 to 0.95.

1.1 Additional Training Details on NuScenes

For the experiments on the NuScenes dataset [**nuscenes**] we make use of the official splits, containing 700 and 150 scenes for training and validation respectively, where each sequence contains roughly 40 samples.

We modify our model and remove the gated camera branch due to the lack of this modality in the dataset. Moreover, we don't make use of the stereo-based depth estimation method for the depth-based transformation in the RGB camera branch due to absence of stereo pairs. Instead, we rely on projecting sparse LiDAR points and densifying per-pixel using a depth-completion algorithm [13].

Due to the absence of the gated branch and the stereo-based depth estimation in the structure of our model during NuScenes experiments, we are able to load pre-trained weights from [23], specifically RGB and LiDAR branches and MMPI layers. We train the radar branch and the multi-modal fusion from scratch to initialize object queries, while freezing the RGB branch Cascade Mask R-CNN backbone. We train our method in an end-to-end fashion for 6 epochs, with cyclic learning rate policy, with max learning rate 2×10^{-4}, weight decay 0.0001 and momentum from 0.85 to 0.95.

1.2 Training of Depth Prediction Network

We train the depth estimation method [14] including both RGB stereo images provided in the SeeingThroughFog [2] dataset. These depth maps are later used for the projection of camera features. We supervise the training process with disparity values D determined by equation

$$D = \frac{b \cdot f_B}{d_{LiDAR} \cdot p}, \tag{1}$$

and derived depth values d_{LiDAR} from the LiDAR point cloud as ground truth. The focal length f_B, baseline b and pixel pitch are read from the calibration files of [2].

We train the architecture on the SeeingThroughFog [2] dataset containing 10,046/1,000/1,941 samples for training, validation and testing, respectively. The training is performed for 10 Epochs, applying a cyclic learning rate policy with maximum at $8 \cdot 10^{-6}$. The final depth map d_{CRE} is obtained by rearranging Eq. 1 to

$$d_{CRE} = \frac{b \cdot f_B}{D_{CRE} \cdot p}, \tag{2}$$

with the parameters f_B, b from before and the disparity map output D_{CRE}.

1.3 Training of Baseline Methods

To provide a fair comparison, we apply identical training procedures to all baselines on the same training, validation and testing datasets [2] as SAMFusion. The camera-based monocular methods M3D-RPN [3], PatchNet [19], Gated3D [12], and Stereo-RCNN [15] are implemented from the corresponding open source repositories, with hyperparameters tuned during training on the SeeingThroughFog dataset [2]. In total, we train the mono-LiDAR method SECOND [22] as well as the multi-modal architectures MVXNet [21], BEVFusion [18], DeepInteraction [23] and utilize the MMDetection3D framework [9]. For the best possible results, all methods are initialized from their publicly available checkpoints. Those checkpoints were either based on the NuScenes [4] or Kitti [11] datasets.

2 Additional Network Details

In this section, we provide detailed descriptions of the network architecture.

2.1 Multi-Modal Feature Weighting Module

In Tab. 1 we provide details of our distance-aware weighting network for the multi-modal query initialization, fusing camera and ranging sensor feature maps.

Multi Modal Feature Map Weighting

Layer #	Component	Sigmoid mask	Output Shape
0_a	Convfuser $(\varphi_L^*, \Gamma_{MLP})$	✓	$128 \times 180 \times 180$
0_b	Convfuser $(\varphi_R^*, \Gamma_{MLP})$	✓	$128 \times 180 \times 180$
1	Convfuser $(0_a, 0_b)$	✗	$128 \times 180 \times 180$
Combined feature map φ_{fuse}		**Shape:**	$128 \times 180 \times 180$

Feature map blending module

Layer #	Layer Description	Output Shape
Convfuser	Conv2d (3x3)	$128 \times 180 \times 180$
	GroupNorm (num_groups=16)	
	ReLU	
	Conv2d (3x3)	
	GroupNorm (num_groups=16)	
	ReLU	
	Conv2d (3x3)	
	GroupNorm (num_groups=16)	
	ReLU	

Table 1: Additional architecture details of our feature fusion module, combining all four modalities with distance encoding. We present the overall structure of the feature map blending module (left). This layer is initialized with sigmoid weights Γ_{MLP} and applied to the ranging sensor feature maps φ_L^* and φ_R^* from radar and LiDAR sensors. On the right we present the detailed structure of the Convfuser weighting module.

We encode distance cues into the LR weighting scheme by fusing both enriched feature maps φ_L^* and φ_R^*. The LiDAR feature maps are weighted by a gaussian mask with learned variance while the radar features are weighted by the inverse of the same mask. Accordingly, we prioritize φ_L^* features at close distances while favoring φ_R^* at far distances. Our weighting scheme learns to modulate LiDAR and radar feature maps according to the Γ_{MLP} weights per feature coordinate.

3 Additional Quantitative Evaluation

To further evaluate the SAMFusion architecture in comparison to state-of-the-art multimodal methods on larger, non adverse weather datasets, we evaluate our method on the NuScenes dataset [4], utilizing the NuScenes detection score (NDS) [4] and mean Average Precision (mAP) [10] metrics across 10 classes: car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, and traffic cone.

We summarize our results in Tab. 2. Our method is able to achieve similar performance to state-of-the-art models like DeepInteraction [23] and BEVFusion [18] on non-adverse weather scenarios, showcasing that large detection improvements on adverse weather don't compromise performance on normal conditions.

We highlight that our method focuses on autonomous driving edge cases, tackling situations where the perception task is challenged by adverse weather and long detection ranges. In this regard, the NuScenes dataset showcases only a small pool of samples characterized by harsh weather. Moreover, the detection range of NuScenes is limited to 50 meters, while our method showcases large improvements over previous state-of-the-art methods specifically on ranges above this threshold (50-80 m). The limited range of the dataset samples does not allow SAMFusion to exploit the advantages of longer ranges of the radar modality compared to LiDAR, and due to the absence of the gated camera modality, our method can only rely on these additional sparse radar point clouds compared to other methods. Evaluating the method between 20-50m for the NuScenes dataset we document an improvement of 2.2% AP and 1% NDS. The numbers are shown in Table 3. For the NuScenes dataset it is noted in the literature that many NuScenes radar detections give no information about the height at which they were received, and that radar detections contain objects which are not relevant for the task at hand [20], such as ghost and irrelevant objects, as well as ground detections, all of which would require an additional pre-processing step or data augmentation step to learn to handle. However, training in harsh conditions on the SeeingThroughFog dataset presents plenty of situations where sensor modalities fail, motivating the usage of the robust radar modality. In general, radars return strong echoes from distant metallic objects like cars, but their low angular resolution of 1° in the Seeing Through Fog dataset hinder accurate lateral positioning, impairing long-range detection box regression. Adding radar data improves car recall but lowers the MSE of bounding boxes. Both properties are reflected in the mAP score and therefore, explain diminishing margins in far distances presented in the main document. The NuScenes dataset includes perfect LiDAR and camera data which rarely motivates the necessity of radar data and, therefore, of an additional modality.

4 Runtime Evaluation

Next, we present runtime evaluations. Our full method is bottlenecked by the feature extraction backbones and optimized in our prototype system with one

Method	Modality	mAP ↑	NDS ↑
FUTR3D [7]	CL	64.5	68.3
AMVP [24]	CL	67.1	70.8
AutoAlignV2 [8]	CL	67.1	71.2
TransFusion [1]	CL	67.5	71.3
BEVFusion [16]	CL	67.9	71.0
BEVFusion [18]	CL	68.5	71.4
DeepInteraction [23]	CL	**69.9**	**72.7**
SAMFusion	CLR	68.6	71.7

Table 2: Results on nuScenes dataset validation split.

Method	Modality	mAP ↑	NDS ↑
DeepInteraction [23]	CL	56.6	64.6
SAMFusion	CLR	**58.8**	**65.6**

Table 3: Results on nuScenes dataset validation split for detections on the 20-50 meters range.

Model	Inference time [ms] ↓	Frames per Second ↑
MVXNet [21]	74.0	13.5
BEVFusion [18]	57.4	17.5
DeepInteraction [23]	**48.3**	**20.7**
SAMFusion	70.7	14.3

Table 4: Inference time comparison to existing multi-modal detection methods.

GPU for each feature extractor. Our plain pytorch implementation of SAMFusion without onnx optimization operates at 14.3 FPS on a Nvidia A100 GPU with a batch size of one. Therefore, SAMFusion is matching the sampling rate of the LiDAR sensor operating at 10 Hz and hence provides real-time detection capabilities. We load and preprocess the stereo depth maps for the cross-modal transformation in a parallel step on a separate fifth GPU to optimize computational efficiency during adaptive blending. We present a comparison of the inference time to state-of-the-art sensor fusion reference methods which use less modalities in Tab. 4.

5 Additional Qualitative Evaluation

In this section, we provide additional qualitative examples in Fig. 5, 6 and 7. For all methods, the same confidence threshold of 0.15 is applied and the detections are projected into the left RGB camera image (1920×1080). In comparison to [18, 21, 23], we attain a higher recall and fewer false positives.

In particular, the first row of Fig. 6 demonstrates robust detections in strong fog due to the gated camera and radar sensors, where camera-LiDAR architectures fail due to backscatter and loss of contrast. The improvement is notable, especially for narrow-profile, low-contrast objects such as pedestrians, where SAMFusion yields higher detection accuracy. The bounding box orientations are also better aligned for our method, as can be seen in Fig. 7. This emphasizes the precise geometric understanding caused by the transformations in the adaptive blending module.

Camera-LiDAR approaches such as [1, 23, 5, 18, 16, 21] are able to detect narrow-profiled objects at close distances, but struggle with recognizing objects at far distances, especially in low light conditions. This is addressed by our method, due to the integration of active sensors with night vision abilities. The active illumination of the gated camera offers especially high contrast at long ranges. Fig. 5 validates SAMFusion detection reliability, comparing it to state-of-the-art sensor fusion architectures [18, 21, 23]. We observe multiple missed detections correlating with increased distance for all camera-LiDAR-based methods as shown in the first row of Fig. 5. This results from sparse LiDAR point clouds in far distances with low angular resolution, leading to few points per-object, especially for pedestrians, and decreased signal-to-noise-ratio in passive RGB captures due to limited scene illumination from headlights alone. In particular, light-absorbing dark clothes on pedestrians reduce contrast for camera-based detections even further. Meanwhile, SAMFusion relies on four sensor inputs with improved night vision due to the gated camera equally illuminating the scene and suppressing backscatter efficiently. This reduces the number of random false positives due to increased information density.

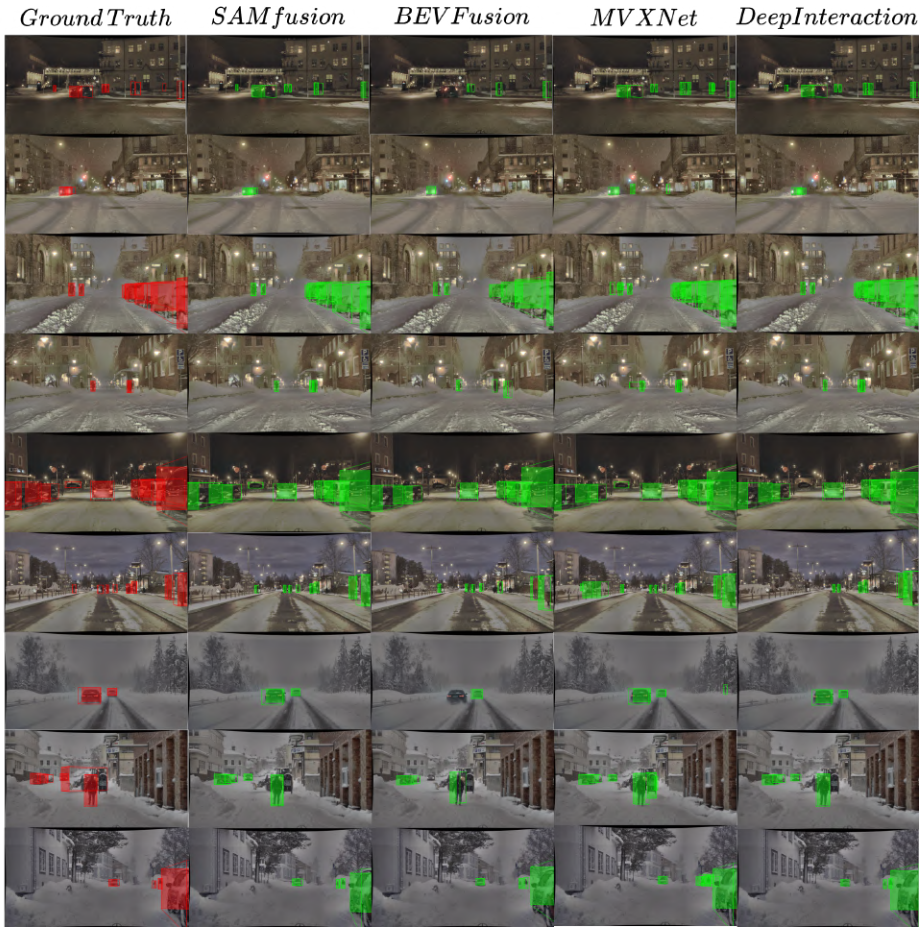

Table 5: Nine additional sequences (rows) with qualitative results compared across reference approaches (columns). The ground truth is illustrated on the left with red bounding boxes, followed by SAMFusion, BEVFusion [16], MVXNet [21] and DeepInteraction [23].

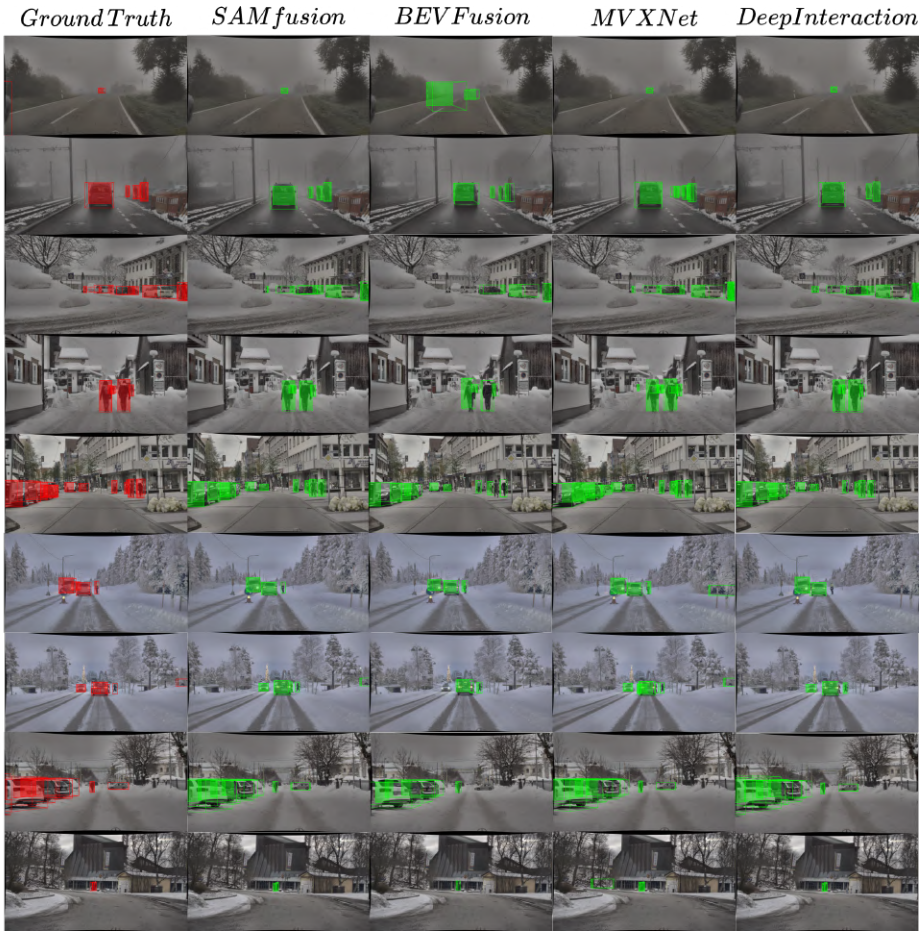

Table 6: Nine additional sequences (rows) with qualitative results compared across reference approaches (columns). The ground truth is illustrated on the left with red bounding boxes, followed by SAMFusion, BEVFusion [16], MVXNet [21] and DeepInteraction [23].

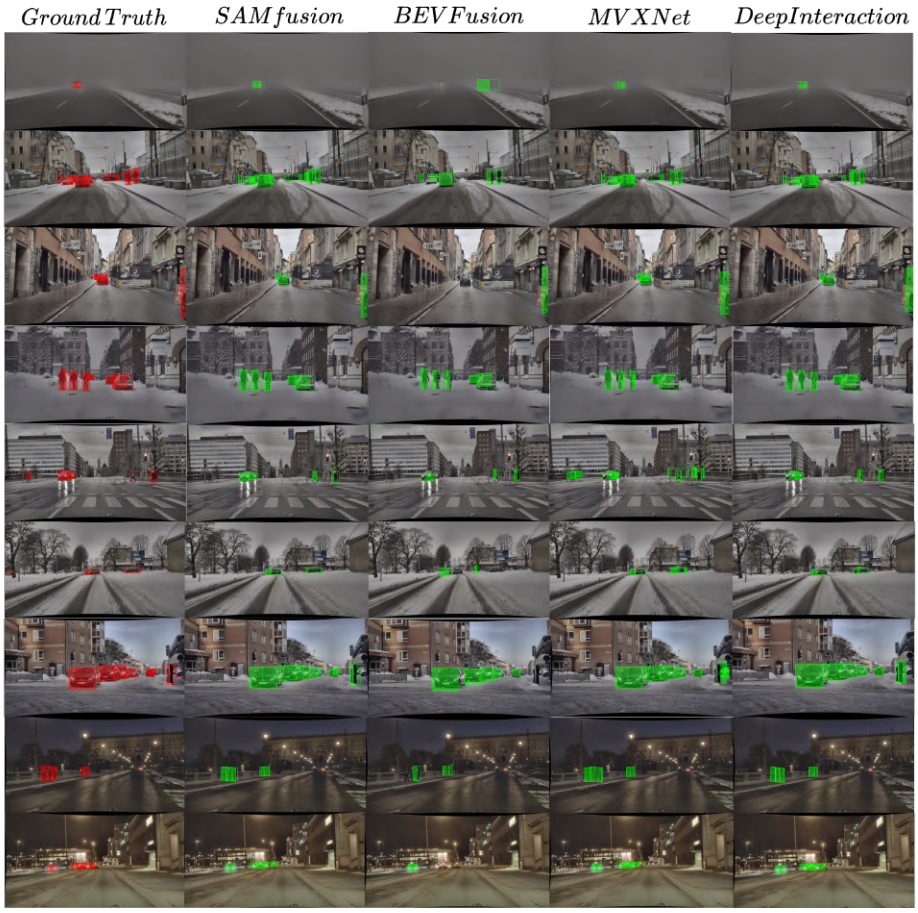

Table 7: Nine additional sequences (rows) with qualitative results compared across reference approaches (columns). The ground truth is illustrated on the left with red bounding boxes, followed by SAMFusion, BEVFusion [16], MVXNet [21] and DeepInteraction [23].

References

[1] Xuyang Bai et al. "Transfusion: Robust lidar-camera fusion for 3d object detection with transformers". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 1090–1099.

[2] Mario Bijelic et al. "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 11682–11692.

[3] Garrick Brazil and Xiaoming Liu. "M3D-RPN: Monocular 3d region proposal network for object detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 9287–9296.

[4] Holger Caesar et al. "Nuscenes: A multimodal dataset for autonomous driving". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 11621–11631.

[5] Hongxiang Cai et al. "BEVFusion4D: Learning LiDAR-Camera Fusion Under Bird's-Eye-View via Cross-Modality Guidance and Temporal Aggregation". In: *arXiv preprint arXiv:2303.17099* (2023).

[6] Zhaowei Cai and Nuno Vasconcelos. "Cascade R-CNN: High quality object detection and instance segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 43.5 (2019), pp. 1483–1498.

[7] Xuanyao Chen et al. "FUTR3D: A unified sensor fusion framework for 3d detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2023, pp. 172–181.

[8] Zehui Chen et al. *AutoAlign: Pixel-Instance Feature Aggregation for Multi-Modal 3D Object Detection*. 2022. arXiv: 2201.06493 [cs.CV].

[9] MMDetection3D Contributors. *MMDetection3D: OpenMMLab next-generation platform for general 3D object detection*. https://github.com/open-mmlab/mmdetection3d. 2020.

[10] Mark Everingham et al. "The Pascal Visual Object Classes (VOC) Challenge". In: *International Journal of Computer Vision* 88 (2010), pp. 303–338.

[11] Andreas Geiger, Philip Lenz, and Raquel Urtasun. "Are we ready for autonomous driving? the kitti vision benchmark suite". In: *2012 IEEE conference on computer vision and pattern recognition*. IEEE. 2012, pp. 3354–3361.

[12] Frank Julca-Aguilar et al. "Gated3d: Monocular 3d object detection from temporal illumination cues". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 2938–2948.

[13] Jason Ku, Ali Harakeh, and Steven L Waslander. "In defense of classical image processing: Fast depth completion on the cpu". In: *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE. 2018, pp. 16–22.

[14] Jiankun Li et al. "Practical stereo matching via cascaded recurrent network with adaptive correlation". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 16263–16272.

[15] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. "Stereo r-cnn based 3d object detection for autonomous driving". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 7644–7652.

[16] Tingting Liang et al. "Bevfusion: A simple and robust lidar-camera fusion framework". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 10421–10434.

[17] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[18] Zhijian Liu et al. "BEVFUSION: Multi-task multi-sensor fusion with unified bird's-eye view representation". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 2774–2781.

[19] Xinzhu Ma et al. "Rethinking pseudo-lidar representation". In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIII 16*. Springer. 2020, pp. 311–327.

[20] Felix Nobis et al. *A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection*. 2020. arXiv: 2005.07431 [cs.CV].

[21] Vishwanath A Sindagi, Yin Zhou, and Oncel Tuzel. "MVX-NET: Multimodal voxelnet for 3d object detection". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 7276–7282.

[22] Yan Yan, Yuxing Mao, and Bo Li. "Second: Sparsely embedded convolutional detection". In: *Sensors* 18.10 (2018), p. 3337.

[23] Zeyu Yang et al. "DeepInteraction: 3d object detection via modality interaction". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 1992–2005.

[24] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. "Multimodal virtual point 3d detection". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 16494–16507.

[25] Benjin Zhu et al. "Class-balanced grouping and sampling for point cloud 3d object detection". In: *arXiv preprint arXiv:1908.09492* (2019).