Local LiDAR Geometry for Improved LiDAR Sensor Placements

Dušan Malić[1,2]   Christian Fruhwirth-Reisinger[1,2]   Alexander Prutsch[1]
Wei Lin[3]   Samuel Schulter[4,*]   Horst Possegger[1,2]

[1]Institute of Visual Computing, Graz University of Technology
[2]Christian Doppler Laboratory for Embedded Machine Learning
[3]Institute for Machine Learning, Johannes Kepler University Linz   [4]Amazon

{dusan.malic, reisinger, alexander.prutsch, possegger}@tugraz.at

Abstract

This technical report outlines the top-ranking solution for RoboSense 2025: Track 3, achieving state-of-the-art performance on 3D object detection under various sensor placements. Our submission utilizes GBlobs, a local point cloud feature descriptor specifically designed to enhance model generalization across diverse LiDAR configurations. Current LiDAR-based 3D detectors often suffer from a "geometric shortcut" when trained on conventional global features (i.e., absolute Cartesian coordinates). This introduces a position bias that causes models to primarily rely on absolute object position rather than distinguishing shape and appearance characteristics. Although effective for in-domain data, this shortcut severely limits generalization when encountering different point distributions, such as those resulting from varying sensor placements. By using GBlobs as network input features, we effectively circumvent this geometric shortcut, compelling the network to learn robust, object-centric representations. This approach significantly enhances the model's ability to generalize, resulting in the exceptional performance demonstrated in this challenge.

1. Introduction

The majority of LiDAR-based 3D object detection architectures [1, 23, 26, 27, 29, 35, 36] rely on global input features, specifically, the absolute Cartesian coordinates of the points. Because of their ease of use and strong in-domain performance, global features have become the standard input representation for most leading detection models [7, 15, 20, 22, 25, 28]. However, these models suffer from a geometric shortcut [30] exhibiting a significant bias toward object location rather than local characteristics like shape or appearance [24]. Consequently, this significantly limits their

ability to generalize to environments with different object location distributions, such as those induced by different sensor placements [5, 6, 16–19, 34].

Our contribution to the RoboSense 2025 challenge demonstrates that leveraging local point cloud geometry can substantially boost model generalization across diverse sensor configurations. Specifically, we employ GBlobs [24], a novel representation that treats local neighborhoods as Gaussian blobs, defined by their mean and covariance matrices. This formulation enables the model's encoding to be independent of the object's absolute position, effectively removing the geometric shortcut and directing the model's learning toward localized attributes, such as the shape and appearance of the objects of interest.

Calculating GBlobs requires a minimum of three points in close proximity. Due to the inherent sparsity of LiDAR data, this requirement frequently leads to degeneracy in the far range, where local neighborhoods often lack sufficient points. To effectively mitigate this issue, we employ a hybrid detection strategy: a secondary model is trained using conventional global Cartesian coordinates and is deployed for far-range predictions. Both the primary (GBlobs-based) and secondary (global-coordinate-based) models are independently processed with Test-Time Augmentation (TTA). We then revert the augmentations and apply Non-Maximum Suppression (NMS) to the output of each model separately. Finally, the resulting predictions are spatially fused using a distance-based threshold: predictions from the GBlobs-trained model are utilized for the near-range (up to 30m), while predictions from the global-coordinate model are used exclusively beyond this threshold.

Our approach ranked 1st in the RoboSense Challenge 2025 Track 3: Sensor Placement. This work aims to highlight the potential of local geometric features to significantly enhance model generalization, a topic we believe remains critically underexplored. This report not only confirms this potential but also delivers a detailed explanation and analysis

---

of the techniques employed.

2. Methodology

A significant challenge towards generalizable 3D object detection across different sensor placements is the susceptibility of deep learning models to geometric shortcuts. When networks are trained directly on absolute Cartesian coordinates, they often learn to over-rely on an object's absolute position within the scene, rather than its inherent geometric and structural properties. This reliance on global coordinates hinders the model's robustness and generalization to different sensor placements.

To address geometric shortcuts, we decouple the network's learning from absolute object positions. Instead of using global Cartesian coordinates, we encode the local geometric information of the point cloud. More precisely, given an input point cloud $X = \{\boldsymbol{p}_j = (x, y, z)\}_{j=1}^{M}$ of M points specified by their global 3D Cartesian coordinates, we represent a local neighborhood of N points as a GBlob [24], characterized by its mean $\boldsymbol{\mu}$ and covariance Σ, denoted as $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, where

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{p}_i, \quad \text{and} \qquad (1)$$

$$\Sigma = \frac{1}{N} \sum_{i=1}^{N} (\boldsymbol{p}_i - \boldsymbol{\mu})(\boldsymbol{p}_i - \boldsymbol{\mu})^{\top}. \qquad (2)$$

By transforming the input from absolute coordinates to GBlobs, the network is compelled to learn from the shape and local geometric structure of the object, thereby mitigating the geometric shortcut problem.

A major challenge in long-range object detection from LiDAR data is the sparse nature of point clouds. In these sparse regions, a local neighborhood often contains only a single point, which makes it impossible to compute a meaningful covariance matrix. As a result, our GBlobs representation degenerates into a mean-only feature, severely limiting the effectiveness of local encoding and hindering the performance of the detector.

To overcome this limitation, we introduce a dual-model approach. We augment the GBlobs-based detector with a parallel detector that directly processes standard Cartesian coordinates. This parallel architecture ensures robust performance even when the local encoding of the GBlobs model is compromised by point sparsity.

We enhance the robustness of both detectors using Test-Time Augmentation (TTA). We augment the input point cloud with various transformations, including translation, rotation, and scaling. Detections are then inferred from these augmented inputs and transformed back to the original coordinate system. Non-Maximum Suppression (NMS) is then applied to aggregate the predictions and remove redundant or noisy estimates.

For the final prediction, we fuse the outputs from both detectors using a simple range-based thresholding scheme. We define a distance threshold, $\delta_d = 30$ meters, to combine the predictions: detections from the GBlobs model are used for objects within this range, while predictions from the Cartesian-coordinate model are used for all objects beyond it. We summarize our method in Fig. 1. This straightforward yet effective fusion strategy leverages the strengths of each model, maintaining high performance across the full range of point densities and distances.

3. Experiments

In the following section we provide the detailed experimental setup (Sec. 3.1), implementation details (Sec. 3.2) and our ablation studies (Sec. 3.3).

3.1. Experimental Setup

Dataset We use the official data provided by the *RoboSense Challenge 2025* [9] held at IROS 2025. This competition builds upon the legacy of the *RoboDepth Challenge 2023* [12, 13] at ICRA 2023 and the *RoboDrive Challenge 2024* [14, 33] at ICRA 2024, continuing the collective effort to advance robust and scalable robot perception. Each track in this competition is grounded on an established benchmark designed for evaluating real-world robustness and generalization [2, 4, 10, 11, 20, 21, 31, 32]. Specifically, this task is built upon the **Place3D** dataset [20] in **Track 3**, which provides a standardized foundation for benchmarking performance under challenging conditions such as cross-domain shifts, sensor variability, and multi-modal alignment. Place3D comprises 11 LiDAR layouts in total – seven baseline layouts inspired by production AV rigs and four layouts obtained via sensor placement optimization.

For this track, the organizers provided a synthetic LiDAR dataset [20] generated using CARLA [3]. This dataset comprises $18\,000$ samples, partitioned into a training, validation, and test split of 5000, 3000, and $10\,000$ samples, respectively. A key challenge of this track was overcoming the domain shift induced by different sensor placements. The public splits, used for training and validation, featured four known sensor placements. In contrast, the test split introduced six completely unseen and distinct placements. As depicted in Fig. 2, these configuration changes translate into significant differences in the resulting point clouds. To ensure a fair evaluation of robustness and generalization, the ground truth labels and sensor positions for the training and validation sets were released; however, the ground truth labels and the corresponding sensor positions for the test set remained confidential.
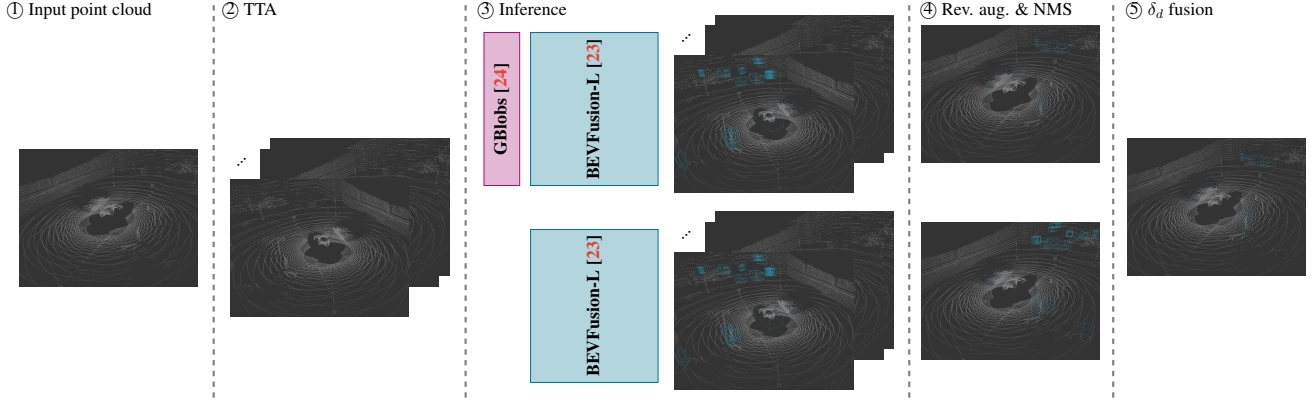
Figure 1. Our method first takes an input point cloud ① and generates a batch of randomly augmented frames ②. This batch is then inferred ③ by two models: the BEVFusion-L [23] baseline model and the same model trained with GBlobs [24]. The augmented predictions are then reversed (de-augmented) and combined using Non-Maximum Suppression (NMS) ④. Finally, the predictions from the two models are fused ⑤ via δ_t fusion, where the GBlobs-trained predictions are used up to the δ_t distance threshold, and the standard Cartesian model predictions are used beyond it. Best viewed on a monitor and zoomed in for detail.

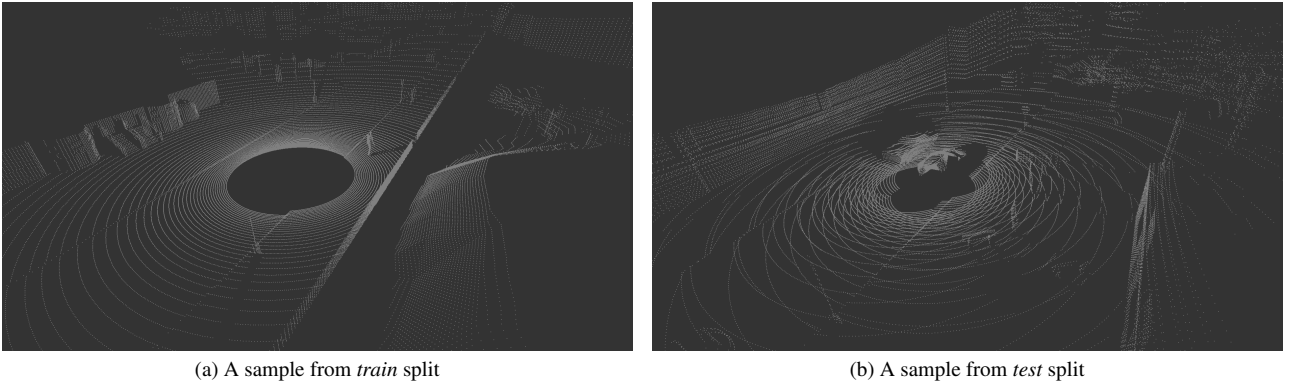| (a) A sample from *train* split | (b) A sample from *test* split |

Figure 2. Exemplary LiDAR frames from the RoboSense Challenge 2025, Track 3: Sensor Placement dataset.

Metrics The final ranking of submissions is determined by the mean Average Precision (mAP). To ensure robustness against statistically minor variations, a secondary metric is utilized exclusively for breaking ties. Specifically, if the absolute difference in mAP between two submissions is within 0.01 (one percentage point), the tie is resolved using the NuScenes Detection Score (NDS). Thus, mAP establishes the principal rank order, with NDS serving as the decisive criterion when primary performance metrics are nearly equivalent.

Detector We follow the challenge baseline and utilize BEVFusion-L [23] as our detector. Unlike the original BEVFusion, which fuses features from both camera and LiDAR sensors, BEVFusion-L is a variant that operates as a LiDAR-only detector. This model leverages a spatial cross-attention mechanism to transform raw point cloud features into a bird's-eye-view (BEV) representation, which is then used for 3D object detection.

3.2. Implementation Details

Our implementation is publicly available on GitHub at github.com/malicd/GBlobs[1] for transparency and reproducibility. We use BEVFusion-L with a combined training and validation dataset, employing a class-balanced sampling [37] strategy to address class imbalance. The model is trained for 90 epochs using the Adam optimizer [8] and a cyclical learning rate scheduler with a maximum learning rate of 0.001. We define the LiDAR range as $[-108.0, -108.0, -5.0, 108.0, 108.0, 3.0]$ meters, using a voxel size of $[0.075, 0.075, 0.2]$ meters and accumulate 10 consecutive frames for both training and inference. To enhance model generalization, we incorporate standard augmentation techniques, including ground truth sampling [35], random rotation, translation, and scaling. We disable ground truth sampling for the final 5 epochs to stabilize training. To optimize performance for the mAP metric, we down-weight

---

[1]Our implementation is based on OpenPCDet.

the size and rotation head losses by a factor of 0.05 and do not predict object velocity.

For inference, we employ Test-Time Augmentation (TTA), augmenting each input frame 10 times for both the global and Gblobs detectors with random $\pm 60°$ rotation, x and y axis flip, and $[0.95, 1.05]$ scaling. The augmented frames are processed to generate predictions, and Non-Maximum Suppression (NMS) is applied to consolidate detections for each individual detector. Finally, we merge the predictions from both detectors based on a distance threshold. We select all predictions from the Gblobs detector within $\delta_d = 30$ meters and combine them with predictions from the global detector that are beyond this threshold.

3.3. Ablation Study

We analyze the impact of our core design choices: utilizing GBlobs [24] as model inputs and employing Test-Time Augmentation (TTA) to enhance prediction robustness. All models are trained on the training split of the RoboSense 2025 Challenge: Track 3 dataset and evaluated on the corresponding validation split. Note, however, that the dataset's test split (used for the official challenge) contains significantly different sensor placements than across the training and validation data. Consequently, the following ablation findings on the improved generalization capabilities are even more pronounced on the official test set.

The baseline model for our ablation study uses the standard (global Cartesian) input representation without TTA. Models trained with GBlobs circumvent the geometric shortcut, which allows them to generalize better across different sensor placements. This benefit is clearly demonstrated in the second row of Tab. 1. Replacing the standard input with GBlobs, without any other modifications, yields a significant performance gain. Specifically, the model trained with GBlobs improves the baseline performance by 1.67 AP points. To further boost the model's accuracy, we investigate the contribution of Test-Time Augmentation (TTA). Applying TTA to the GBlobs-based model provides an additional improvement of 1.12 AP points. This confirms TTA's role in stabilizing predictions and further increasing robustness. Overall, the synergistic effect of both design choices is substantial. The combination of GBlobs input and TTA improves the baseline model's performance by a total of 2.79 AP points, demonstrating the effectiveness of our proposed approach for the RoboSense 2025 challenge.

As detailed in Sec. 2, in sparse LiDAR regions (*e.g.*, far-range), a local neighborhood often contains only a single point. This makes it impossible to compute a covariance matrix as per Eq. (2). Consequently, our GBlobs representation degenerates into a mean-only feature, severely limiting the effectiveness of local encoding and hindering the performance of the detector. To mitigate this, we employ a hybrid approach: we utilize GBlobs predictions up to a distance
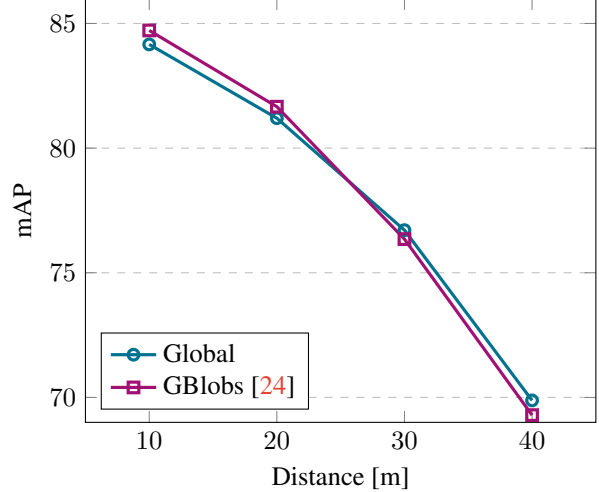

Figure 3. We evaluate two BEVFusion-L [23] models: a standard model trained with global input features (—⊙—) and a model trained with GBlobs [24] (—□—). The x-axis represents the distance cut-off, where predictions and ground truth below this value are ignored during evaluation. The y-axis reports the mean Average Precision (mAP) computed across all detection classes on the validation split. Both models were trained on the training split of the RoboSense 2025 Challenge: Track 3 dataset.

threshold δ_d and use predictions from the standard global-feature baseline model beyond this threshold. To determine an optimal δ_d, we performed an ablation study by evaluating the performance while gradually removing objects up to a certain distance. In Fig. 3, we observe that the GBlobs-based model outperforms the baseline up to approximately 28 meters. However, as sparsity increases with distance, the GBlobs model's performance then rapidly deteriorates. Based on this trade-off, we select δ_d to be 30 meters for the final evaluation, using GBlobs-based detections within this range and baseline predictions for all objects beyond it.

4. Conclusion

Our solution for the RoboSense 2025: Track 3 competition showcases a robust approach to 3D object detection with varied sensor placements. By leveraging GBlobs, we demonstrate a generalizable model that maintains high robustness across diverse sensor configurations. This work not only provides a high-performing solution but also identifies a key direction for the field: improving model generalization with local features, especially in sparse areas. We hope our methodology will serve as a strong foundation for future research in this domain.

References

[1] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxel-NeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking. In *Proc. CVPR*, 2023.

Method	GBlobs	TTA	car	truck	bus	motorcycle	bicycle	pedestrian	mAP
BEVFusion-L	✗	✗	0.9260	0.9013	0.8705	0.9021	0.8320	0.9020	0.8790
BEVFusion-L	✓	✗	0.9244	0.9194	0.8726	0.9171	0.8400	0.9004	0.8957
BEVFusion-L	✓	✓	**0.9329**	**0.9252**	**0.8758**	**0.9199**	**0.8416**	**0.9460**	**0.9069**

Table 1. We evaluate the contribution of key components, including *GBlobs* [24] and *Test-Time Augmentation (TTA)*, using the BEVFusion-L [23] baseline. Results are reported as per-class Average Precision (AP) and mean Average Precision (mAP) on the *validation set*. **Bold** indicates the best performance.

[2] Meng Chu, Zhedong Zheng, Wei Ji, Tingyu Wang, and Tat-Seng Chua. Towards natural language-guided drones: GeoText-1652 benchmark with spatial relation matching. In *European Conference on Computer Vision*, pages 213–231, 2024.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An Open Urban Driving Simulator. In *Proc. CoRL*, 2017.

[4] Zeying Gong et al. From cognition to precognition: A future-aware framework for social navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9122–9129, 2025.

[5] Xiaoshuai Hao et al. Is your HD map constructor reliable under sensor corruptions? In *Advances in Neural Information Processing Systems*, pages 22441–22482, 2024.

[6] Xiaoshuai Hao et al. SafeMap: Robust HD map construction from incomplete observations. In *International Conference on Machine Learning*, pages 22091–22102. PMLR, 2025.

[7] Tianshuai Hu et al. Vision-language-action models for autonomous driving: Past, present, and future. *arXiv preprint arXiv:2512.16760*, 2025.

[8] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *Proc. ICLR*, 2015.

[9] Lingdong Kong, Shaoyuan Xie, Zeying Gong, Ye Li, Meng Chu, Ao Liang, Yuhao Dong, Tianshuai Hu, Ronghe Qiu, Rong Li, Hanjiang Hu, Dongyue Lu, Wei Yin, Wenhao Ding, Linfeng Li, Hang Song, Wenwei Zhang, Yuexin Ma, Junwei Liang, Zhedong Zheng, Lai Xing Ng, Benoit R. Cottereau, Wei Tsang Ooi, Ziwei Liu, Zhanpeng Zhang, Weichao Qiu, Wei Zhang, Ji Ao, Jiangpeng Zheng, Siyu Wang, Guang Yang, Zihao Zhang, Yu Zhong, Enzhu Gao, Xinhan Zheng, Xueting Wang, Shouming Li, Yunkai Gao, Siming Lan, Mingfei Han, Xing Hu, Dusan Malic, Christian Fruhwirth-Reisinger, Alexander Prutsch, Wei Lin, Samuel Schulter, Horst Possegger, Linfeng Li, Jian Zhao, Zepeng Yang, Yuhang Song, Bojun Lin, Tianle Zhang, Yuchen Yuan, Chi Zhang, Xuelong Li, Youngseok Kim, Sihwan Hwang, Hyeonjun Jeong, Aodi Wu, Xubo Luo, Erjia Xiao, Lingfeng Zhang, Yingbo Tang, Hao Cheng, Renjing Xu, Wenbo Ding, Lei Zhou, Long Chen, Hangjun Ye, Xiaoshuai Hao, Shuangzhi Li, Junlong Shen, Xingyu Li, Hao Ruan, Jinliang Lin, Zhiming Luo, Yu Zang, Cheng Wang, Hanshi Wang, Xijie Gong, Yixiang Yang, Qianli Ma, Zhipeng Zhang, Wenxiang Shi, Jingmeng Zhou, Weijun Zeng, Kexin Xu, Yuchen Zhang, Haoxiang Fu, Ruibin Hu, Yanbiao Ma, Xiyan Feng, Wenbo Zhang, Lu Zhang, Yunzhi Zhuge, Huchuan Lu, You He, Seungjun Yu, Junsung Park, Youngsun Lim, Hyunjung Shim, Faduo Liang, Zihang Wang, Yiming Peng, Guanyu Zong, Xu Li, Binghao Wang, Hao Wei, Yongxin Ma, Yunke Shi, Shuaipeng Liu, Dong Kong, Yongchun Lin, Huitong Yang, Liang Lei, Haoang Li, Xinliang Zhang, Zhiyong Wang, Xiaofeng Wang, Yuxia Fu, Yadan Luo, Djamahl Etchegaray, Yang Li, Congfei Li, Yuxiang Sun, Wenkai Zhu, Wang Xu, Linru Li, Longjie Liao, Jun Yan, Benwu Wang, Xueliang Ren, Xiaoyu Yue, Jixian Zheng, Jinfeng Wu, Shurui Qin, Wei Cong, and Yao He. The RoboSense challenge: Sense anything, navigate anywhere, adapt across platforms. https://robosense2025.github.io, 2025.

[10] Lingdong Kong et al. Rethinking range view representation for LiDAR segmentation. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 228–240, 2023.

[11] Lingdong Kong et al. Robo3D: Towards robust and reliable 3D perception against corruptions. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 19994–20006, 2023.

[12] Lingdong Kong et al. RoboDepth: Robust out-of-distribution depth estimation under corruptions. In *Advances in Neural Information Processing Systems*, pages 21298–21342, 2023.

[13] Lingdong Kong et al. The RoboDepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023.

[14] Lingdong Kong et al. The RoboDrive challenge: Drive anytime anywhere in any condition. *arXiv preprint arXiv:2405.08816*, 2024.

[15] Lingdong Kong et al. 3D and 4D world modeling: A survey. *arXiv preprint arXiv:2509.07996*, 2025.

[16] Rong Li, Yuhao Dong, Tianshuai Hu, et al. 3EED: Ground everything everywhere in 3D. In *Advances in Neural Information Processing Systems*, 2025.

[17] Rong Li et al. SeeGround: See and ground for zero-shot open-vocabulary 3D visual grounding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3707–3717, 2025.

[18] Ye Li et al. Influence of camera-LiDAR configuration on 3D object detection for autonomous driving. In *IEEE International Conference on Robotics and Automation*, pages 9018–9025, 2024.

[19] Ye Li et al. Optimizing LiDAR placements for robust driving perception in adverse conditions. *arXiv preprint arXiv:2403.17009*, 2024.

[20] Ye Li et al. Is your LiDAR placement optimized for 3D scene understanding? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 34980–35017, 2024.

[21] Ao Liang et al. Perspective-invariant 3D object detection. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 27725–27738, 2025.

[22] Ao Liang et al. WorldLens: Full-spectrum evaluations of driving world models in real world. *arXiv preprint arXiv:2512.10958*, 2025.

[23] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation. In *Proc. ICRA*, 2023.

[24] Dušan Malić, Christian Fruhwirth-Reisinger, Samuel Schulter, and Horst Possegger. GBlobs: Explicit Local Structure via Gaussian Blobs for Improved Cross-Domain LiDAR-based 3D Object Detection. In *Proc. CVPR*, 2025.

[25] Dominik Scheuble, Chenyang Lei, Seung-Hwan Baek, Mario Bijelic, and Felix Heide. Polarization Wavefront Lidar: Learning Large Scene Reconstruction from Polarized Wavefronts. In *Proc. CVPR*, 2024.

[26] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. In *Proc. CVPR*, 2019.

[27] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. *IJCV*, 131(2):531–551, 2023.

[28] Song Wang et al. Forging spatial intelligence: A roadmap of multi-modal data pre-training for autonomous systems. *arXiv preprint arXiv:2512.24385*, 2025.

[29] Xuzhi Wang et al. NUC-Net: Non-uniform cylindrical partition network for efficient LiDAR semantic segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 35(9):9090–9104, 2025.

[30] Xiaoyang Wu, Daniel DeTone, Duncan Frost, Tianwei Shen, Chris Xie, Nan Yang, Jakob Engel, Richard Newcombe, Hengshuang Zhao, and Julian Straub. Sonata: Self-Supervised Learning of Reliable Point Representations. In *Proc. CVPR*, 2025.

[31] Shaoyuan Xie et al. RoboBEV: Towards robust bird's eye view perception under corruptions. *arXiv preprint arXiv:2304.06719*, 2023.

[32] Shaoyuan Xie et al. Are VLMs ready for autonomous driving? an empirical study from the reliability, data, and metric perspectives. In *IEEE/CVF International Conference on Computer Vision*, pages 6585–6597, 2025.

[33] Shaoyuan Xie et al. Benchmarking and improving bird's eye view perception robustness in autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5):3878–3894, 2025.

[34] Xiang Xu et al. FRNet: Frustum-range networks for scalable LiDAR segmentation. *IEEE Transactions on Image Processing*, 34:2173–2186, 2025.

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

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

[37] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection. *arXiv CoRR*, abs/1908.09492, 2019.