PlaceRecover: A Transformer-based Point Cloud Recovery Network with Implicit Neural Representations for Robust LiDAR Placement Adaptation

Zihang Wang[1,2], Yiming Peng[1], Guanyu Zong[1], Xu Li[1], Binghao Wang[3], Hao Wei[4], Yongxin Ma[5], Yunke Shi[1,2], Shuaipeng Liu[1,2], Dong Kong[6]

[1]Southeast University, [2]Ruimove.AI, [3]Jiangsu University of Science and Technology
[4]Zhejiang University, [5]Shandong University, [6]Shandong University of Science and Technology

{wzhanggg, 220243715, guanyuzong, xuliseu}@seu.edu.cn
232210305227@stu.just.edu.cn, isweihao@zju.edu.cn, yxma@mail.sdu.edu.cn
yunkeshi@seu.edu.cn, shuaipengliu@seu.edu.cn, kd.trans@sdust.edu.cn

Abstract

This technical report presents PlaceRecover, a Transformer-based point cloud recovery network with implicit neural representations (INR) for robust LiDAR sensor placement adaptation. The proposed method optimizes LiDAR sensor placements by leveraging the INR-PointTransformer encoder for feature extraction to capture both local structures and global implicit representations of point clouds. A Layout-Aware PointTransformer decoder, enhanced by the layout-context Block, is employed to recover point clouds from different sensor placements, which accounts for the global context and ensures robust recovery across various environmental conditions and sensor configurations. We demonstrate the effectiveness of the method by combining PlaceRecover with BEVFusion-L for 3D object detection on point clouds from hybrid sensor placements, achieving a MAP score of 72.4. This represents a significant improvement of 11.9 points over the baseline method BEVFusion-L, highlighting the effectiveness of the proposed approach in adapting to diverse sensor configurations and environmental perturbations.

Keywords: *LiDAR placements, robust 3D detection, point cloud recovery, implicit neural representation.*

1. Introduction

3D object detection is crucial for the safety and efficiency of autonomous systems [13, 19, 32, 36, 47, 52]. Place3D [38] has shown that the performance of 3D object detection significantly depends on sensor configurations, particularly the placement of LiDAR sensors. However, most existing LiDAR-based perception methods [6, 35] assume fixed sensor configurations, overlooking the impact of sensor placement on point cloud recovery and object detection performance [39].

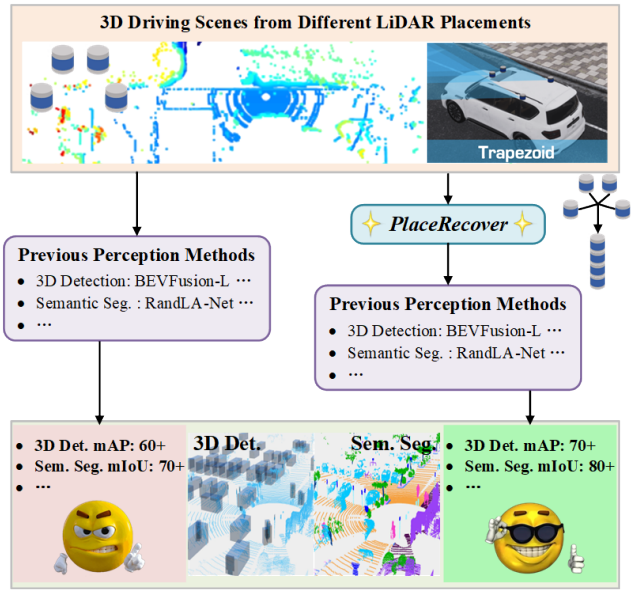

Figure 1. The performance of the previous point cloud perception method with PlaceRecover shows significant improvement across different point cloud placements.

While progress has been made in 3D object detection and semantic segmentation, current methods still have limitations. Most state-of-the-art approaches focus on object detection [42, 60, 63, 64] and semantic segmentation [1, 10, 12, 21, 40, 45, 46, 54, 59, 61, 65], assuming fixed sensor layouts. These methods perform poorly under non-ideal configurations, especially unconventional sensor layouts. Existing models often struggle with adapting to different sensor

configurations, leading to degraded point cloud quality and reduced detection accuracy. Additionally, spatial and contextual information, crucial for recovering point clouds from diverse layouts, is often neglected. Place3D [38] has shown that evaluating the impact of sensor placement on 3D scene understanding is a significant challenge, complicating the determination of optimal sensor configurations for better performance.

To address these challenges, we propose PlaceRecover, a Transformer-based network for recovering point clouds from various LiDAR sensor layouts. PlaceRecover utilizes the INR-PointTransformer encoder for feature extraction, ensuring both geometric consistency and semantic coherence in point cloud recovery. The PointTransformer V3 (PTv3) [55] encoder extracts spatial features, while the network employs implicit neural representation (INR) [7] to learn implicit 3D representations, enhancing recovery quality and robustness. A layout-aware PointTransformer decoder, enhanced by the layout-context block, is used to recover point clouds from different sensor placements, considering global context and ensuring robust recovery across diverse environmental conditions and sensor configurations, as shown in Figure 1.

Our approach secured fourth place in Track 3 of the 2025 RoboSense Challenge, demonstrating its effectiveness in recovering point clouds across various LiDAR sensor placements. This achievement highlights the robustness of our method in adapting to complex sensor configurations and environmental challenges, marking further progress in point cloud recovery for autonomous systems.

2. Related Work

LiDAR Sensing and 3D Perception LiDAR-based sensing plays a critical role in the 3D scene understanding tasks, including 3D object detection and semantic segmentation, which are essential for autonomous driving and robotics applications [2, 4, 5, 24, 33, 34, 43, 44, 62]. Recent advances have focused on enhancing the accuracy and robustness of these tasks under ideal conditions. Methods such as [20, 21, 27, 28, 35, 37, 42, 47, 62] have explored the use of deep learning models, including convolutional neural networks (CNNs) and transformers, to process LiDAR point clouds for semantic segmentation and object detection [15–17, 25, 48, 53, 56]. However, most of these methods rely on static or fixed LiDAR sensor configurations and do not account for the impact of sensor placement variations [38].

Sensor Placement Optimization The importance of optimizing LiDAR sensor placements has only recently been recognized, as it can significantly affect the quality of 3D perception. Previous public datasets, such as [3] and [13], explored sensor configurations but typically considered only static placements, failing to optimize for varying conditions. Place3D [38] introduced the concept of evaluating and optimizing LiDAR placements to enhance 3D perception under diverse environmental conditions. It demonstrated that sensor placements can be optimized using a novel metric, M-SOG, and optimized configurations lead to better performance in both 3D detection and segmentation tasks under adverse conditions.

Implicit Neural Representations (INR) and Transformers Recent advancements have explored combining deep learning models like transformers with implicit neural representations (INR) to model complex 3D data. INR has shown promise in various applications, including point cloud reconstruction [22, 49] and image generation [18, 51], where it provides continuous representations of 3D space. The combination of INR with transformer models, such as in the INR-Transformer [7], allows for dynamic feature extraction and spatial understanding, further improving point cloud recovery from different sensor layouts.

3. Methodology

3.1. Overall Pipeline

Figure 2 illustrates the overall pipeline of our PlaceRecover network. The network integrates the global geometric modeling capability of the PointTransformer V3, the implicit spatial relationship modeling features from INR, and layout contextual information, enabling robust point cloud recovery across various LiDAR placements.

Specifically, given a point cloud input $P_c \in \mathbb{R}^{N\times 4}$, where N is the number of points in the point cloud and each point is represented by four features including its 3D coordinates (x, y, z) and intensity i, the point cloud is first passed through a convolutional network to extract features. This process produces a feature map $F_0 \in \mathbb{R}^{N\times C}$, where C denotes the number of feature channels. Subsequently, these features are refined through our carefully designed PlaceRecover restoration encoder-decoder architecture.

The encoder consists of three INR-PointTransformer V3 modules, each integrating self-attention mechanisms, spatial feature learning, and implicit neural representations. The PointTransformer V3 encoder first processes the input point cloud through multi-head self-attention and feed-forward layers, capturing complex spatial dependencies among points. In parallel, the INR component introduces learnable weight tokens that are concatenated with data tokens and jointly optimized within the Transformer. These weight tokens implicitly encode the mapping from different LiDAR sensor layouts to a canonical center layout, thereby enhancing structural consistency and geometric detail in the recovered features.

As the features pass through these layers, the spatial resolution progressively decreases, and the channel count in-

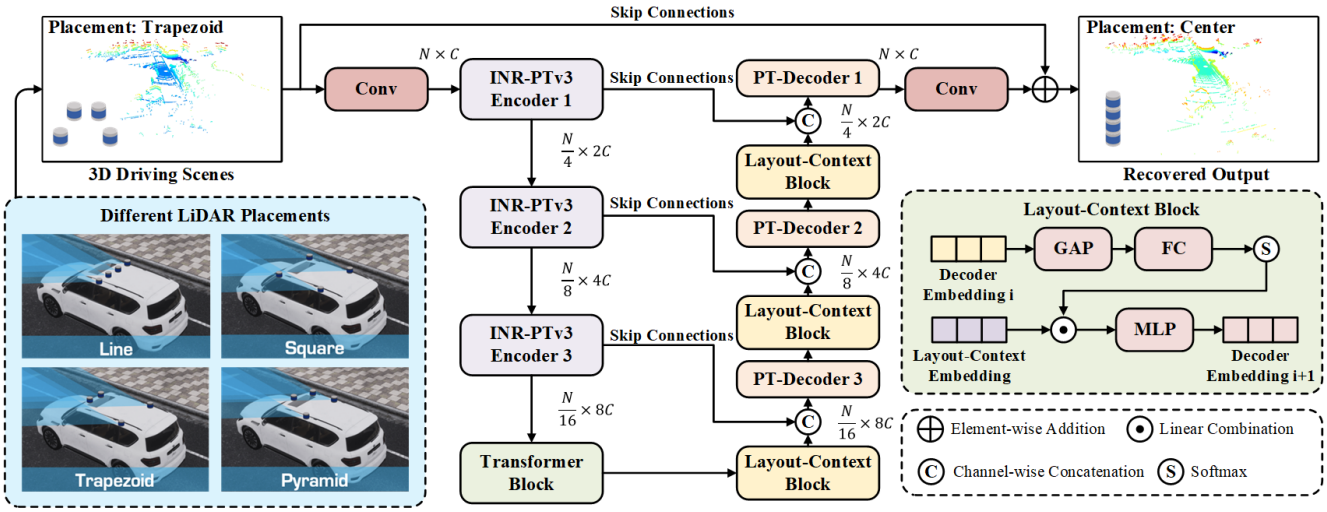

Figure 2. Overall architecture of PlaceRecover, which employs INR-PointTransformer encoders, a Transformer block, and PointTransformer decoders with Layout-Context Blocks to recover point clouds across different LiDAR sensor placements.

creases, ultimately generating a compact latent representation $F_d \in \mathbb{R}^{N/16 \times 8C}$. An additional transformer module is applied on the bottleneck feature to capture spatial correlations, thereby enhancing the feature representation ability.

Then, a three-layer layout-aware (LA) PointTransformer decoder is applied to this compact latent representation, progressively recovering the spatial details of the point cloud. In each layer of the decoder, we introduce a layout-context block, which uses contextual information related to the sensor layout to refine the features. This block helps guide the recovery process, ensuring geometric consistency and robustness of point cloud restoration across various sensor configurations. Skip connections between the encoder and decoder preserve critical spatial details, ensuring high-fidelity recovery.

Detailed descriptions of the INR-PointTransformer Encoder architecture and LayerAware-PointTransformer-Decoder design are provided.

3.2. INR-PointTransformer Encoder

Recovering point clouds from varying LiDAR layouts requires an encoder capable of both local geometric perception and global implicit structure. We propose the INR-PointTransformer Encoder, which integrates PTv3's serialized patch-attention with INR-Transformer's weight generation mechanism. This design enables efficient long-range context extraction and the direct generation of INR weights, ensuring robust mapping to a canonical center layout (Figure 3).

Input point clouds are first serialized along a space-filling curve (Z-order curve):

$$\text{Encode}(p, b, g) = (b \ll k) \mid \phi^{-1}\left(\lfloor p/g \rfloor\right), \qquad (1)$$

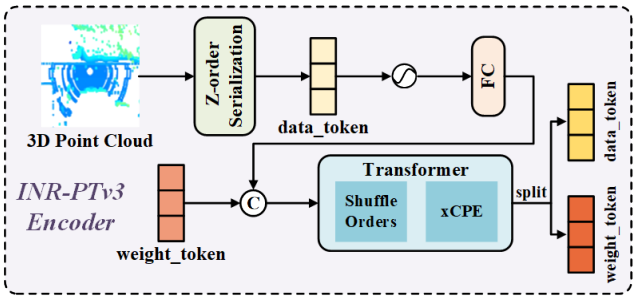

Figure 3. Structure of the INR-PTv3 encoder, where input point clouds are serialized into data tokens and combined with weight tokens, which are then refined through a Transformer with Shuffle Orders and xCPE.

where p denotes point coordinates, b is the batch index, g is the grid size, and ϕ^{-1} is the inverse mapping of the space-filling curve. Points are grouped into patches, padded as necessary, and each patch is enhanced with a learnable positional embedding e_i and projected to a *data token*:

$$\text{data\_token}_i = \text{FC}(p_i + e_i). \qquad (2)$$

This approach shifts attention from local KNN to efficient patch-attention, capturing broader spatial context.

We introduce a set of learnable weight tokens for the INR weight matrix, which are concatenated with the data tokens and processed by the Transformer. The outputs at the weight token positions are mapped to INR weight columns:

$$W_i = \text{FC}(\text{weight\_token}_i), \qquad (3)$$

with *weight grouping* applied to reduce computational complexity:

$$w_i = \text{normalize}(u_{\lfloor i/k \rfloor}) \cdot \bar{w}_i, \qquad (4)$$

where u_j is a shared vector for each group of size k, and $\bar{w}_i$ is an independent scaling parameter. This strategy enables efficient token-to-INR weight generation, ensuring robust implicit alignment.

Within each Transformer block, patch tokens undergo patch-attention, enhanced with conditional positional encoding (xCPE) via sparse convolutions and skip connections. PTv3's patch-interaction strategies (Shuffle-Order) further expand the receptive field and reduce serialization bias, facilitating both local and global context integration.

Finally, a bottleneck Transformer captures global dependencies on the compact latent representation, merging local and global features to guide the decoder in progressively restoring detailed point cloud structures.

3.3. LA-PointTransformer Decoder

Point cloud reconstruction requires a decoder that not only recovers high-resolution point clouds but also adapts to varying LiDAR layouts and environmental conditions, particularly with degraded data. We propose the LA-PointTransformer Decoder, which uses stepwise upsampling and feature refinement to enhance point cloud quality and adaptability to environmental and layout changes.

The decoder processes low-resolution point clouds with linear transformations, batch normalization (BN), and ReLU activation, followed by bilinear interpolation to restore high-resolution features. Skip connections fuse encoder features with high-resolution outputs, preserving key information. These features are then refined in the Point Transformer Block.

The PointTransformer block uses self-attention to capture spatial dependencies, with relative position encoding improving adaptability to LiDAR layouts. The core formula is:

$$y_i = \sum_{x_j \in X(i)} \rho\left(\gamma\left(\phi(x_i) - \psi(x_j) + \delta\right)\right) \odot \left(\alpha(x_j) + \delta\right). \tag{5}$$

Here, $X(i)$ is the set of k nearest neighbors of x_i, δ is the relative position encoding, γ is the attention function, and ρ is the softmax normalization.

To further enhance adaptability, the decoder introduces the layout-context Block, which dynamically adjusts features to handle degradation patterns, especially in adverse weather. It extracts global information via global average pooling (GAP), computes a weight vector, and multiplies it with context embedding (CE) to generate adjusted features. Convolutional operations are then applied to refine spatial consistency.

The feature adjustment formula in the layout-context Block is:

$$W_{wb} = \text{Softmax}\left(\text{FC}\left(\text{GAP}(F_{wb})\right)\right), \tag{6}$$

$$F_{cb} = \text{MLP}\left(W_{wb} \cdot CE\right). \tag{7}$$

Here, F_{wb} represents features refined by the Point Transformer Block, W_{wb} is the GAP-extracted weight vector, CE is the context embedding, and F_{cb} is the adjusted feature incorporating degradation information.

The layout-context block significantly improves adaptability, enabling effective point cloud recovery under diverse environments and sensor configurations.

3.4. Loss Function

To train our PlaceRecover network for point cloud recovery across varying LiDAR layouts, we define the loss function L as follows:

$$L = \frac{1}{N} \sum_{j=1}^{N} \left| d_j - \hat{d}_j \right| + \left| i_j - \hat{i}_j \right|, \tag{8}$$

where d_j and i_j are the original 3D coordinates (position) and intensity values, and $\hat{d}_j$ and $\hat{i}_j$ represent the model's predictions. Here, N denotes the total number of points in the point cloud.

This loss function penalizes the absolute differences between the predicted and original position and intensity, ensuring accurate point cloud restoration across different LiDAR layouts. By minimizing this loss, the model learns to recover smooth and reliable point clouds, even from degraded data, while being robust to outliers and ensuring stable convergence.

4. Experiments

4.1. Datasets & Evaluation Metrics

We use the official data provided by the *RoboSense Challenge 2025* [31] held at IROS 2025. This competition builds upon the legacy of the *RoboDepth Challenge 2023* [26, 29] at ICRA 2023 and the *RoboDrive Challenge 2024* [30, 58] 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 [8, 14, 38, 41, 57]. Specifically, this task is built upon the **Place3D** dataset [38] 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.

The dataset is divided into training, validation, and test subsets. Specifically, we used 200 scenes (32,000 frames) from Phase 1, with 125 scenes (20,000 frames) for model training. Each of the sensor placement layouts (Line, Trapezoid, Pyramid, Center) contains 5,000 frames of LiDAR data, with the Center layout data used as ground truth labels.

The validation set includes 75 scenes (12,000 frames), with each layout containing 3,000 frames.

The dataset contains camera images captured from four different viewpoints, LiDAR data from four distinct placements, and corresponding 3D detection annotation files. In our experiments, we used only LiDAR data to train the PlaceRecover model. The model's performance was evaluated directly through point cloud similarity, which was measured using the Chamfer Distance (CD) [11] and Earth Mover's Distance (EMD) [50], two widely used metrics for comparing point clouds. Additionally, we performed indirect evaluation using 3D detection accuracy, measured by mean Average Precision (mAP), mean Absolute Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Angle Error (mAAE), and Normalized Detection Score (NDS) [3].

To further validate the generalization capability of the algorithm, RoboSense Track3 Phase 2 provides an independent test dataset that includes camera images from six viewpoints and LiDAR data from six different layouts. Due to the lack of Center layout point cloud data in the test set, we directly evaluated the recovered point clouds by performing 3D detection tasks in the test phase, thereby indirectly validating the effectiveness of PlaceRecover.

4.2. Implementation Details

The PlaceRecover model is implemented based on the MMDetection3D codebase [9] and built with PyTorch [23]. Training was performed on a single NVIDIA RTX 4090 GPU with a batch size of 1 for 10 epochs. The encoder was initialized using pre-trained weights from the official PointTransformer V3 model. The design of the INR module is inspired by the INR-Transformer codebase, ensuring efficient integration of implicit neural representations with transformer architectures. During training, a cosine annealing strategy was employed to adjust the learning rate, starting from 5e-5 and gradually decaying to zero, facilitating stable convergence. The Adam optimizer was used for weight updates, ensuring efficient training dynamics and robust model performance.

For the 3D detection task following point cloud recovery, we used BEVFusion-L [42] as the 3D detector. The official pre-trained weights, trained on the Place3D dataset with the center placement training set for 20 epochs, were employed to initialize the model. BEVFusion-L leverages a bird's-eye view (BEV) representation for LiDAR-only 3D object detection, focusing on processing single LiDAR data to improve detection accuracy. By utilizing these pre-trained weights, the model was able to perform robust 3D detection on the recovered point clouds.

Table 1. Point Cloud Recovery Performance on Validation Set

Sensor Layout	CD ↓	EMD ↓
Line	0.032	0.118
Trapezoid	0.028	0.105
Pyramid	0.022	0.093
Average	**0.027**	**0.105**

4.3. Comparative Study

Point Cloud Recovery Evaluation: To evaluate the performance of PlaceRecover for point cloud recovery, we conducted experiments on the validation set of the RoboSense Track3 Phase 1 dataset. The recovered point clouds were evaluated using two common metrics: Chamfer Distance (CD) and Earth Mover's Distance (EMD). These metrics measure the similarity between the predicted and ground truth point clouds, where lower values indicate better recovery quality. Our method was tested on different sensor layouts (Line, Trapezoid, and Pyramid) to assess its robustness across varying LiDAR configurations. The results show that PlaceRecover excels in overall point cloud structure recovery and local detail recovery, demonstrating the effectiveness of the proposed approach for point cloud recovery. The detailed comparison of the recovery performance on the validation set is shown in Table 1 and Figure 4 (Rows 2).

3D Detection Evaluation: For the 3D detection evaluation, we utilized three representative 3D detectors, including PointPillars [35], CenterPoint [64], and BEVFusion-L [42], to perform object detection on the recovered point clouds. The models' performance was evaluated on both the validation and test sets of the Place3D dataset. The 3D detection performance was measured using multiple metrics, including mean Average Precision (mAP), mean Absolute Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error (mAOE), mean Average Velocity Error (mAVE), mean Average Angle Error (mAAE), and Normalized Detection Score (NDS).

On the validation set, the model was evaluated across different LiDAR sensor layouts, including Center, Line, Trapezoid, and Pyramid configurations. As shown in Table 2 and Figure 4 (Rows 3–8), the results indicate consistent improvements when integrating PlaceRecover with different 3D detectors. For PointPillars, the mAP improves from 0.552 to 0.604 and the mATE decreases from 0.182 to 0.163. Similarly, for CenterPoint, PlaceRecover increases the mAP from 0.601 to 0.664 and reduces the mATE from 0.161 to 0.142, along with gains in other metrics. These results demonstrate that PlaceRecover not only enhances lightweight detectors like PointPillars, but also improves center-based models such as CenterPoint. The most significant improvement is observed on BEVFusion-L, where PlaceRecover boosts the

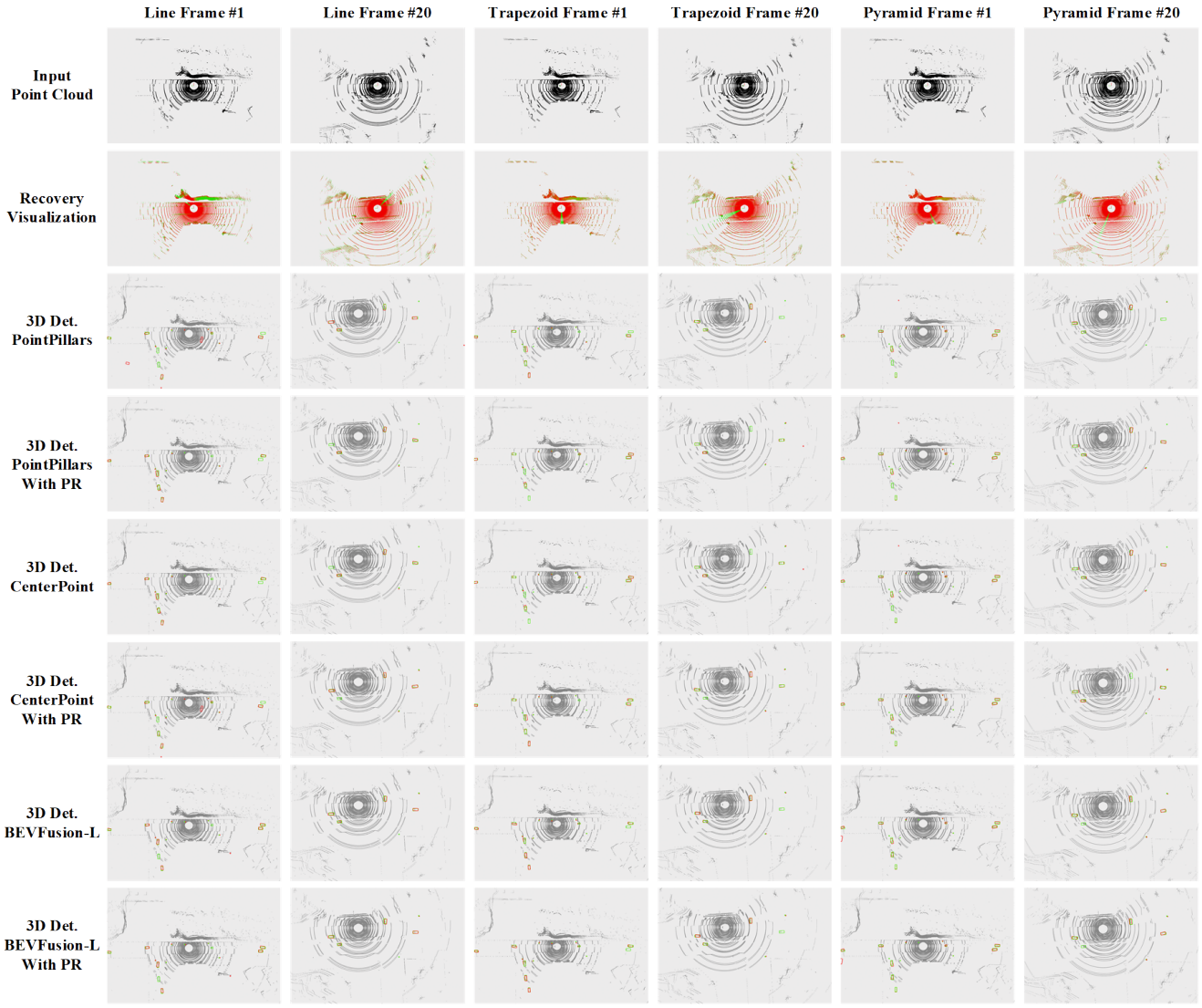

Figure 4. Visualization of point cloud recovery and 3D object detection on the validation set. The first row shows input point clouds from different LiDAR placements (Line, Trapezoid, and Pyramid). The second row presents recovery visualizations, where red points denote the recovered point clouds predicted by PlaceRecover (PR), and green points represent the ground truth point clouds at the Center placement. Rows 3–8 show 3D detection results from different detectors (PointPillars [35], CenterPoint [64], and BEVFusion-L [42]) with and without PR. In these detection visualizations, red boxes indicate the predicted bounding boxes, while green boxes denote the ground truth annotations.

Table 2. Performance Comparison of Different 3D Detectors with and without PlaceRecover on Validation Set

Method	**mAP** ↑	**mATE** ↓	**mASE** ↓	**mAOE** ↓	**mAVE** ↓	**mAAE** ↓	**NDS** ↑
PointPillars	0.552	0.182	0.220	1.452	3.012	0.420	0.482
PlaceRecover + PointPillars	0.604	0.163	0.208	1.321	2.875	0.395	0.521
CenterPoint	0.601	0.161	0.198	1.385	2.764	0.377	0.534
PlaceRecover + CenterPoint	0.664	0.142	0.182	1.242	2.635	0.351	0.573
BEVFusion-L	0.732	0.112	0.116	1.231	2.469	0.359	0.607
PlaceRecover + BEVFusion-L	**0.807**	**0.097**	**0.082**	**1.014**	**2.416**	**0.335**	**0.652**

Table 3. Performance Comparison of BEVFusion-L and PlaceRecover + BEVFusion-L on Test Set

Method	**mAP** ↑	**mATE** ↓	**mASE** ↓	**mAOE** ↓	**mAVE** ↓	**mAAE** ↓	**NDS** ↑
BEVFusion-L	0.605	0.121	0.123	1.164	2.252	0.398	0.538
PlaceRecover + BEVFusion-L	**0.726**	**0.117**	**0.129**	**1.163**	**0.817**	**0.060**	**0.651**

Table 4. Ablation Study Results on Point Cloud Recovery and 3D Detection Performance on the Validation Set

Method	**CD** ↓	**EMD** ↓	**mAP** ↑	**mATE** ↓
Baseline (PointTransformer)	0.045	0.135	0.705	0.118
+ No INR	0.055	0.142	0.687	0.121
+ No LayoutContext-Block	0.038	0.129	0.720	0.112
Full Model (PlaceRecover + PTv3)	**0.027**	**0.105**	**0.807**	**0.097**

mAP from 0.732 to 0.807 and reduces the mATE from 0.112 to 0.097. Other metrics, including mASE, mAOE, mAVE, mAAE, and NDS, also show consistent advantages. Overall, these results highlight that PlaceRecover provides universal benefits across multiple detectors, with the strongest gains achieved on BEVFusion-L, confirming its effectiveness in enhancing 3D detection performance on the validation set.

On the test set, the model was evaluated across different LiDAR sensor layouts, including Center, Line, Trapezoid, Pyramid, Square, and Random configurations. As shown in Table 3, PlaceRecover also outperforms BEVFusion-L in both mAP and mATE. The mAP score improves from 0.605 with BEVFusion-L to 0.726 with PlaceRecover, an improvement of 0.121. Similarly, mATE is reduced from 0.121 to 0.117, further confirming that PlaceRecover enhances the model's ability to localize objects more accurately on the test set. Additionally, PlaceRecover excels in other metrics, further improving the model's robustness and accuracy across various layouts.

4.4. Ablation Study

To analyze the contribution of each module to the performance of PlaceRecover, we conducted the following ablation experiments. First, we replaced PointTransformer V3 with a standard PointTransformer to evaluate the impact of the enhanced spatial feature learning and attention mechanisms of PTV3 on model performance. Next, we removed the Implicit Neural Representations (INR) module to assess the role of implicit neural representations in point cloud recovery. Finally, we removed the layout-context block to evaluate the impact of sensor layout context information on model robustness and accuracy.

The performance was evaluated using four key metrics: Chamfer Distance (CD), which measures the quality of point cloud structure recovery; Earth Mover's Distance (EMD), which evaluates the detail recovery of the point cloud; mean Average Precision (mAP), which reflects 3D object detection accuracy; and mean Absolute Translation Error (mATE), which indicates object localization accuracy.

Table 4 shows the ablation study results on the validation set, comparing the performance of each model variant across CD, EMD, mAP, and mATE, further validating the contribution of each module.

From the results of the ablation study, it is clear that PointTransformer V3 significantly improves the overall performance, especially in terms of mAP and mATE. By incorporating PointTransformer V3, the model better captures the global structural features of the point cloud data. The multi-head self-attention mechanism in PointTransformer V3 excels at capturing global spatial relationships, enabling the model to effectively capture global information when handling different LiDAR layouts, thus improving the overall recovery of the point cloud structure.

In the experiment where the INR module was removed, the model showed significant degradation in both CD and EMD, with mAP and mATE also declining. This indicates that the INR module plays a crucial role in implicitly modeling different sensor layouts. Without INR, the model failed to recover detailed features, resulting in poorer point cloud quality and detection accuracy.

The experiment of removing the layout-context block shows that, although the performance slightly declined, it still outperformed the baseline model. This demonstrates the importance of the layout-context block for sensor layout context information. By introducing contextual information related to sensor placements, the layout-context block helps the model better adapt to different sensor configurations, improving robustness, especially when handling various LiDAR sensor placements. This ultimately leads to better object localization accuracy and point cloud recovery quality.

4.5. Discussion and Limitations

Although PlaceRecover demonstrates strong performance in point cloud recovery and 3D object detection, there are still some limitations to address. First, due to the Transformer-based architecture of PlaceRecover, the model's computational cost can be high, especially when processing large-scale point clouds or real-time applications. The self-attention mechanism in the Transformer requires computation for all input points, which leads to higher computational complexity. To improve inference speed, further optimization of the model is needed to reduce computational resource consumption. Second, the current approach relies on supervised learning and fixed sensor configurations, which may not generalize well to real-world scenarios where sensor placements are more diverse and dynamic. Future work could explore the integration of unsupervised learning techniques and extend the method to handle more flexible and adaptive sensor configurations, enabling better adaptability across various environments.

5. Conclusion

In this study, we propose PlaceRecover, a Transformer-based point cloud recovery network for robust LiDAR sensor placement adaptation. By integrating PointTransformer V3, implicit neural representations (INR), and the Layout-Context Block, PlaceRecover enhances point cloud recovery and 3D object detection performance. Experimental results show that PlaceRecover significantly improves the mAP and mATE scores, achieving 0.807 and 0.097 on the validation set, respectively, surpassing the baseline model (BEVFusion-L). On the test set, PlaceRecover further outperforms BEVFusion-L, demonstrating its robustness across diverse LiDAR sensor placements. Our approach secured fourth place in the 2025 RoboSense Challenge Track 3, highlighting its effectiveness in adapting to varying sensor configurations.

References

[1] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences. In *IEEE/CVF International Conference on Computer Vision*, pages 9297–9307, 2019. 1

[2] H. Bian et al. DynamicCity: Large-scale 4D occupancy generation from dynamic scenes. In *International Conference on Learning Representations*, 2025. 2

[3] H. Caesar, V. Bankiti, A. H. Lang, et al. nuScenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 2, 5

[4] R. Chen et al. CLIP2Scene: Towards label-efficient 3D scene understanding by CLIP. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7020–7030, 2023. 2

[5] R. Chen et al. Towards label-free scene understanding by vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 75896–75910, 2023. 2

[6] X. Chen, T. Zhang, Y. Wang, et al. FUTR3D: A unified sensor fusion framework for 3D detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 172–181, 2023. 1

[7] Y. Chen and X. Wang. Transformers as meta-learners for implicit neural representations. In *European Conference on Computer Vision*, pages 170–187, Cham, 2022. Springer Nature Switzerland. 2

[8] M. Chu, Z. Zheng, W. Ji, T. Wang, and T.-S. Chua. Towards natural language-guided drones: GeoText-1652 benchmark with spatial relation matching. In *European Conference on Computer Vision*, pages 213–231, 2024. 4

[9] M. Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 5

[10] T. Cortinhal, G. Tzelepis, and E. E. Aksoy. SalsaNext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *International Symposium on Visual Computing*, pages 207–222, Cham, 2020. Springer International Publishing. 1

[11] H. Fan, H. Su, and L. J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 605–613, 2017. 5

[12] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada. Panoptic nuScenes: A large-scale benchmark for LiDAR panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022. 1

[13] A. Geiger, P. Lenz, C. Stiller, et al. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 1, 2

[14] Z. Gong, T. Hu, R. Qiu, and J. Liang. From cognition to precognition: A future-aware framework for social navigation. In *IEEE International Conference on Robotics and Automation*, pages 9122–9129, 2025. 4

[15] X. Hao, M. Wei, Y. Yang, et al. Is your HD map constructor reliable under sensor corruptions? In *Advances in Neural Information Processing Systems*, volume 37, pages 22441–22482, 2024. 2

[16] X. Hao, G. Liu, Y. Zhao, et al. MSC-Bench: Benchmarking and analyzing multi-sensor corruption for driving perception. *arXiv preprint arXiv:2501.01037*, 2025.

[17] X. Hao et al. SafeMap: Robust HD map construction from incomplete observations. In *International Conference on Machine Learning*, pages 22091–22102. PMLR, 2025. 2

[18] K. Haydarov, A. Muhamed, X. Shen, et al. Adversarial text to continuous image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6316–6326, 2024. 2

[19] F. Hong, L. Kong, H. Zhou, X. Zhu, H. Li, and Z. Liu. Unified 3D and 4D panoptic segmentation via dynamic shifting networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3480–3495, 2024. 1

[20] J. S. K. Hu, T. Kuai, and S. L. Waslander. Point density-aware voxels for lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8469–8478, 2022. 2

[21] Q. Hu, B. Yang, L. Xie, et al. RandLA-Net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020. 1, 2

[22] M. Hui, Z. Wei, H. Zhu, et al. Microdiffusion: Implicit representation-guided diffusion for 3d reconstruction from limited 2d microscopy projections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11460–11469, 2024. 2

[23] S. Imambi, K. B. Prakash, and G. R. Kanagachidambaresan. Pytorch. In *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104. Springer International Publishing, Cham, 2021. 5

[24] L. Kong, Y. Liu, R. Chen, Y. Ma, X. Zhu, Y. Li, Y. Hou, Y. Qiao, and Z. Liu. Rethinking range view representation for LiDAR segmentation. In *IEEE/CVF International Conference on Computer Vision*, pages 228–240, 2023. 2

[25] L. Kong, Y. Liu, X. Li, R. Chen, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu. Robo3D: Towards robust and reliable 3D perception against corruptions. In *IEEE/CVF International Conference on Computer Vision*, pages 19994–20006, 2023. 2

[26] L. Kong, Y. Niu, S. Xie, H. Hu, L. X. Ng, B. Cottereau, L. Zhang, H. Wang, W. T. Ooi, R. Zhu, Z. Song, L. Liu, T. Zhang, J. Yu, M. Jing, P. Li, X. Qi, C. Jin, Y. Chen, J. Hou, J. Zhang, Z. Kan, Q. Lin, L. Peng, M. Li, D. Xu, C. Yang, Y. Yao, G. Wu, J. Kuai, X. Liu, J. Jiang, J. Huang, B. Li, J. Chen, S. Zhang, S. Ao, Z. Li, R. Chen, H. Luo, F. Zhao, and J. Yu. The RoboDepth challenge: Methods and advancements towards robust depth estimation. *arXiv preprint arXiv:2307.15061*, 2023. 4

[27] L. Kong, N. Quader, and V. E. Liong. ConDA: Unsupervised domain adaptation for LiDAR segmentation via regularized domain concatenation. In *IEEE International Conference on Robotics and Automation*, pages 9338–9345, 2023. 2

[28] L. Kong, J. Ren, L. Pan, et al. Lasermix for semi-supervised lidar semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21705–21715, 2023. 2

[29] L. Kong, S. Xie, H. Hu, L. X. Ng, B. R. Cottereau, and W. T. Ooi. RoboDepth: Robust out-of-distribution depth estimation under corruptions. In *Advances in Neural Information Processing Systems*, volume 36, pages 21298–21342, 2023. 4

[30] L. Kong, S. Xie, H. Hu, Y. Niu, W. T. Ooi, B. R. Cottereau, L. X. Ng, Y. Ma, W. Zhang, L. Pan, K. Chen, Z. Liu, W. Qiu, W. Zhang, X. Cao, H. Lu, Y.-C. Chen, C. Kang, X. Zhou, C. Ying, W. Shang, X. Wei, Y. Dong, B. Yang, S. Jiang, Z. Ma, D. Ji, H. Li, X. Huang, Y. Tian, G. Kou, F. Jia, Y. Liu, T. Wang, Y. Li, X. Hao, Y. Yang, H. Zhang, M. Wei, Y. Zhou, H. Zhao, J. Zhang, J. Li, X. He, X. Cheng, B. Zhang, L. Zhao, D. Ding, F. Liu, Y. Yan, H. Wang, N. Ye, L. Luo, Y. Tian, Y. Zuo, Z. Cao, Y. Ren, Y. Li, W. Liu, X. Wu, Y. Mao, M. Li, J. Liu, J. Liu, Z. Qin, C. Chu, J. Xu, W. Zhao, J. Jiang, X. Liu, Z. Wang, C. Li, S. Li, C. Yuan, S. Yang, W. Liu, P. Chen, B. Zhou, Y. Wang, C. Zhang, J. Sun, H. Chen, X. Yang, L. Wang, D. Fu, Y. Lin, H. Yang, H. Li, Y. Luo, X. Cheng, and Y. Xu. The RoboDrive challenge: Drive anytime anywhere in any condition. *arXiv preprint arXiv:2405.08816*, 2024. 4

[31] L. Kong, S. Xie, Z. Gong, Y. Li, M. Chu, A. Liang, Y. Dong, T. Hu, R. Qiu, R. Li, H. Hu, D. Lu, W. Yin, W. Ding, L. Li, H. Song, W. Zhang, Y. Ma, J. Liang, Z. Zheng, L. X. Ng, B. R. Cottereau, W. T. Ooi, Z. Liu, Z. Zhang, W. Qiu, W. Zhang, J. Ao, J. Zheng, S. Wang, G. Yang, Z. Zhang, Y. Zhong, E. Gao, X. Zheng, X. Wang, S. Li, Y. Gao, S. Lan, M. Han, X. Hu, D. Malic, C. Fruhwirth-Reisinger, A. Prutsch, W. Lin, S. Schulter, H. Possegger, L. Li, J. Zhao, Z. Yang, Y. Song, B. Lin, T. Zhang, Y. Yuan, C. Zhang, X. Li, Y. Kim, S. Hwang, H. Jeong, A. Wu, X. Luo, E. Xiao, L. Zhang, Y. Tang, H. Cheng, R. Xu, W. Ding, L. Zhou, L. Chen, H. Ye, X. Hao, S. Li, J. Shen, X. Li, H. Ruan, J. Lin, Z. Luo, Y. Zang, C. Wang, H. Wang, X. Gong, Y. Yang, Q. Ma, Z. Zhang, W. Shi, J. Zhou, W. Zeng, K. Xu, Y. Zhang, H. Fu, R. Hu, Y. Ma, X. Feng, W. Zhang, L. Zhang, Y. Zhuge, H. Lu, Y. He, S. Yu, J. Park, Y. Lim, H. Shim, F. Liang, Z. Wang, Y. Peng, G. Zong, X. Li, B. Wang, H. Wei, Y. Ma, Y. Shi, S. Liu, D. Kong, Y. Lin, H. Yang, L. Lei, H. Li, X. Zhang, Z. Wang, X. Wang, Y. Fu, Y. Luo, D. Etchegaray, Y. Li, C. Li, Y. Sun, W. Zhu, W. Xu, L. Li, L. Liao, J. Yan, B. Wang, X. Ren, X. Yue, J. Zheng, J. Wu, S. Qin, W. Cong, and Y. He. The RoboSense challenge: Sense anything, navigate anywhere, adapt across platforms. https://robosense2025.github.io, 2025. 4

[32] L. Kong, X. Xu, Y. Liu, J. Cen, R. Chen, W. Zhang, L. Pan, K. Chen, and Z. Liu. LargeAD: Large-scale cross-sensor data pretraining for autonomous driving. *arXiv preprint arXiv:2501.04005*, 2025. 1

[33] L. Kong, X. Xu, J. Ren, et al. Multi-modal data-efficient 3D scene understanding for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(5): 3748–3765, 2025. 2

[34] L. Kong, W. Yang, J. Mei, Y. Liu, A. Liang, D. Zhu, D. Lu, W. Yin, X. Hu, M. Jia, J. Deng, K. Zhang, Y. Wu, T. Yan, S. Gao, S. Wang, L. Li, L. Pan, Y. Liu, J. Zhu, W. T. Ooi, S. C. H. Hoi, and Z. Liu. 3D and 4D world modeling: A survey. *arXiv preprint arXiv:2509.07996*, 2025. 2

[35] A. H. Lang, S. Vora, H. Caesar, et al. PointPillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 2, 5, 6

[36] R. Li, Y. Dong, T. Hu, A. Liang, et al. 3EED: Ground everything everywhere in 3D. *arXiv preprint arXiv:2511.01755*, 2025. 1

[37] R. 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. 2

[38] Y. Li, L. Kong, H. Hu, X. Xu, and X. Huang. Is your LiDAR placement optimized for 3D scene understanding? In *Advances in Neural Information Processing Systems*, volume 37, pages 34980–35017, 2024. 1, 2, 4

[39] Y. Li et al. Optimizing LiDAR placements for robust driving perception in adverse conditions. *arXiv preprint arXiv:2403.17009*, 2024. 1

[40] A. Liang, Y. Liu, Y. Yang, et al. LiDARCrafter: Dynamic 4D world modeling from LiDAR sequences. *arXiv preprint arXiv:2508.03692*, 2025. 1

[41] A. Liang et al. Perspective-invariant 3D object detection. In *IEEE/CVF International Conference on Computer Vision*, pages 27725–27738, 2025. 4

[42] T. Liang, H. Xie, K. Yu, et al. BEVFusion: A simple and robust lidar-camera fusion framework. *Advances in Neural Information Processing Systems*, 35:10421–10434, 2022. 1, 2, 5, 6

[43] Y. Liu et al. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, pages 37193–37229, 2023. 2

[44] Y. Liu et al. UniSeg: A unified multi-modal LiDAR segmentation network and the openpcseg codebase. In *IEEE/CVF International Conference on Computer Vision*, pages 21662–21673, 2023. 2

[45] Y. Liu et al. Multi-space alignments towards universal LiDAR segmentation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14648–14661, 2024. 1

[46] Y. Liu et al. La La LiDAR: Large-scale layout generation from LiDAR data. *arXiv preprint arXiv:2508.03691*, 2025. 1

[47] A. Milioto, I. Vizzo, J. Behley, et al. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019. 1, 2

[48] X. Peng, R. Chen, F. Qiao, et al. Learning to adapt SAM for segmenting cross-domain point clouds. In *European Conference on Computer Vision*, pages 54–71. Springer, 2024. 2

[49] A. W. Reed, H. Kim, R. Anirudh, et al. Dynamic ct reconstruction from limited views with implicit neural representations and parametric motion fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2258–2268, 2021. 2

[50] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000. 5

[51] I. Skorokhodov, S. Ignatyev, and M. Elhoseiny. Adversarial generation of continuous images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10753–10764, 2021. 2

[52] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 1

[53] X. Wang, X. Wu, S. Wang, et al. Monocular semantic scene completion via masked recurrent networks. In *IEEE/CVF International Conference on Computer Vision*, pages 24811–24822, 2025. 2

[54] X. 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. 1

[55] X. Wu, L. Jiang, P. S. Wang, et al. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4840–4851, 2024. 2

[56] S. Xie, L. Kong, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu. RoboBEV: Towards robust bird's eye view perception under corruptions. *arXiv preprint arXiv:2304.06719*, 2023. 2

[57] S. Xie, L. Kong, Y. Dong, C. Sima, W. Zhang, Q. A. Chen, Z. Liu, and L. Pan. 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. 4

[58] S. Xie, L. Kong, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu. 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. 4

[59] J. Xu, W. Yang, L. Kong, Y. Liu, Q. Zhou, R. Zhang, Z. Li, W.-M. Chen, and B. Fei. Visual foundation models boost cross-modal unsupervised domain adaptation for 3d semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 26(11):20287–20301, 2025. 1

[60] X. Xu, L. Kong, H. Shuai, L. Pan, Z. Liu, and Q. Liu. LiMoE: Mixture of LiDAR representation learners from automotive scenes. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27368–27379, 2025. 1

[61] X. Xu et al. 4D contrastive superflows are dense 3D representation learners. In *European Conference on Computer Vision*, pages 58–80, 2024. 1

[62] X. Xu et al. FRNet: Frustum-range networks for scalable LiDAR segmentation. *IEEE Transactions on Image Processing*, 34:2173–2186, 2025. 2

[63] X. Xu et al. Beyond one shot, beyond one perspective: Cross-view and long-horizon distillation for better LiDAR representations. In *IEEE/CVF International Conference on Computer Vision*, pages 25506–25518, 2025. 1

[64] T. Yin, X. Zhou, and P. Krahenbuhl. Center-based 3D object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. 1, 5, 6

[65] D. Zhu, Y. Hu, Y. Liu, et al. Spiral: Semantic-aware progressive LiDAR scene generation and understanding. *arXiv preprint arXiv:2505.22643*, 2025. 1