Frequency-Aware RGB Guidance for 3D Object Detection with Extremely Sparse LiDAR

Ruixiao Xu, Lei Tang*, Junzhe Zhang
School of Information Engineering, Chang'an University, Xi'an, China
xurxqvq@gmail.com, tanglei24@chd.edu.cn, zjz_mulious@163.com

***Abstract*—3D object detection is a critical task in autonomous driving. However, LiDAR-based detectors suffer severe performance degradation when point clouds become extremely sparse due to cost-effective hardware constraints, long-range attenuation, or environmental signal degradation. While fusing monocular images with sparse LiDAR data is a viable solution to bridge this resolution gap, existing methods often confuse high-frequency textures with low-frequency geometric structures, leading to blurred object boundaries and reduced detection accuracy. We propose a 3D detection framework specifically designed for extremely sparse scenarios (*e.g.*, 1% density). Its core is the Dynamic Frequency Guidance (DFG) module. This module decouples and selectively fuses high- and low-frequency information through content-adaptive filtering, enabling more accurate depth reconstruction even with minimal point cloud data. The resulting dense depth map is converted into pseudo-LiDAR data, which can be directly fed into existing detectors. On the KITTI benchmark, when only 1% of LiDAR points are available, our method achieves $\mathbf{AP_{3D}}$ scores of 41.92%, 23.70%, and 18.73% for easy, moderate, and hard settings. Notably, this represents a substantial gain of 12.03% $\mathbf{AP_{3D}}$ in the moderate setting over the baseline PV-RCNN, verifying our method's capability to recover detailed geometry from extremely sparse inputs.**

***Index Terms*—3D Object Detection, Multi-modal Fusion, Depth Completion, Pseudo-LiDAR**

I. Introduction

3D object detection is essential for autonomous driving perception systems. It aims to provide accurate 3D information about surrounding objects, including their semantics, positions, sizes, and orientations. This fine-grained spatial awareness is critical for vehicle decision-making, planning, and control.

Recently, significant progress has been made in 3D object detection research. Existing approaches can be broadly categorized into RGB-based, LiDAR-based, and multi-modal methods. Among these, LiDAR-based approaches have become dominant [1]–[4] due to their ability to provide precise geometric measurements. However, reliance on dense point clouds limits scalability. In practical applications, the effective point density is often compromised by hardware constraints, distance-dependent sparsity, and signal degradation from environmental factors [5]. In these scenarios, LiDAR fails to capture sufficient structural details, while monocular cameras often retain dense semantic and textural information. Therefore, ensuring robust detection performance under extremely sparse conditions (*e.g.*, with only 1% valid points) by leveraging the complementary strengths of visual data is crucial for the safety and reliability of autonomous driving systems.

To address sparsity, monocular 3D detection [6]–[8] offers a low-cost alternative but suffers from poor localization due to the lack of precise depth. While fusing RGB images with sparse LiDAR [9], [10] is a promising direction, existing methods often indiscriminately mix high-frequency textures with low-frequency geometric structures in the feature space. This leads to feature interference and blurred object boundaries, ultimately degrading detection performance.

To this end, we propose a 3D object detection method based on monocular cameras and extremely sparse LiDAR. Our method first employs a depth completion network to generate a dense depth map using the RGB image and the extremely sparse LiDAR. Considering that inaccurate depth information at edges leads to pseudo-LiDAR distortion, we introduce a dynamic frequency guidance strategy to refine the depth map using rich semantic information from the RGB image. Subsequently, the dense depth map is converted into pseudo-LiDAR. A LiDAR-based detector is used to predict 3D information from pseudo-LiDAR points. To validate the effectiveness of our approach, we conduct evaluations on both the single-stage detector PointPillars [3] and the two-stage detector PV-RCNN [2].

We evaluated the proposed method on the KITTI dataset. On the KITTI test set, with an input density of only 1%, our method achieves 41.92%, 23.70%, and 18.73% of AP_{3D} in easy, moderate and hard settings. With PV-RCNN as the downstream detector, our method substantially improves AP_{3D} under the 1% setting on KITTI. It demonstrates the sound performance of our approach in degenerate scenarios.

The main contributions are summarized as follows:

- We propose a 3D detection framework specifically designed to operate effectively under extremely sparse LiDAR (*e.g.*, 1%) conditions by leveraging monocular RGB guidance.
- We introduce a Dynamic Frequency Guidance (DFG) module that separates and selectively fuses high- and low-frequency features to refine depth and preserve sharp object boundaries.
- We conduct a systematic robustness study across varying LiDAR density levels, verifying that our method remains

*Corresponding author.

robust at 1% density and stable across the full range of density conditions.

II. Related Work

A. Depth Completion

RGB-guided depth completion aims to predict dense depth from sparse input by using the semantics and textures of RGB images. Under extreme sparsity, however, depth provides weak geometric constraints, making models increasingly dependent on RGB to infer fine structures (*e.g.*, object boundaries). Existing methods struggle to disentangle these textural artifacts from genuine geometric boundaries during fusion, often resulting in depth maps with blurred boundaries.

To mitigate this, PENet [11] employs a dual-branch design to estimate depth from RGB and sparse depth separately. GuideNet [12] introduces guided dynamic filters for cross-modal fusion but incurs heavy computation due to large parameterization. Wang et al. [13] propose decomposed guided dynamic filters to improve efficiency and accuracy. Nevertheless, these approaches typically mix frequency components indiscriminately within the feature space, and their performance remains limited when depth input becomes extremely sparse.

To further sharpen boundaries, spatial propagation networks (SPNs) [14]–[18] refine an initial depth map via iterative affinity-based propagation. However, propagation is often decoupled from feature fusion, which hinders end-to-end optimization of global structure. Recently, Yan et al. [19] reformulated depth completion as a degradation-recovery task and selectively enhance high-frequency cues, but still do not explicitly disentangle geometric edges from semantic textures.

B. LiDAR-Based 3D Detection

LiDAR-based 3D object detection methods have progressed significantly. Grid-based methods, such as VoxelNet [20] and SECOND [4], encode point clouds into voxels and apply 3D or sparse convolutions for efficient feature extraction. PointPillars [3] further optimizes efficiency by projecting points into vertical pillars. Conversely, point-based methods [21]–[23] directly process raw point clouds via hierarchical sampling to preserve precise geometry, though often at a higher computational cost. To balance efficiency and accuracy, 3DSSD [21] introduces a fusion sampling strategy, while PV-RCNN [2] integrates voxel-based abstractions with point-based set abstraction to enhance RoI feature representation. These geometry-dependent networks suffer severe performance degradation when the input density drops to extremely low levels (*e.g.*, 1%), as the sparse points fail to capture sufficient structural details.

To compensate for sparsity, multi-modal fusion has been extensively explored. Point-level fusion approaches [24]–[26] decorate LiDAR points with image features but remain sensitive to calibration errors. BEVFusion [27] lifts image features into a shared Bird's-Eye-View (BEV) space for dense fusion, significantly improving global perception. Additionally, SFD [28] enhances robustness by fusing sparse LiDAR points with dense pseudo-points. However, these methods depend heavily on the density of the input point cloud. Pseudo-LiDAR frameworks [29], [30] offer a potential solution by transforming depth predictions into 3D point cloud representations for detection. Consequently, under sparse LiDAR conditions, generating depth maps with sharp edges and accurate geometry during the depth completion stage is crucial for improving 3D detection performance.

III. Methodology

Our overall framework is illustrated in Fig. 1. It primarily consists of two stages: (1) **Depth Completion**, which generates a dense depth map by fusing an RGB image with a sparse LiDAR scan. (2) **LiDAR-based detector**, where the dense map is projected into a pseudo-LiDAR point cloud using camera intrinsics. The point cloud is then fed into a LiDAR-based 3D object detection network to obtain 3D bounding boxes. These two stages are trained separately and connected during the inference process.

A. Depth Completion

To bridge the resolution gap between dense RGB images and sparse LiDAR measurements, we employ an asymmetric dual-branch network architecture. As illustrated in Fig. 1, the RGB branch extracts multi-scale semantic features, which are progressively integrated into the depth branch to guide densification. Unlike methods that rely on simple concatenation, which indiscriminately mixes RGB textures with geometric structures, our approach ensures a more structured integration. We introduce the Dynamic Frequency Guidance (DFG) module, which comprises two core components. First, the Dynamic Frequency Decoupling (DFD) module explicitly decouples features into high- and low-frequency components, ensuring that textural noise does not corrupt geometric consistency. Subsequently, a Selective Frequency Fusion (SFF) module adaptively recombines these components via multi-scale attention, prioritizing high-frequency details at boundaries while preserving depth priors in smooth regions.

1) *Dynamic Frequency Decoupling.* The DFD module is illustrated in Fig.3. It takes guidance features $f_{ri} \in \mathbb{R}^{C^r \times H^r \times W^r}$ from the RGB decoder and depth features $f_{edi} \in \mathbb{R}^{C^d \times H^d \times W^d}$ from the depth encoder as input. First, we fuse them via a standard convolution to obtain an initial representation $f_f \in \mathbb{R}^{C^f \times H^f \times W^f}$. To disentangle structural geometry from textural details, we propose a *Dynamic Frequency Decoupling* strategy. Since pixel-wise dynamic filters impose prohibitive memory costs of $\mathcal{O}(H^f \times W^f \times k^2 \times C^f)$, we devise a "spatially-shared, channel-grouped" mechanism. This design generates scene-adaptive kernels shared across spatial locations, while reducing the parameter cost to $\mathcal{O}(g \times k^2)$.

Specifically, we divide the C^f channels into g groups (where $g \ll C^f$) and generate a shared $k \times k$ kernel for each group. An auxiliary network generates the low-pass filter weights $\mathcal{W}_d \in \mathbb{R}^{g \times k^2 \times 1 \times 1}$ from the global context vector $\mathbf{v}$, obtained through Global Average Pooling (GAP):

$$\mathbf{v} = \mathcal{G}(f_f), \tag{1}$$

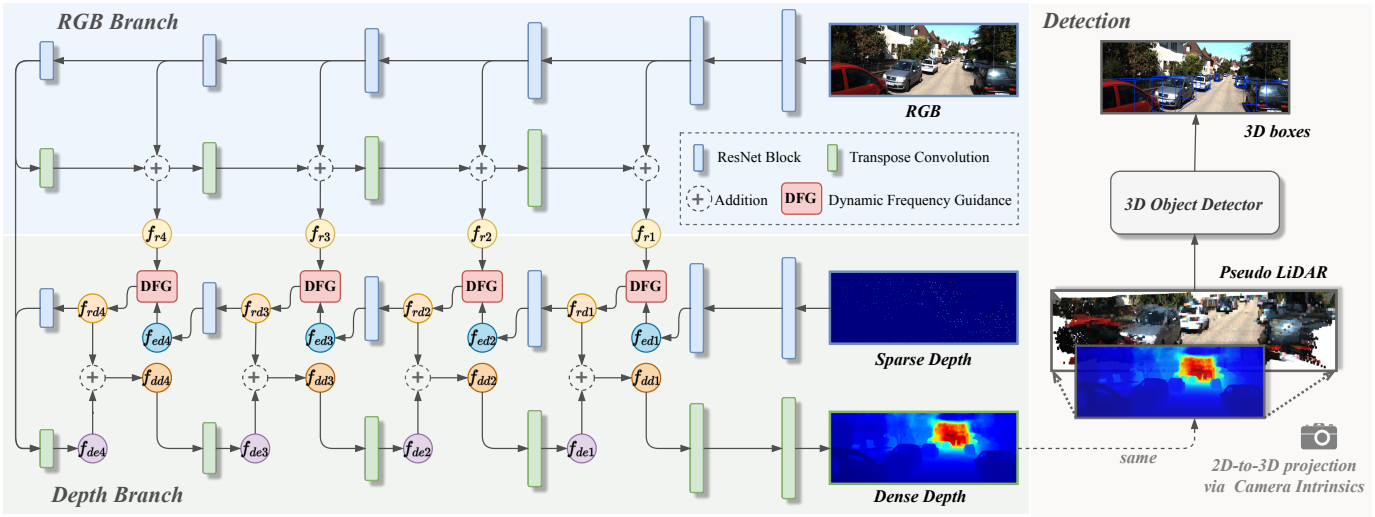

Fig. 1. Our whole framework. It consists of two stages: (1) depth map prediction and (2) LiDAR-based 3D detection. The depth completion network contains the RGBNet and DepthNet branches, and their multi-scale features are fused through the proposed method.

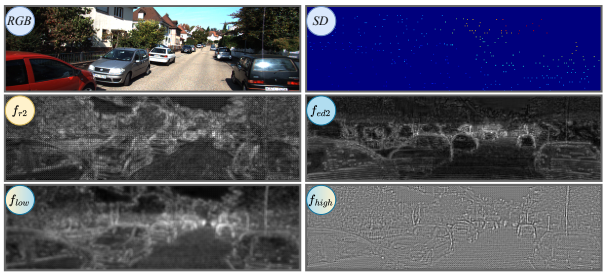

Fig. 2. Visualization of the Dynamic Frequency Decoupling. We average the feature maps along the channel dimension to illustrate the aggregated response intensity. Given an input RGB image and a sparse depth map, the proposed DFG module separates low-frequency signals (representing smooth regions of global depth structure) from high-frequency signals (corresponding to sharp edges of object boundaries). This demonstrates that the module learns to focus on structural cues, which are crucial for accurate depth completion and downstream 3D detection.

$$\mathcal{W}_d = \text{Softmax}\left(\text{BN}\left(\mathcal{C}_a(\mathbf{v}) \odot \sigma\left(\mathcal{C}_g(\mathbf{v})\right)\right)\right), \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^{C^f \times 1 \times 1}$, $\mathcal{C}_a$ and $\mathcal{C}_g$ denote 1×1 convolutions, σ is the Sigmoid function, and the Softmax ensures the weights act as a normalized smoothing operator.

We define the decoupling process as a complementary decomposition. The low-frequency component $f_{low} \in \mathbb{R}^{C^f \times H^f \times W^f}$ (representing smooth geometric surfaces) is obtained by applying this learnable smoothing operator, while the high-frequency component $f_{high} \in \mathbb{R}^{C^f \times H^f \times W^f}$ (capturing sharp edges and textures) is derived as the residual:

$$f_{low} = \mathcal{W}_d \circledast f_f, \quad f_{high} = f_f - f_{low}, \tag{3}$$

where $\circledast$ denotes the grouped convolution. As visualized in Fig. 2, this decomposition effectively isolates structural cues in f_{low} while highlighting high-frequency boundaries in f_{high}.

2) *Selective Frequency Fusion.* Since frequency contributions vary spatially, simple summation is suboptimal. Therefore, we introduce the Selective Frequency Fusion (SFF) module to adaptively recombine them into the final feature $f_{rd} \in \mathbb{R}^{C^f \times H^f \times W^f}$. After concatenating the high- and low-frequency components into $f_{lh} \in \mathbb{R}^{2C^f \times H^f \times W^f}$, we apply multi-scale spatial attention. We use dilated convolutions with rates $d \in \{1, 3, 5\}$ to enlarge receptive fields, which allows us to aggregate distant geometric context and mitigate sparsity when generating the spatial weight map $\mathbf{m}_s \in \mathbb{R}^{1 \times H^f \times W^f}$. Meanwhile, we compute channel weights $\mathbf{m}_c \in \mathbb{R}^{C^f \times 1 \times 1}$ to suppress redundancy. Finally, the fused representation is obtained by dynamically re-weighting components via the mask $\mathcal{M} \in \mathbb{R}^{C^f \times H^f \times W^f}$:

$$\mathcal{M} = \mathbf{m}_c \odot \mathbf{m}_s, \tag{4}$$

$$f_{rd} = \mathcal{M} \odot f_{high} + (1 - \mathcal{M}) \odot f_{low}, \tag{5}$$

This achieves a balance between high-frequency detail enhancement and low-frequency semantic preservation.

B. LiDAR-based detector

1) *Pseudo-LiDAR points generation.* Each pixel (u, v) and its depth value $d(u, v)$ in the depth map is back-projected into 3D space using the camera intrinsics, thereby reconstructing 3D points. The specific coordinate transformation formula is as follows:

$$x = \frac{u - c_u}{f_u} \times z, \quad y = \frac{v - c_v}{f_v} \times z, \quad z = d(u, v). \tag{6}$$

where (c_u, c_v) denotes the principal point coordinates of the camera, and f_u, f_v represent the focal lengths in the horizontal and vertical directions, respectively. Applying this operation to all pixels on the depth map, we obtain a 3D point cloud $\{(x^{(n)}, y^{(n)}, z^{(n)})\}_{n=1}^{N}$, where N is the total number of pixels.

2) *Processing and Detection.* To ensure compatibility with standard LiDAR detectors, we align the pseudo-LiDAR distribution with real LiDAR data through two post-processing steps. First, following [31], we remove all points higher than 1 m above the sensor to filter out irrelevant overhead outliers (*e.g.*, traffic signs) on the KITTI benchmark. Second, since the

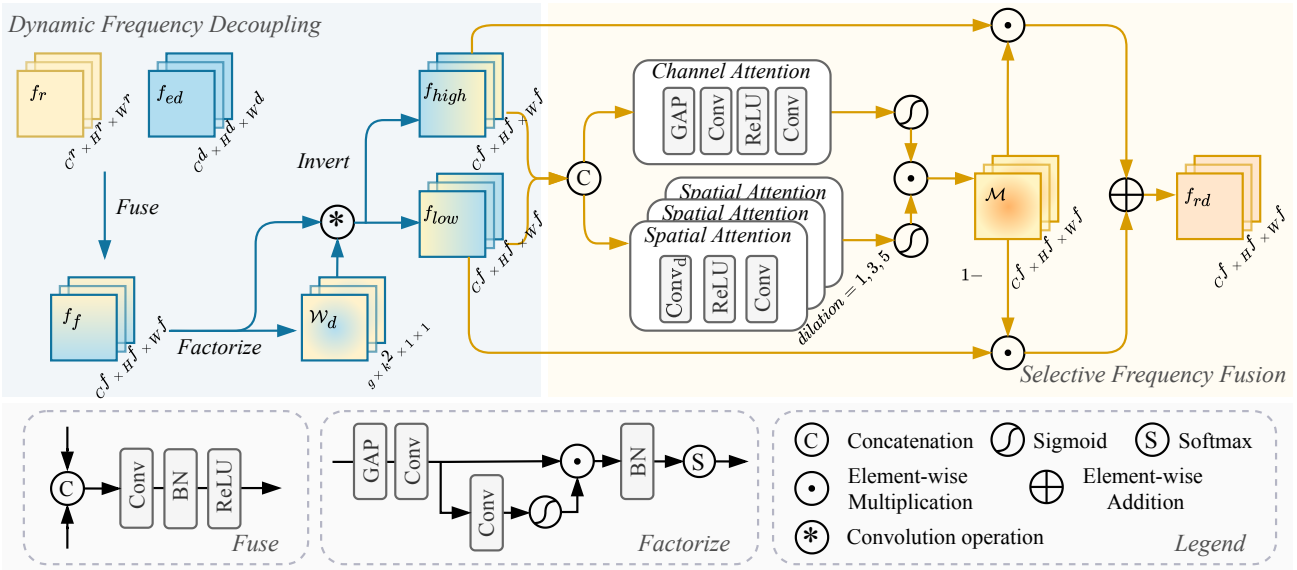

Fig. 3. Dynamic Frequency Guidance (DFG) module. Given RGB features and sparse depth features, DFG decomposes them into low-frequency and high-frequency components, applies dynamic filtering and selective fusion, and produces a frequency-enhanced depth representation that preserves global structure while sharpening object boundaries.

depth completion network does not infer material properties, we pad the missing reflectance dimension with a constant value of 1.0. Finally, the processed point cloud is directly fed into off-the-shelf 3D detectors. In this work, we employ both PointPillars [3] and PV-RCNN [2] to validate the effectiveness of our framework.

IV. Experiments

A. Implementation Details

1) *Depth Completion Network.* Our depth completion network is implemented based on the PyTorch framework and trained on a single NVIDIA GeForce RTX 4090 GPU. All experiments are conducted using the AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.999$, and we employ L_2 loss during training. The batch size is set to 8, and the initial learning rate is 10^{-3}. The learning rate is reduced by 50% every 5 epochs. We train our depth completion network for a total of 30 epochs. To improve the model's robustness to point cloud density, we introduce a practical random masking strategy during the training phase, randomly dropping 0% to 95% of valid depth points to simulate varying degrees of density. For the downstream 3D object detection task, we fix the input LiDAR densities (1%, 6%, 8%, 10%, and 12%) through uniform sampling, infer dense depth maps, and back-project them to pseudo-LiDAR point clouds for detection.

2) *3D Object Detection.* To ensure a fair comparison, we implement and retrain all baseline methods using the official OpenPCDet [32] codebase. Since the performance of these methods under extremely sparse point cloud conditions is not readily available in existing literature, we conduct all experiments using uniform settings provided by this standard framework to guarantee consistent evaluation.

B. Datasets and Evaluation Metrics

1) *KITTI Depth Completion Dataset.* We use the KITTI depth completion dataset [33] to train the first-stage network. This dataset contains sparse depth maps collected by a Velodyne 64-line LiDAR and their corresponding RGB images. The ground truth depth maps are generated by accumulating 11 consecutive LiDAR scans. Following the settings of existing depth completion methods, we use 86k images to train our network.

2) *KITTI 3D Object Detection Dataset.* Our method is evaluated on the KITTI object detection benchmark [34], [35], which contains 7,481 training images and 7,518 test images. We follow the same training and validation splits as suggested by Chen et al. [36], containing 3,712 and 3,769 images, respectively. Each sample includes an RGB image, the corresponding Velodyne LiDAR point cloud, stereo imagery, and calibration data. Evaluation is based on the standard Average Precision (AP) with 40 recall points for the Car category, using an IoU threshold of 0.7 for both 3D and bird's eye view (BEV) detection.

C. Comparison Results

To verify the effectiveness of our method, we present the evaluation results on the KITTI validation and test sets for 3D and BEV views in Table I and Table II. We compare our method with several recent methods, including LiDAR-based [1]–[4], [37], RGB-based [6]–[8] and multi-modal [9], [10], [38]. These baselines rank high on the KITTI benchmark. PointPillars [3] and PV-RCNN [2] are used as baseline detectors.

The results clearly indicate that our method significantly improves the detection performance of the baseline models. Specifically, when extremely sparse LiDAR data (*e.g.*, 1% density) is directly fed into LiDAR-based methods, the AP_{3D}

TABLE I
COMPARISON RESULTS OF AP_{3D}/AP_{BEV} ON KITTI VAL SET AT 1% DENSITY (CAR, IOU=0.7). BEST IN **BOLD**, SECOND IN UNDERLINE.

Method	Modality	AP_{3D}/AP_{BEV}		
		Easy	Moderate	Hard
MonoDGP [7]	Image	30.76 / 39.40	22.34 / 28.20	19.02 / 24.42
MonoDETR [6]	Image	28.84 / 37.86	20.61 / 26.95	16.38 / 22.80
MonoMAE [8]	Image	30.29 / 40.26	20.90 / 27.08	17.61 / 23.14
SECOND [4]	1% Points	19.54 / 32.36	12.25 / 20.26	10.22 / 16.89
Part-A^2 [37]	1% Points	11.34 / 17.03	8.07 / 12.17	6.78 / 10.18
PointPillars [3]	1% Points	19.82 / 34.41	12.87 / 21.79	10.66 / 18.20
PV-RCNN [2]	1% Points	23.69 / 34.69	15.39 / 22.00	12.67 / 18.12
Voxel-RCNN [1]	1% Points	23.14 / 31.67	14.61 / 19.69	11.93 / 16.12
EPNet [9]	Image + 1% Points	24.34 / 39.01	16.84 / 26.36	13.27 / 20.86
EPNet++ [10]	Image + 1% Points	19.94 / 33.21	12.60 / 20.66	9.95 / 16.35
Focal Conv [38]	Image + 1% Points	46.34 / 58.97	30.08 / 38.82	25.35 / 32.40
Ours + PointPillars	Image + 1% Points	46.21 / 59.33	27.86 / 37.84	22.19 / 31.31
Ours + PV-RCNN	Image + 1% Points	**51.59 / 61.93**	**33.03 / 41.42**	**27.02 / 34.72**

TABLE II
COMPARISON RESULTS OF AP_{3D}/AP_{BEV} ON KITTI TEST SET AT 1% DENSITY (CAR, IOU=0.7). BEST IN **BOLD**, SECOND IN UNDERLINE.

Method	Modality	AP_{3D}/AP_{BEV}		
		Easy	Moderate	Hard
MonoDGP [7]	Image	26.35 / 35.24	18.72 / 25.23	15.97 / 22.02
MonoDETR [6]	Image	25.00 / 33.60	16.47 / 22.11	13.58 / 18.60
MonoMAE [8]	Image	25.60 / 34.15	18.84 / 24.93	16.78 / 21.76
SECOND [4]	1% Points	14.39 / 26.72	8.51 / 15.15	6.83 / 12.25
Part-A^2 [37]	1% Points	8.78 / 15.20	5.72 / 9.50	4.47 / 7.45
PointPillars [3]	1% Points	16.33 / 28.95	9.69 / 16.68	7.47 / 13.18
PV-RCNN [2]	1% Points	19.69 / 29.78	11.67 / 17.19	9.06 / 13.44
Voxel-RCNN [1]	1% Points	18.38 / 26.44	10.78 / 15.38	8.13 / 11.96
EPNet [9]	Image + 1% Points	17.43 / 28.42	9.66 / 15.71	7.05 / 11.87
Focal Conv [38]	Image + 1% Points	40.99 / 52.92	23.09 / 31.57	18.52 / 25.88
Ours + PointPillars	Image + 1% Points	34.93 / 50.59	19.09 / 28.86	15.25 / 23.25
Ours + PV-RCNN	Image + 1% Points	**41.92 / 54.03**	**23.70 / 31.93**	**18.73 / 25.94**

values drop significantly. In contrast, our approach mitigates this issue by incorporating RGB data to enhance the detection process. When compared with purely RGB-based methods, our approach demonstrates considerable improvements in both AP_{3D} and AP_{BEV}. This improvement is attributed to the effective utilization of rich visual textures, which recover missing geometric structures and refine object edges within the sparse point clouds. Fig. 4 visually compares the detection performance of our method with PV-RCNN [2] as the baseline model under the extreme condition of 1% LiDAR density. The visual results show that our method not only achieves accurate localization of both near and distant vehicles but also reliably detects occluded objects that are partially invisible in the sparse LiDAR data.

D. Ablation Study

To verify the contribution of different components in our method, we conduct an ablation study comparing various fusion modules under 1% LiDAR density, as shown in Table III. Experiment (a) serves as the baseline, using raw 1% density points as input for PointPillars [3]. Due to the extreme sparsity, the baseline suffers severe performance degradation, achieving only 12.87% AP_{3D} in the moderate setting. In (b), we introduce an RGB-guided depth completion network

TABLE III
ABLATION STUDY OF DIFFERENT FEATURE FUSION STRATEGIES UNDER 1% LIDAR DENSITY.

Exp.	Method	AP_{3D} / AP_{BEV}		
		Easy	Moderate	Hard
(a)	PointPillars	19.82 / 34.41	12.87 / 21.79	10.66 / 18.20
(b)	+ Add + PL	40.26 / 56.89	23.99 / 35.96	19.55 / 29.36
(c)	+ Concat. + PL	42.42 / 57.89	25.35 / 36.40	20.73 / 29.52
(d)	+ Ours (w/o DFD) + PL	44.16 / 59.24	26.66 / 37.73	21.53 / 30.95
(e)	+ Ours (Full) + PL	**46.21 / 59.33**	**27.86 / 37.84**	**22.19 / 31.31**

that fuses RGB and LiDAR features via element-wise addition to predict dense depth, which is then back-projected to pseudo-LiDAR for detection. This strategy substantially recovers performance (+11.12% AP_{3D}) by incorporating essential semantic cues. However, this naive fusion remains suboptimal, as high-frequency RGB textures interfere with geometric representations. Channel-wise concatenation (c) further improves performance to 25.35% AP_{3D} by preserving modality-specific representations and alleviating cross-modal interference. Furthermore, the variant without the DFD module (d) falls short of the full model, indicating that frequency decoupling is essential for suppressing textural noise and maintaining geometric consistency. Finally, (e) uses the full

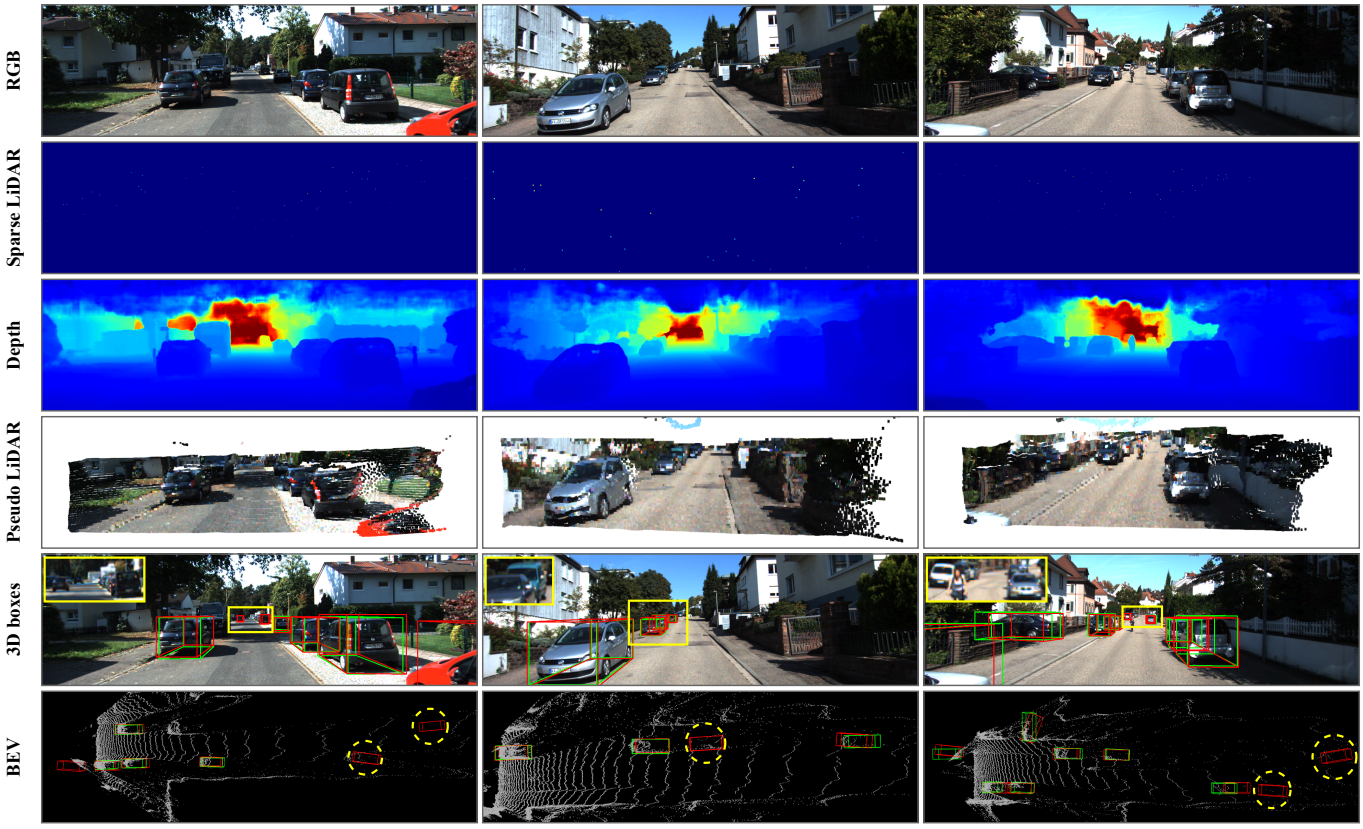

Fig. 4. Qualitative comparison on the KITTI validation set. Ground-truth boxes (green) and predictions (red) are shown for both 3D bounding boxes and bird's eye view (BEV).

DFG module and achieves the best performance of 27.86% AP_{3D} in the moderate setting. This confirms that decoupling high-frequency edge details from low-frequency geometric structure is crucial for accurate pseudo-LiDAR reconstruction and precise 3D detection.

E. Sparsity Robustness Analysis

To evaluate robustness against LiDAR degradation, we test densities from 1% to 12% using PointPillars as the baseline. As Table IV shows, the baseline suffers severe performance degradation under extreme sparsity (1%), dropping to 12.87% AP_{3D} in the moderate setting. This confirms that geometry-dependent networks struggle to capture object shapes when point clouds are insufficient. In contrast, our RGB-guided approach mitigates this issue, achieving a +14.99% increase in AP_{3D}. This shows that the DFG module effectively leverages visual textures to compensate for missing structural details.

However, as the LiDAR density increases to 10–12%, the performance gains of our method decrease, leading to a performance reversal in the hard setting. We attribute this failure mode to the inherent trade-off between the precision of pseudo-LiDAR and the reliability of real geometric information. In challenging scenarios involving distant or heavily occluded objects, the depth completion network inevitably introduces estimation noise. Although this noise remains tolerable under extremely sparse point-cloud conditions, increased density makes the relatively accurate real geometric cues susceptible to contamination by the noisy pseudo-LiDAR. Consequently, the resulting geometric ambiguity leads to a slight degradation in localization accuracy compared to PointPillars, which relies solely on sparse yet clean real geometric information.

TABLE IV
SPARSITY ROBUSTNESS COMPARISON UNDER DIFFERENT LIDAR DENSITIES.

Density	Method	AP_{3D} / AP_{BEV}		
		Easy	Moderate	Hard
1%	PointPillars	19.82 / 34.41	12.87 / 21.79	10.66 / 18.20
	+ Ours	**46.21 / 59.33**	**27.86 / 37.84**	**22.19 / 31.31**
6%	PointPillars	68.35 / 80.09	48.01 / 59.63	43.11 / 54.09
	+ Ours	**76.77 / 84.58**	**54.14 / 64.49**	**47.07 / 57.35**
8%	PointPillars	73.02 / 84.21	52.92 / 64.72	47.77 / 59.56
	+ Ours	**78.57 / 86.82**	**56.63 / 67.55**	**49.60 / 60.23**
10%	PointPillars	76.54 / 86.93	56.66 / 67.93	**51.79 / 64.19**
	+ Ours	**79.79 / 87.38**	**57.24 / 68.18**	51.25 / 61.86
12%	PointPillars	77.99 / 86.95	58.44 / 70.01	**53.94 / 66.35**
	+ Ours	**80.70 / 87.66**	**59.37 / 70.15**	52.48 / 63.28

V. CONCLUSION

In this paper, we propose a 3D object detection framework that fuses monocular RGB images with extremely sparse LiDAR data. The proposed Dynamic Frequency Guidance fusion design enables more reliable depth completion and

pseudo-LiDAR reconstruction. Experimental results on both the KITTI validation and test sets confirm that our method significantly improves detection accuracy under sparse sensing conditions. Furthermore, we analyze different density levels and demonstrate that our approach effectively alleviates performance degradation caused by point cloud sparsification.

REFERENCES

[1] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel r-cnn: Towards high performance voxel-based 3d object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 35, no. 2, 2021, pp. 1201–1209.

[2] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "Pv-rcnn: Point-voxel feature set abstraction for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10 529–10 538.

[3] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 697–12 705.

[4] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[5] J. Kim, B.-j. Park, and J. Kim, "Empirical analysis of autonomous vehicle's lidar detection performance degradation for actual road driving in rain and fog," *Sensors*, p. 2972, 2023.

[6] R. Zhang, H. Qiu, T. Wang, Z. Guo, Z. Cui, Y. Qiao, H. Li, and P. Gao, "Monodetr: Depth-guided transformer for monocular 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 9155–9166.

[7] F. Pu, Y. Wang, J. Deng, and W. Yang, "Monodgp: Monocular 3d object detection with decoupled-query and geometry-error priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 6520–6530.

[8] X. Jiang, S. Jin, X. Zhang, L. Shao, and S. Lu, "Monomae: Enhancing monocular 3d detection through depth-aware masked autoencoders," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 37, pp. 11 392–11 411, 2024.

[9] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 35–52.

[10] Z. Liu, T. Huang, B. Li, X. Chen, X. Wang, and X. Bai, "Epnet++: Cascade bi-directional fusion for multi-modal 3d object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 45, no. 7, pp. 8324–8341, 2022.

[11] M. Hu, S. Wang, B. Li, S. Ning, L. Fan, and X. Gong, "Penet: Towards precise and efficient image guided depth completion," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*. IEEE, 2021, pp. 13 656–13 662.

[12] J. Tang, F.-P. Tian, W. Feng, J. Li, and P. Tan, "Learning guided convolutional network for depth completion," *IEEE Transactions on Image Processing (TIP)*, vol. 30, pp. 1116–1129, 2020.

[13] Y. Wang, Y. Mao, Q. Liu, and Y. Dai, "Decomposed guided dynamic filters for efficient rgb-guided depth completion," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 34, no. 2, pp. 1186–1198, 2023.

[14] X. Cheng, P. Wang, and R. Yang, "Learning depth with convolutional spatial propagation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 2361–2379, 2019.

[15] X. Cheng, P. Wang, C. Guan, and R. Yang, "Cspn++: Learning context and resource aware convolutional spatial propagation networks for depth completion," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 10 615–10 622.

[16] Z. Xu, H. Yin, and J. Yao, "Deformable spatial propagation networks for depth completion," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 913–917.

[17] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[18] Y. Lin, T. Cheng, Q. Zhong, W. Zhou, and H. Yang, "Dynamic spatial propagation network for depth completion," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022, pp. 1638–1646.

[19] Z. Yan, Z. Wang, K. Wang, J. Li, and J. Yang, "Completion as enhancement: A degradation-aware selective image guided network for depth completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 26 943–26 953.

[20] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.

[21] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3dssd: Point-based 3d single stage object detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 040–11 048.

[22] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure aware single-stage 3d object detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 873–11 882.

[23] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 770–779.

[24] C. Wang, C. Ma, M. Zhu, and X. Yang, "Pointaugmenting: Cross-modal augmentation for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 794–11 803.

[25] T. Yin, X. Zhou, and P. Krähenbühl, "Multimodal virtual point 3d detection," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 16 494–16 507, 2021.

[26] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," in *Proceedings of the IEEE International Conference on Robotics and Automation, (ICRA)*. IEEE, 2019, pp. 7276–7282.

[27] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "Bevfusion: A simple and robust lidar-camera fusion framework," *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 10 421–10 434, 2022.

[28] X. Wu, L. Peng, H. Yang, L. Xie, C. Huang, C. Deng, H. Liu, and D. Cai, "Sparse fuse dense: Towards high quality 3d detection with depth completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5418–5427.

[29] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 8445–8453.

[30] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[31] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 7652–7660.

[32] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.

[33] J. Uhrig, N. Schneider, L. Schneider, U. Franke, T. Brox, and A. Geiger, "Sparsity invariant cnns," in *Proceedings of the IEEE International Conference on 3D Vision (3DV)*, 2017.

[34] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research (IJRR)*, 2013.

[36] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[37] S. Shi, Z. Wang, J. Shi, X. Wang, and H. Li, "From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 43, no. 8, pp. 2647–2664, 2020.

[38] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, "Focal sparse convolutional networks for 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 5428–5437.