Multi-Modal 3D Object Detection with Object Relation

Magdy Mahmoud
Chair of Robotics, Artificial Intelligence and Real-time Systems
TUM School of Computation, Information, and Technology
Munich, Germany
magdy.mahmoud@tum.de

***Abstract*—Accurate 3D object detection is crucial for autonomous driving, robotics, and spatial reasoning applications. While LiDAR provides precise geometric information, it lacks the semantic richness found in 2D images, which is essential for robust object recognition. Prior works have utilized inter-object relationships via Graph Neural Networks (GNN) but relied solely on LiDAR data, limiting their capability to fully leverage visual semantic cues.**

In this work, we propose a modular, multi-modal 3D object detection framework that integrates rich semantic information from 2D image features alongside LiDAR point clouds. Specifically, we extract image features using a self-supervised pretrained model (DINO) and investigate two flexible fusion strategies: (1) object-level fusion, combining 2D image features with LiDAR-based 3D proposal features to enhance object classification and localization, and (2) early fusion, embedding 2D image features earlier in the detection pipeline as a plug-and-play module suitable for various LiDAR-based detection architectures.

Our systematic evaluation demonstrates substantial improvements in detection accuracy, particularly in challenging classes. Compared to the baseline, our method achieves over 5% improvement on the car class and approximately 6% improvement on the pedestrian class. These results underscore the effectiveness of incorporating multi-modal fusion and relational reasoning, significantly enhancing 3D perception in complex driving scenarios.

I. Introduction

The rapid advancement of autonomous driving technologies has brought significant attention to 3D object detection, a core task necessary for enabling safe and reliable self-driving systems [1], [2]. Accurate detection and localization of objects such as vehicles, pedestrians, and obstacles are vital for real-time decision making and navigation. Beyond the automotive industry, 3D object detection also plays a crucial role in robotics [3], augmented reality [4], and medical imaging, where understanding spatial structures is essential.

Light Detection and Ranging (LiDAR) has emerged as one of the most important sensors for 3D object detection because of its ability to capture precise geometric information about the environment. By emitting laser pulses and measuring their return time, LiDAR generates point clouds that provide accurate depth and spatial data. Despite advancements in sensor technology, processing the wealth of 3D data captured by LiDAR sensors remains a formidable challenge. Traditional methods often struggle to extract meaningful information from the complex and high-dimensional point clouds generated by these sensors. Recent research has shown promising results in leveraging point clouds for various tasks, with approaches such as PointNet [5], PointNet++ [6], DGCNN [7], PointNext [8] and RepSurf [9] demonstrating the potential of point cloud processing in object detection, segmentation, and classification.

However, while LiDAR excels in spatial perception, it lacks the rich semantic details present in 2D visual data, such as color, texture, and appearance—critical elements for object recognition and classification. To bridge this gap, recent research has focused on multi-modal approaches that combine LiDAR point clouds with 2D image data from cameras, such as [10]–[12]. These approaches leverage the complementary strengths of both sensor modalities: LiDAR provides precise spatial structure, while images offer detailed semantic information.

A limitation of many existing 3D object detection frameworks is that they treat objects in isolation, neglecting contextual information provided by relationships between neighboring objects. However, inter-object interactions—such as the relative positioning of pedestrians and vehicles—can provide valuable cues for more accurate predictions. For instance, understanding patterns such as parallel parking on narrow streets or multi-lane traffic interactions can refine object motion and classification.

To address this, GraphRelate3D: Context-Dependent 3D Object Detection with Inter-Object Relationship Graphs [13], introduced an object relation module based on a message-passing Graph Neural Network (GNN) to explicitly model spatial and contextual relationships between detected objects. By representing objects as nodes and their interactions as edges in a graph, this approach propagated information between objects to refine their features based on surrounding context. However, this method relied solely on LiDAR data and did not incorporate multi-modal fusion with 2D image features. As a result, while it successfully captured object relationships, it lacked the ability to leverage rich semantic details from images that could further improve recognition and localization.

Building upon GraphRelate3D, we extend the 3D object detection framework to a multi-modal setting by integrating 2D image features alongside LiDAR data. Our approach utilizes self-supervised pretrained image models, specifically

the DINO model [14], to extract highly semantic-rich image features. We propose two flexible fusion strategies: an object-level fusion, where semantic-rich 2D image features are fused with LiDAR features at the object proposal stage, and an early fusion strategy, integrating 2D image features earlier in the detection pipeline to enrich the feature extraction process.

Our systematic evaluation reveals significant performance improvements, notably achieving almost 6% enhancement on the challenging pedestrian class, which often suffers from occlusions and limited geometric data. For the car class, we obtain improvements exceeding 5%, highlighting the effectiveness of our multi-modal integration strategy. However, limited performance gains on the cyclist class indicate that the current DINO model pretrained on general datasets may not have encountered sufficient cyclist examples. This suggests that additional pretraining or fine-tuning on autonomous driving-specific datasets could further enhance detection performance.

Overall, our modular and easily integrable multi-modal fusion and inter-object reasoning approach significantly advances 3D perception capabilities, offering promising improvements in real-world autonomous driving scenarios and beyond.

II. Related Works

A. 3D Object Detection

LiDAR point clouds provide rich spatial information for 3D object detection. Common approaches for extracting features from point clouds include voxelization [15], range-view [16], and bird's-eye view (BEV) methods [17]. PointPillars [15], a pioneering voxel-based method, efficiently organizes points into pillars but achieves limited accuracy. MSF [18] explores temporal information by utilizing sequential LiDAR frames. VoxelNext [19] proposes an efficient detection framework using sparse voxel features directly.

Inspired by advancements in 2D object detection, two-stage detectors such as PV-RCNN [20] enhance accuracy by employing multi-scale Region-of-Interest (RoI) feature abstraction. Similarly, PartA2 [21] utilizes a two-stage pipeline featuring part-aware and aggregation stages for improved precision. CenterPoint [22] adopts center-based detection refined by additional point features, achieving robust performance but still facing challenges in highly occluded scenarios.

B. GNNs in LiDAR Point Clouds

Graph Neural Networks (GNNs) excel in capturing relational information through message-passing between graph nodes [23], making them well-suited for representing complex spatial relationships. Several studies have leveraged GNNs for point cloud feature extraction. Bi et al. DGCNN [24] dynamically computes and updates graph features layer-wise, showing strong performance in classification and segmentation tasks. These methods highlight GNNs' potential in processing complex spatial structures within point clouds.

C. Graph-based 3D Object Detection

Motivated by the success of GNNs, recent works have explored their application specifically for 3D object detection in autonomous driving. Point-GNN [25] constructs graphs directly from irregular point clouds to achieve effective object detection. BADet [26] generates local graphs around object proposals to enhance boundary awareness. Ret3D [27] leverages transformer-based graph mechanisms to refine detection performance using sequential frames. Object-DGCNN [28] integrates bird's-eye view (BEV) features and queries for enhanced detection refinement. GACE [29] optimizes detection by aggregating point features from adjacent bounding boxes.

In particular, GraphRelate3D [13] introduces an effective object-relation graph using proposals within individual frames, refining object features by capturing spatial and contextual relationships through message-passing GNNs. Our approach extends GraphRelate3D by further integrating semantic-rich 2D image features, significantly enhancing detection performance and robustness, especially for challenging classes such as pedestrians.

D. Multi-modal Fusion for 3D Object Detection

Multi-modal fusion strategies combining LiDAR and camera data have been increasingly explored to exploit the complementary strengths of geometric and semantic information. MVX-Net [30] fuses voxel-based LiDAR features with image-based semantic features, demonstrating improved detection accuracy. Similarly, Box3D [11] and GAFusion [12] propose adaptive fusion methods to effectively integrate LiDAR and image modalities.

E. 2D Image Feature Extraction

Effective extraction of semantic-rich features from images is essential for enhancing multi-modal perception tasks. Traditional methods such as ResNet [31] have been widely used due to their reliable hierarchical feature extraction capabilities. Recently, transformer-based methods such as Swin Transformer [32] have shown superior performance by incorporating attention mechanisms with hierarchical structures, capturing richer contextual information.

Self-supervised learning has also revolutionized image feature extraction by enabling models to learn robust representations without explicit labels. Notably, DINO [14] and its successor DINOv2 [33] achieve remarkable performance, effectively capturing semantic structures from unlabeled datasets, making them highly valuable for downstream tasks including object detection and segmentation. Additionally, methods such as FeatUP [34], which enhance feature resolution through upsampling techniques, have demonstrated improved accuracy for detailed object representation. Our work leverages these advanced image feature extractors, primarily DINO, to significantly improve multi-modal detection performance, particularly for challenging categories like pedestrians.

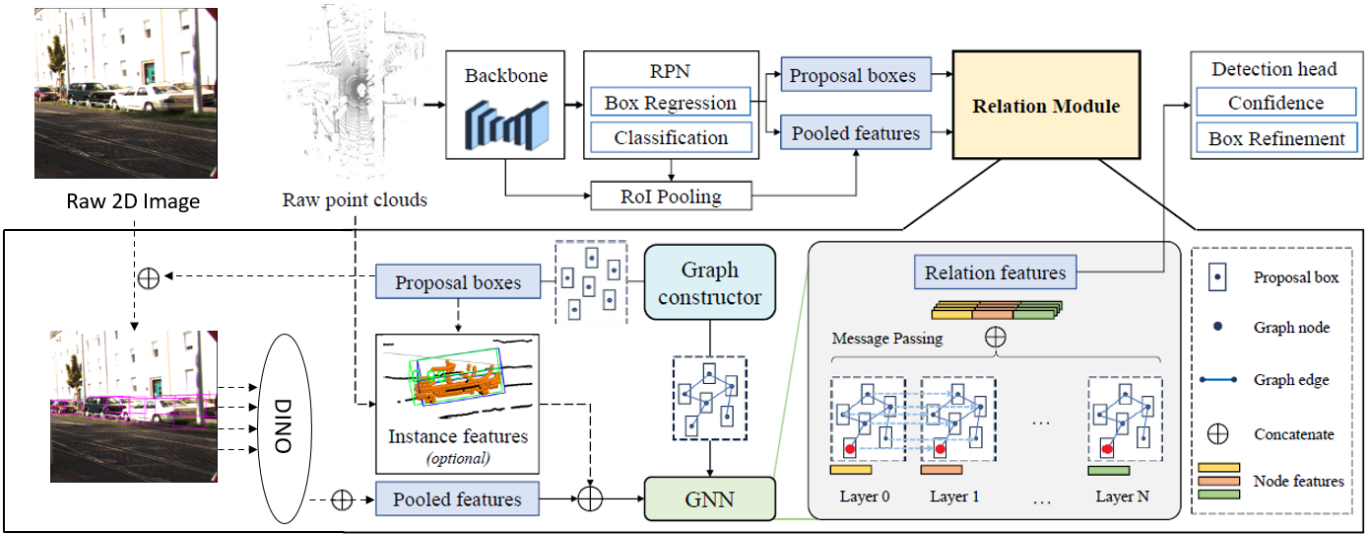

Fig. 1: Overview of the proposed multi-modal 3D object detection framework. The pipeline enhances LiDAR-based detection by integrating 2D image features and refining object proposals using a Graph Neural Network (GNN). The fusion of multi-modal features provides richer object representations, while the GNN models inter-object relationships to improve detection accuracy in complex scenes.

III. Methodology

A. Overview

Our framework builds upon a two-stage LiDAR-based 3D object detector by incorporating plug-and-play multi-modal fusion and inter-object reasoning modules to enhance detection accuracy. As illustrated in Fig. 1, we integrate 2D image features with LiDAR point cloud features, while modeling spatial and contextual relationships between object proposals using a Graph Neural Network (GNN).

In the first stage, a Region Proposal Network (RPN) generates coarse 3D object proposals from LiDAR data. These proposals are enriched with semantic 2D features extracted from RGB images, resulting in multi-modal representations combining precise geometry and rich semantic information.

In the second stage, we employ the plug-and-play GNN module from [13], which refines object proposals by capturing inter-object relationships. The GNN models spatial and contextual dependencies through iterative message passing, yielding discriminative and context-aware features for improved object detection.

Additionally, we explore an early fusion strategy as another flexible, plug-and-play module that integrates 2D image features earlier in the LiDAR-based pipeline, enhancing overall feature extraction capabilities.

The following subsections detail each component of our proposed methodology, highlighting their modular and easily integrable nature within general LiDAR-based detection architectures.

B. Graph Neural Network for Inter-Object Relationships

To refine feature representations and capture inter-object relationships, we integrate a GNN into the second stage of the detection pipeline. Following GraphRelate3D [13], we construct an inter-object graph where nodes represent 3D object proposals, each defined by their bounding boxes and initial feature vectors. Edges between nodes are established based on spatial proximity, using either k-nearest neighbors (KNN) or a radius-based approach.

Node features are initialized by combining the proposal's RoI features and bounding box attributes via an MLP:

$$v_i^{(0)} = \text{MLP}([f_i, b_i]). \tag{1}$$

During message passing, nodes aggregate features from neighbors. Edge features between nodes i and j are defined as:

$$e_{ij}^{(l)} = \text{MLP}([v_j^{(l)} - v_i^{(l)}, b_j - b_i]), \tag{2}$$

with node features updated by:

$$v_i^{(l+1)} = \max_{j \in \mathcal{N}(i)} \text{MLP}([e_{ij}^{(l)}, v_i^{(l)}]). \tag{3}$$

After L iterations, node features from all layers are concatenated to form refined features used for improved object predictions. The modular nature of this GNN module allows seamless integration into various detection networks.

C. Fusion of 2D Image Features with Object Proposals

To enhance the feature representations of object proposals, we integrate rich semantic information from 2D images with geometric information provided by LiDAR point cloud data. This fusion process leverages the strengths of both modalities by first extracting semantically meaningful features using a self-supervised DINO model and subsequently aligning and combining these features with corresponding 3D object proposals. By carefully aligning the extracted image features to their associated 3D bounding boxes, we generate enriched

multi-modal representations that improve object detection performance, especially in challenging scenarios involving occlusions, sparse data, or complex environments. The detailed steps of the 2D feature extraction and fusion methods are elaborated in the following subsections.

1) 2D Feature Extraction: To extract meaningful semantic information from RGB images, we utilize a self-supervised 2D image feature extractor, specifically DINO [14]. Models trained with self-supervised approaches like DINO have demonstrated a strong capability in extracting high-level semantic information without relying on explicit object-level annotations. By learning visual representations directly from unlabeled data, these models effectively encode the structure, texture, and prominent objects within images, making their features highly valuable for downstream tasks such as object detection, segmentation, and scene understanding.

In our main approach, for each 3D object proposal obtained from LiDAR, we first project the corresponding 3D bounding box onto the 2D image plane, following the method outlined by MVX-Net [30]. Specifically, given a 3D bounding box in LiDAR coordinates:

$$b_{\text{3D}} = [x, y, z, d_x, d_y, d_z, \theta], \tag{4}$$

we first convert it from LiDAR coordinates to camera coordinates using the calibration matrix $\mathbf{T}_{\text{cam}} \in \mathbb{R}^{4\times4}$:

$$b_{\text{cam}} = \mathbf{T}_{\text{cam}} \cdot [x, y, z, 1]^T. \tag{5}$$

Next, the 3D bounding box in camera coordinates is projected onto the 2D image plane using the intrinsic calibration matrix $\mathbf{K} \in \mathbb{R}^{3\times4}$:

$$b_{\text{img}} = \mathbf{K} \cdot b_{\text{cam}}, \tag{6}$$

yielding the vertices of the projected 3D bounding box. From these projected vertices, we compute the tightest enclosing 2D bounding box b_{2D} defined by pixel coordinates:

$$b_{\text{2D}} = [u_{\min}, v_{\min}, u_{\max}, v_{\max}], \tag{7}$$

where $(u_{\min}, v_{\min})$ and $(u_{\max}, v_{\max})$ represent the top-left and bottom-right corners of the box, respectively.

For each obtained 2D bounding box, we crop the corresponding image region from the original RGB image and extract localized features using DINO. Specifically, we extract the *class token*, a special feature vector produced by DINO, which provides a concise yet semantically rich representation of the entire cropped image region. For the DINO ViT-base (DINO_b16) model, this class token is a 768-dimensional vector encapsulating extensive contextual and semantic information, thus providing a robust representation aligned closely with each 3D proposal.

In addition to this main approach, we also conducted an alternative experiment to explore the potential benefits of early feature extraction. In this method, we first pass the entire image through DINO to obtain global image-level features, resulting in a low-resolution feature map (e.g., 14x14 spatial dimensions for DINO_b16 with 768 feature channels). Subsequently, we experiment with different upsampling techniques, such as bilinear interpolation and feature upsampling modules like FeatUp [34], to achieve higher-resolution feature maps. We then crop proposal-specific regions from these enhanced global feature maps based on the projected 2D bounding boxes b_{2D} and reduce each proposal's region into a single representative feature vector using aggregation methods like max-pooling, average-pooling, or a learned multi-layer perceptron (MLP).

Both feature extraction methods leverage the semantic richness of DINO-derived representations, enabling effective integration with the geometric information provided by LiDAR data, as described in the subsequent fusion step.

2) Fusion of Semantic 2D and Geometric 3D Features: After extracting the 2D features from image crops corresponding to each projected 3D bounding box, we fuse these 2D features with their corresponding LiDAR-based 3D features. Specifically, for each object proposal, we concatenate the extracted DINO class token vector, representing semantic information of the image region, with the original LiDAR-based 3D proposal feature. This concatenation combines both semantic richness from images and geometric precision from LiDAR, resulting in a comprehensive representation of each object proposal:

$$\mathbf{f}\text{fused} = [\mathbf{f}3\text{D}, \mathbf{f}_{\text{2D-class-token}}], \tag{8}$$

where $\mathbf{f}3\text{D}$ denotes the original LiDAR feature vector and $\mathbf{f}2\text{D-class-token}$ is the DINO class token vector.

The concatenation of these features results in increased dimensionality. To manage this dimensionality increase, we experimented with two approaches: (1) adapting subsequent network layers to directly handle the higher-dimensional input, which led to increased computational costs and longer inference times, and (2) applying a multi-layer perceptron (MLP) to project the fused features into a lower-dimensional space, maintaining computational efficiency while preserving essential semantic and geometric information:

$$\mathbf{f}\text{final} = \text{MLP}(\mathbf{f}\text{fused}). \tag{9}$$

In our alternative experiment, where we initially extract global image-level feature maps, we similarly concatenate cropped regional feature maps with LiDAR-based features. In this scenario, after cropping and aggregating spatial features into a single representative vector (via pooling or an additional MLP), we follow the same fusion and dimensionality reduction approaches as described above.

The resulting object-level features, which now contain both 2D image and LiDAR point cloud information, are then passed to the GNN module described in Section III-B. This allows the GNN to model inter-object relationships using a richer set of features, further improving the detection accuracy.

D. Early Fusion of 2D and 3D Features

In addition to the object-level fusion approach, we propose an early fusion method as a complementary, plug-and-play fusion module that can be integrated independently or alongside

the Graph Neural Network-based module with fused object proposals, offering enhanced flexibility for general LiDAR-based detection frameworks.

Specifically, within the PV-RCNN framework, we experimented with two early fusion strategies. In the first strategy, we incorporated 2D image features directly into the voxel set abstraction (VSA) module. The VSA module aggregates multi-scale semantic features from 3D CNN feature volumes around each keypoint p_i by applying a set abstraction operation. This operation gathers neighboring voxel-wise features $f_j^{(lk)}$ within a radius r_k, combined with their relative positional information $v_j^{(lk)} - p_i$, as:

$$S_i^{(lk)} = \left\{ \left[f_j^{(lk)}; v_j^{(lk)} - p_i \right] \mid \|v_j^{(lk)} - p_i\|_2 < r_k \right\}. \quad (10)$$

These voxel-wise features are processed through a multi-layer perceptron (MLP) and aggregated via a max-pooling operation. To integrate image-based information, we expanded this feature aggregation by adding extracted 2D features f_i^{2D} as additional inputs, enabling the MLP to jointly encode LiDAR and image modalities:

$$f_i^{\text{fused}} = \text{MLP}(S_i^{(lk)}, f_i^{2D}). \quad (11)$$

In the second strategy, we introduced the 2D image features after the VSA module's initial MLP transformation. We tested two methods within this approach: (1) directly concatenating the output features from the VSA module f_i^{VSA} with the 2D image features followed by an MLP for dimensionality adjustment,

$$f_i^{\text{final}} = \text{MLP}([f_i^{VSA}, f_i^{2D}]), \quad (12)$$

and (2) directly adding (element-wise summation) the VSA output features with the 2D features:

$$f_i^{\text{final}} = f_i^{VSA} + f_i^{2D}. \quad (13)$$

These experiments aimed to explore the most effective and computationally efficient way to integrate rich semantic information from images at an early stage of the detection pipeline. Through these early fusion experiments, we provide additional insights into multi-modal integration strategies that can further improve detection accuracy, offering flexible, modular solutions applicable to various LiDAR-based detection architectures.

By combining 2D image features, LiDAR point clouds, and inter-object relationships, our framework represents a significant step forward in multi-modal 3D object detection, offering improved accuracy and robustness for autonomous driving and related applications.

IV. Experiments and Results

In this section, we introduce the datasets and models used in our experiments. Then, we explain the evaluation metrics and experimental setup and analyze quantitative results in depth.

A. Datasets

We utilize the KITTI 3D Object Detection dataset [35], a fundamental dataset widely used in autonomous driving, to train and evaluate our proposed inter-object relationship method. The KITTI dataset consists of 7,481 annotated LiDAR scans sampled from different driving scenes. We utilize the common train-val split with 3712 training and 3769 validation samples to train and evaluate the baseline and our model. We also test the effectiveness of our approach on the official test set, which includes 7518 samples.

Evaluation Metrics. The KITTI 3D average precision (AP) and Bird's-Eye-View (BEV) AP metrics are used in our experiments. Following the usual setting, we leverage the IoU threshold to 0.7 for the car class and 0.5 for pedestrians and cyclists under three difficulty levels (easy, moderate, and hard). All AP values are calculated with 40 recall positions on the validation set and the official test server.

B. Experimental Setup

We conduct all experiments using the OpenPCDet repository based on PyTorch with one NVIDIA GeForce 4080 (16GB) GPU.

Hyperparameters. For all experiments, we set the batch size to 2, using the Adam OneCycle optimizer with an initial learning rate of 0.01. All models are trained for 80 epochs, except for ablation study, and we report the best result of each experiment. Network Architecture. We use PV-RCNN [20] as the baseline models and implement our proposed modules in them. For the proposed module, we apply KNN (k = 16) as the graph generator for the main experiments and use a four-layer GNN to extract features. To keep the feature dimension alignment, we set the input and output of the GNN module to 256, the same as the dimensions of the pooled features and the input features of the detection head of the two-stage 3D detectors.

Training Loss. Our proposed framework does not introduce external loss calculation, making our module easily adapted to any other two-stage detectors. Therefore, in our experiments, we follow the default loss function of PV-RCNN to conduct experiments.

Data Augmentation. We utilize four widely adopted data augmentation strategies in our experiments:

- Ground truth sampling [36], randomly selecting several ground truths from other scenes and adding them into the current frame.
- Random flipping along the x-axis.
- Random rotation within the angle range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ around the z-axis.
- Random scaling with a scaling factor within the range $[0.95, 1.05]$.

C. Results and Discussion

Table I summarizes the performance comparison between our proposed multi-modal fusion method and the baseline on the KITTI validation set. Our approach demonstrates significant improvements across both car and pedestrian categories,

clearly highlighting the benefit of integrating rich semantic information from 2D image features into the 3D LiDAR detection pipeline.

For the car class, our method shows moderate yet consistent improvement, particularly notable in the moderate difficulty setting, where we achieve a substantial gain of 5.41% over the baseline. This improvement is crucial, as the moderate difficulty setting closely represents typical driving scenarios encountered in practical autonomous driving environments.

The pedestrian class results exhibit even more substantial improvements, which is particularly encouraging given that pedestrians are among the most challenging classes to detect accurately in the KITTI dataset due to their smaller size and frequent occlusions. Our method achieves improvements of 2.91%, 3.45%, and 4.20% in Easy, Moderate, and Hard categories, respectively. Furthermore, evaluating pedestrian performance using the AP_{R40}@0.50,0.25,0.25 metric further emphasizes this success, with enhancements ranging from 3.85% (Easy) to an impressive 5.96% (Moderate).

In contrast, the cyclist class did not exhibit significant improvement in our experiments. This might be attributed to the fact that the DINO model used in our pipeline was pretrained primarily on general-purpose datasets lacking sufficient representation of cyclists, commonly seen in autonomous driving contexts. Therefore, additional pretraining or fine-tuning on datasets specifically tailored for autonomous driving scenarios, such as nuScenes or Waymo Open, might further improve the cyclist detection performance.

Overall, the presented results represent the best configuration we obtained after systematically evaluating various fusion methods and feature extraction settings. This confirms that our proposed multi-modal approach not only significantly boosts detection performance for challenging classes but also demonstrates strong generalization potential for real-world autonomous driving scenarios.

D. Ablation Studies

In this section, we summarize different settings and configurations evaluated during our experimentation.

Different 2D Feature Extractors. We compared various backbones for extracting 2D features, including DINO variants (ViT-B/8, ViT-B/16, DINO v2), FeatUP ResNet, and Swin Transformer. The results varied slightly across different classes, with the DINO ViT-B/8 backbone consistently achieving the highest performance.

Feature Fusion Methods. During the fusion of 2D and 3D features, we experimented with several strategies, including direct summation, concatenation followed by an MLP for dimensionality reduction, and concatenation with subsequent network layers adapted for increased dimensionality. The best performance was observed when features were directly concatenated, and subsequent layers were adjusted to handle the higher dimensionality.

Early Fusion Strategies. For early fusion, we evaluated incorporating image features before and after the voxel set abstraction (VSA) module. The most effective approach integrated image features directly into the VSA feature aggregation list, allowing the VSA's internal MLP to jointly process both LiDAR and image-based inputs.

Training Configuration. Extensive experimentation with different epoch counts and hyperparameters indicated that training for 80 epochs using the default hyperparameters from GraphRelate3D [13] provided the best results in terms of detection accuracy and stability.

V. Conclusion

In this work, we presented a modular and lightweight multi-modal 3D object detection framework that effectively combines LiDAR point clouds with rich semantic information extracted from 2D images, incorporating inter-object relationships through a Graph Neural Network (GNN). Our proposed fusion strategies—object-level fusion and early fusion—successfully integrate the complementary strengths of geometric and semantic data, significantly enhancing the accuracy and robustness of 3D object detection. The integration of relational reasoning via the GNN further enables the model to accurately interpret complex spatial and contextual interactions, proving especially beneficial in challenging scenarios such as pedestrian detection.

Experimental results demonstrated substantial improvements, notably achieving over 5% enhancement on the car class and approximately 6% on the pedestrian class compared to baseline approaches. However, limited improvement observed in the cyclist class suggests the potential benefit of further pretraining on autonomous driving-specific datasets.

Moving forward, we plan to refine our plug-and-play fusion modules to test their adaptability and compatibility with a broader range of LiDAR-based detection architectures, enhancing generalizability and ease of integration. Additionally, future efforts will focus on optimizing the framework for real-time deployment, exploring further semantic feature extraction strategies, and evaluating performance across diverse sensor configurations and environmental conditions.

References

[1] P. Ambati, M. F. Hashmi, and A. Gupta, "Deep learning frontiers in 3d object detection: A comprehensive review for autonomous driving," *IEEE Access*, vol. PP, pp. 1–1, 01 2024.

[2] B. Udugama, "Review of deep reinforcement learning for autonomous driving," 2023.

[3] J. Mao, S. Shi, X. Wang, and H. Li, "3d object detection for autonomous driving: A comprehensive survey," 2023.

[4] M. Billinghurst, A. Clark, and G. Lee, "A survey of augmented reality," *Foundations and Trends® in Human-Computer Interaction*, vol. 8, pp. 73–272, 01 2015.

[5] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[7] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

TABLE I: Comparison with baseline on KITTI validation set. For cars and cyclists, IoU thresholds are 0.7 and 0.5 respectively. Metrics include standard AP and AP_{R40} at 0.50, 0.25, 0.25.

	Car 3D AP (%) ↑			Pedestrian 3D AP (%) ↑			Pedestrian AP_{R40}@0.50,0.25,0.25 (%) ↑		
Method	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Baseline	89.43	79.16	78.37	64.19	57.03	51.55	78.49	71.94	67.91
Ours (ViT-B/8)	**89.46**	**84.57**	**78.89**	**67.10**	**60.48**	**55.75**	**82.34**	**77.90**	**73.52**
Improvement	+0.03	+5.41	+0.52	+2.91	+3.45	+4.20	+3.85	+5.96	+5.61

[8] G. Qian, Y. Li, H. Peng, J. Mai, H. Hammoud, M. Elhoseiny, and B. Ghanem, "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23192–23204, 2022.

[9] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18942–18952, 2022.

[10] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3d lidar and camera data for object detection in autonomous vehicle applications," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 4901–4913, 2020.

[11] M. A. V. Saucedo, N. Stathoulopoulos, V. Sumathy, C. Kanellakis, and G. Nikolakopoulos, "Box3d: Lightweight camera-lidar fusion for 3d object detection and localization," 2024.

[12] X. Li, B. Fan, J. Tian, and H. Fan, "Gafusion: Adaptive fusing lidar and camera with multiple guidance for 3d object detection," 2024.

[13] M. Liu, E. Yurtsever, M. Brede, J. Meng, W. Zimmer, X. Zhou, B. L. Zagar, Y. Cui, and A. Knoll, "Graphrelate3d: Context-dependent 3d object detection with inter-object relationship graphs," 2024.

[14] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021.

[15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," 2019.

[16] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," 2019.

[17] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. Waslander, "Joint 3d proposal generation and object detection from view aggregation," 2018.

[18] C. He, R. Li, Y. Zhang, S. Li, and L. Zhang, "Msf: Motion-guided sequential fusion for efficient 3d object detection from point cloud sequences," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5196–5205, June 2023.

[19] Y. Chen, J. Liu, X. Zhang, X. Qi, and J. Jia, "Voxelnext: Fully sparse voxelnet for 3d object detection and tracking," 2023.

[20] 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," 2021.

[21] 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," 2020.

[22] T. Yin, X. Zhou, and P. Krähenbühl, "Center-based 3d object detection and tracking," 2021.

[23] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[24] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," 2019.

[25] W. Shi, Ragunathan, and Rajkumar, "Point-gnn: Graph neural network for 3d object detection in a point cloud," 2020.

[26] R. Qian, X. Lai, and X. Li, "Badet: Boundary-aware 3d object detection from point clouds," *Pattern Recognition*, vol. 125, p. 108524, May 2022.

[27] Y.-H. Wu, D. Zhang, L. Zhang, X. Zhan, D. Dai, Y. Liu, and M.-M. Cheng, "Ret3d: Rethinking object relations for efficient 3d object detection in driving scenes," 2022.

[28] Y. Wang and J. Solomon, "Object dgcnn: 3d object detection using dynamic graphs," 2021.

[29] D. Schinagl, G. Krispel, C. Fruhwirth-Reisinger, H. Possegger, and H. Bischof, "Gace: Geometry aware confidence enhancement for black-box 3d object detectors on lidar-data," 2023.

[30] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," 2019.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[32] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[33] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024.

[34] S. Fu, M. Hamilton, L. Brandt, A. Feldman, Z. Zhang, and W. T. Freeman, "Featup: A model-agnostic framework for features at any resolution," 2024.

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, 2012.

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