3D Dual-Fusion: Dual-Domain Dual-Query Camera-LiDAR Fusion for 3D Object Detection

Yecheol Kim\textsuperscript{1}, Konyul Park\textsuperscript{1}, Minwook Kim\textsuperscript{1}, Dongsuk Kum\textsuperscript{2}, and Jun Won Choi\textsuperscript{1}
Hanyang University\textsuperscript{1}
Korea Advanced Institute of Science & Technology (KAIST)\textsuperscript{2}

Abstract—Fusing data from cameras and LiDAR sensors is an essential technique to achieve robust 3D object detection. One key challenge in camera-LiDAR fusion involves mitigating the large domain gap between the two sensors in terms of coordinates and data distribution when fusing their features. In this paper, we propose a novel camera-LiDAR fusion architecture called, 3D Dual-Fusion, which is designed to mitigate the gap between the feature representations of camera and LiDAR data. The proposed method fuses the features of the camera-view and 3D voxel-view domain and models their interactions through deformable attention. We redesign the transformer fusion encoder to aggregate the information from the two domains. Two major changes include 1) dual query-based deformable attention to fuse the dual-domain features interactively and 2) 3D local self-attention to encode the voxel-domain queries prior to dual-query decoding. The results of an experimental evaluation show that the proposed camera-LiDAR fusion architecture achieved competitive performance on the KITTI and nuScenes datasets, with state-of-the-art performances in some 3D object detection benchmark categories.

Index Terms—3D object detection, sensor fusion, transformer, autonomous driving, deep learning

I. INTRODUCTION

A 3D object detector identifies the presence, location, and class of an object in a 3D world coordinate system based on sensor data. Both cameras and light detection and ranging (LiDAR) sensors provide useful information for 3D object detection. These two sensors have significantly different characteristics, because they use different physical sources and measurement processes. Camera sensors provide dense visual information on objects such as their color, texture, and shape, whereas LiDAR sensors produce accurate but relatively sparse range measurements. Consequently, these sensors exhibit different behavior and performance characteristics depending on the scene and object conditions. Camera-LiDAR sensor fusion aims to combine the complementary information provided by the two sensing modalities to achieve robust 3D object detection.

Yecheol Kim and Jun Won Choi are with Department of Electrical Engineering, Hanyang University, 04753 Seoul, Republic of Korea. (e-mail: yckim@spa.hanyang.ac.kr, junwchoi@hanyang.ac.kr) (Corresponding author: Jun Won Choi)

Konyul Park and Minwook Kim are with Department of Artificial Intelligence, Hanyang University, 04753 Seoul, Republic of Korea. (e-mail: {konyulp, minwookkim}@spa.hanyang.ac.kr)

Dongsuk Kum is with the Graduate School of Mobility, Korea Advanced Institute of Science and Technology (KAIST), 34141 Daejeon, Republic of Korea, (email: dskum@kaist.ac.kr)

Recently, deep neural network (DNN) models have achieved considerable success in 3D object detection tasks, and numerous DNN architectures have been developed for LiDAR-based 3D object detection. These detectors encode LiDAR point clouds using backbone networks and detect objects based on the features obtained from the encoder. Voxel-based encoding is a widely used LiDAR encoding method, that voxelizes LiDAR points in 3D space and encodes the points in each voxel \cite{1, 2}. Camera images also provide useful information for 3D object detection. Visual features obtained by applying convolutional neural network (CNN) model to camera images can be used to perform 3D object detection as well \cite{3}. To leverage diverse information provided by two multiple sensors, a feature-level fusion strategy has been proposed, which uses both voxel features and camera features together to perform 3D object detection. However, these methods involve the challenge that two features are represented in different coordinate domains (i.e., camera-view versus voxel domains), so one of the coordinate representations must be aligned and adapted into another without losing information of the original domains. Therefore, the development of method to reduce...
such a domain gap in aggregating dual-domain features is a key to improve the performance of camera-LiDAR sensor
fusion methods.

Existing camera-LiDAR fusion methods used various do-
main transformation strategies at the point level, feature
level, and proposal level. Point-level fusion methods [4]–[6]
projected the semantic information obtained from a camera
image into the LiDAR points in 3D space and combined the
data with point-wise LiDAR features. One limitation of these
methods is that LiDAR data do not participate in fusion at
the same semantic level as the camera features. Proposal-
level fusion methods [9]–[15] generated 2D detection proposals
from a camera image and associated LiDAR points with each
proposal. Then, the camera features and LiDAR features were
fused to refine each proposal. However, the performance of
these methods was limited by the accuracy of the generated
proposals. Feature-level fusion methods [9]–[15] are designed
to extract semantic features separately from the camera image
and LiDAR data and aggregate them in the voxel domain.

In this paper, we aim to reduce the domain gap between
the camera features and LiDAR features to boost the effect
of sensor fusion. We propose a new camera-LiDAR fusion
architecture, referred to as 3D Dual-Fusion, for 3D object
detection. The key idea of the proposed approach is dual-

B. Camera-LiDAR Fusion for 3D Object Detection

To date, various camera-LiDAR sensor fusion methods have
been proposed to achieve robust 3D object detection. These
methods can be roughly categorized into point-level, feature-
level, and proposal-level fusion methods.

Point-level fusion augments the semantic information ob-
tained from a camera image to point-wise features extracted
from LiDAR data. MVXNet [4] transformed the visual fea-
tures extracted from a 2D object detector, while PointPainting
[5] and FusionPainting [24] transferred the semantic segmen-
tation masks to the LiDAR points. Proposal-level fusion relies on the detection proposals obtained by processing a single sensor to achieve sensor
fusion. RoarNet [25], F-PointNet, [8] and PointFusion [7]
predicted 2D detection proposals based on a camera image
and associated the corresponding LiDAR points using proposal
aggregation through deformable attention.

We leverage the sparsity of LiDAR points to reduce the
computational complexity and memory usage of 3D Dual-
Fusion. We assign dual-queries only for the non-
empty voxels and the corresponding pixels of the camera
features projected from those voxels. Because only a small fraction of voxels are non-empty, the number of
queries used for dual-domain interactive feature fusion is
much smaller than the size of the entire voxel. Further-
more, adopting the structure of deformable attention, the
3D Dual-Fusion architecture is lightweight compared to
global-scale attention in the Transformer architecture.

When combined with the TransFusion detection head
[14], the proposed 3D Dual-Fusion method achieves state-of-the-art performance on some categories of KITTI
and nuScenes benchmarks.

The source codes used in this work will be released
publicly.

II. RELATED WORK

A. LiDAR-only 3D Object Detection

Numerous architectures have been proposed to perform
3D object detection based on LiDAR point clouds. Two
well-known backbone networks have been developed to
encode LiDAR data, including voxel-based [1], [2], [18]
and PointNet++-based backbones [19]–[21]. The voxel-based
backbone partitions the LiDAR points using a voxel or a
pillar structure and encodes the points of each grid element.
In contrast, the PointNet++-based backbone groups the points
using the farthest point sampling algorithm and increases the
semantic level of the point-wise features in a hierarchical
manner. Recently, Transformer models have been used to
encode LiDAR point clouds [22], [23].
where \(m\) indexes the attention head, \(k\) indexes the sampled keys, and \(K\) is the total number of the sampled keys. \(W_m \in \mathbb{R}^{C \times C/M}\) and \(W_{mk}^r \in \mathbb{R}^{C/M \times C}\) denote learnable projection matrices. The attention weight \(A_{mk}\) predicted from the query feature \(z_q\) is in the range \([0, 1]\), and \(\sum_{k=1}^{K} A_{mk} = 1\). \(\Delta p_{mk}\) is the predicted offset from the reference point, and \(F_c(p_q + \Delta p_{mk})\) is the input feature at the \(p_q + \Delta p_{mk}\) location. In this study, we tailor deformable DETR to support the proposed dual-domain fusion of camera and LiDAR features.

C. Review of Deformable DETR

Deformable DETR [30] realized Transformer attention [31] at low computational complexity using a local attention mechanism. Deformable attention only attends to a small set of key sampling points around a reference point. For the sake of brevity, we skip the multi-scale term of deformable kernels. Given an input feature map \(F_c \in \mathbb{R}^{H \times W \times C}\), let \(q\) index a query with feature \(z_q\) and the reference point \(p_q\). The deformable attention is formulated by

\[
\text{DefAttn}(z_q, p_q, F_c) = \sum_{m=1}^{M} W_m \sum_{k=1}^{K} A_{mk} \cdot W_{mk}^r F_c(p_q + \Delta p_{mk}),
\]

where \(m\) indexes the attention head, \(k\) indexes the sampled keys, and \(K\) is the total number of the sampled keys. \(W_m \in \mathbb{R}^{C \times C/M}\) and \(W_{mk}^r \in \mathbb{R}^{C/M \times C}\) denote learnable projection matrices. The attention weight \(A_{mk}\) predicted from the query feature \(z_q\) is in the range \([0, 1]\), and \(\sum_{k=1}^{K} A_{mk} = 1\). \(\Delta p_{mk}\) is the predicted offset from the reference point, and \(F_c(p_q + \Delta p_{mk})\) is the input feature at the \(p_q + \Delta p_{mk}\) location. In this study, we tailor deformable DETR to support the proposed dual-domain fusion of camera and LiDAR features.

III. 3D Dual-Fusion

In this section, we present the details of 3D Dual-Fusion.

A. Overall Architecture

The overall architecture of the proposed 3D Dual-Fusion is illustrated in Fig 2. The camera image \(X_c \in \mathbb{R}^{W_c \times H_c \times C}\) is encoded by the standard CNN backbone, where \(W_c\) and \(H_c\) denote the width and height of the camera image, respectively. In multi-view camera setup, multiple camera images are encoded by CNN separately. LiDAR point clouds \(X_v \in \mathbb{R}^{N \times 3}\) are also encoded by voxel encoding backbone, where \(N\) is the number of LiDAR points. We voxelize the LiDAR points using the voxel grid structure of the width \(W_v\), length \(L_v\), and height \(H_v\). The structure of the voxel encoding backbone is adopted from [2]. We also use the CNN backbone of DeepLabV3 [32]. Voxel encoding backbone network produces the voxel-domain LiDAR features \(F_v \in \mathbb{R}^{W_v \times L_v \times H_v \times C_v}\). The CNN backbone network produces the camera-domain image features \(F_c \in \mathbb{R}^{W_c/8 \times H_c/8 \times C_c}\).

The voxel-domain LiDAR features \(F_v\) and camera-domain image features \(F_c\) are fed into the Dual-Fusion Transformer. Before that, the camera-domain image features \(F_c\) are enhanced by an adaptive gated fusion network (AGFN). AGFN projects the voxel-domain LiDAR features to the camera domain and combines them with the camera-domain features. We denote the features enhanced by AGFN as \(F_c'\). Dual-Fusion Transformer applies 3D local self-attention (3D-LSA) and dual-query deformable attention (DDA) to fuse \(F_c\) and \(F_v\). The voxel-domain features in non-empty voxels are first encoded by 3D local self-attention. Then, DDA performs simultaneous feature fusion in both camera-view and voxel domains through cross attention with dual-queries. After multiple attention layers, the final voxel-domain features are transformed into bird’s eye view (BEV) features [2]. BEV features are passed through the detection head to generate a 3D bounding box and classification score. Note that the entire network of 3D Dual-Fusion is end-to-end trainable.
where \( W \) query, key and value, respectively, \( l \) updates the c-queries \( \{q_{c,1}, q_{c,2}, \ldots, q_{c,Q}\} \) are assigned to the non-empty voxels and used to refine their voxel-domain features. c-queries \( q_c = \{q_{c,1}, q_{c,2}, \ldots, q_{c,Q}\} \) are assigned to the image pixels indicated by projecting the center points of the non-empty voxels into the camera domain and used to refine the image-domain features.

### B. Dual-Query

The dual queries consist of a camera-query (c-query) and voxel-query (v-query). Suppose that the LiDAR voxelization step yields \( Q \) non-empty voxels, where \( Q \) can vary depending on the distribution of the input point clouds. v-queries \( q_v = \{q_{v,1}, q_{v,2}, \ldots, q_{v,Q}\} \) are assigned to the non-empty voxels and used to refine their voxel-domain features. c-queries \( q_c = \{q_{c,1}, q_{c,2}, \ldots, q_{c,Q}\} \) are assigned to the image pixels indicated by projecting the center points of the non-empty voxels into the camera domain and used to refine the image-domain features.

### C. 3D Local Self-Attention

In the beginning, v-queries are initialized with the LiDAR features in non-empty voxels. The self-attention layer successively encodes v-queries by modeling their spatial relationships. Because global self-attention is computationally intensive, 3D local self-attention is devised to reduce the scope of attention within a local region. The non-empty voxels are clustered into local regions by applying the farthest point sampling algorithm \([19]\) to their center points. We find \( K \) voxels within a fixed radius around each centroid of each local region. Let \( F = \{f_i|i \in N(x_{ct})\} \) and \( X = \{x_i|i \in N(x_{ct})\} \) be a set of query features and 3D positions assigned to the centroid \( x_{ct} \), respectively. Then, 3D-LSA module performs self-attention as

\[
PE(x_i,x_j) = FFN(x_i - x_j) \\
q_i^{(l)} = f_i^{(l)} W_q \quad k_i^{(l)} = f_i^{(l)} W_k \quad v_i^{(l)} = f_i^{(l)} W_v \\
y_i^{(l)} = \sum_{j \in N(x_{ct})} \text{softmax}(q_i^{(l)} k_j^{(l)} / \sqrt{d} + PE(x_i,x_j)) \\
f_i^{(l+1)} = f_i^{(l)} + FFN(y_i^{(l)}), \quad (2)
\]

where \( W_q, W_k \), and \( W_v \) are the projection matrices for query, key and value, respectively, \( l \) is the index of L-layer Transformer block and \( d \) is the scaling factor for normalizing dot-product attention. \( FFN(\cdot) \) denotes a position-wise feed forward network, and \( PE(x_i,x_j) \) is the positional encoding function that encodes the difference of two 3D coordinates \( x_i \) and \( x_j \) through FFN. If the number of non-empty voxels in the radius is larger than \( K \), we randomly choose \( K \) voxels. The v-queries associated with a group of \( K \) voxels are separately encoded by self-attention \([31]\). After multiple self-attention layers, the input v-queries \( q_v = \{q_{v,1}, q_{v,2}, \ldots, q_{v,Q}\} \) are updated by the new v-queries \( q_v' = \{q_{v,1}', q_{v,2}', \ldots, q_{v,Q}'\} \).

### D. Dual-Query Cross Attention

The proposed DDA is designed to perform dual-domain feature fusion efficiently. The original transformable attention \([30]\) supports only attention on a single modality, so some modifications are needed to support joint attention on both 2D and 3D modalities.

The structure of DDA is shown in Fig. 3. Let \( q \) index \( Q \) non-empty voxels. Consider a 3D reference point \( p_{3D,q} = (x,y,z) \) at the center point of the \( q \)-th non-empty voxel. The corresponding 2D reference point \( p_{2D,q} = (u,v) \) is determined by projecting \( p_{3D,q} \) on the camera domain and quantizing it on the pixel grid. In multi-view camera setup, \( p_{3D,q} \) can be projected on multiple images. In this case, only a single image is chosen for projection. For consistency, we choose the camera to the far right of the ego vehicle’s direction of travel. The c-query \( q_{c,q} \) is initialized by the camera-domain features \( f_c \) indicated by the reference point \( p_{2D,q} \). The v-query \( q_{v,q}' \) is obtained from 3D local self-attention.

The depth-aware positional encoding is first applied to both v-queries and c-queries. Rather than using \( (u,v) \)-based encoding in the original deformable DETR, the positional embedding is computed based on the depth \( x \) of \( p_{3D,q} \)

\[
PE_{(q,2i)} = \sin(x/(10000^{2i/d_{model}})) \quad (3) \\
PE_{(q,2i+1)} = \cos(x/(10000^{2i/d_{model}})), \quad (4)
\]
The dual-query cross attention decodes the dual-queries $q_v'$ and $q_c'$ over multiple attention layers. First, deformable attention is applied to transform the c-queries $q_c'$ using the camera features $F_c'$ as key and value. For a given 2D reference point $p_{2D,v} = (u, v)$, a deformable mask with adaptive offsets and weights is applied to the camera-domain features $F_c'$. The mask offsets $\Delta p_{mqk}$ and mask weights $A_{mqk}$ are determined as

$$\Delta p_{mqk} = FFN(q_{v,q}) \cdot A_{mqk} = FFN(q_{v,q} + q_{v,q}'),$$

where $+$ denotes the element-wise summation and $FFN$ denotes the feed forward network. Note that the mask weights are determined using both $v$-queries and $c$-queries. This design is intended to boost the effect of feature fusion by determining the attention weights based on voxel-area and camera-area features. Given the offset $\Delta p_{mqk}$ and weight $A_{mqk}$, the attention value $q_v'' = [q_v'', ..., q_v''\cdot q_v'\cdot q_v''\cdot q_v'\cdot q_v''\cdot q_v'\cdot q_v''\cdot q_v'\cdot q_v''\cdot q_v']$ is computed as

$$q_v'' = \sum_{m=1}^{M} W_m \sum_{k=1}^{K} A_{mqk} \cdot W_m' F_c' (p_{2D,v} + \Delta p_{mqk}),$$

where $m$ indexes the attention head, $k$ indexes the sampled keys, and $K$ is the total number of the sampled keys. $W_m \in \mathbb{R}^{C\times C}$ and $W_m' \in \mathbb{R}^{C'/M\times C}$ denote learnable projection matrices.

The queries $q_v'$ and $q_c'$ are further transformed by the gated fusion mechanism [41]. That is, $q_v'$ and $q_c'$ are fused with different ratios given by

$$q_v'' = q_v' + \sigma(Conv_1(q_v' + q_c'))$$

(7)

$$q_c'' = q_c' + \hat{q}_c' + \sigma(Conv_2(q_c' + \hat{q}_c')),$$

(8)

where $\times$ denotes the element-wise multiplication, $\sigma(\cdot)$ is the sigmoid function, and $Conv_1(\cdot)$ and $Conv_2(\cdot)$ are the convolutional layers with different weights. Note that the combining ratios adjust adaptively to the input features since they are a function of $q_v'$ and $q_c'$. Finally, DDA produces the decoded queries $q_v'''$ and $q_c'''$. They are used as input queries for the next Dual-Fusion Transformer layer.

### E. Adaptive Gated Fusion Network

AGFN projects LiDAR features to the camera domain and fuses the projected features with camera features. AGFN also combines two features using the gated fusion mechanism [41]. The AGFN produces the features $F_c'$ through the following operations

$$A = T(F_v) \times \sigma(Conv_v(T(F_v) + F_c))$$

(9)

$$B = F_c \times \sigma(Conv_c(T(F_v) + F_c))$$

(10)

$$F_c' = Conv_r(A \oplus B),$$

(11)

where $T(\cdot)$ denotes the projection on the camera domain and $\oplus$ denotes the concatenation operation.

### IV. EXPERIMENTS

In this section, we evaluate the performance of 3D Dual-Fusion on KITTI dataset [16] and nuScenes dataset [17].

#### A. Experimental setup

1) **KITTI Dataset:** KITTI dataset is a widely used dataset for 3D object detection task. The dataset was collected with a vehicle equipped with a 64-beam Velodyne LiDAR point cloud and a single PointGrey camera. The dataset comprises 7,481 training samples and 7,518 testing samples. The training samples are commonly split into a training set with 3,712 samples and a validation set with 3,769 samples. In the evaluation, we used the **mean average precision** (mAP) as a primary metric. mAP was calculated using recall 40-point precision (R40) and 11-point precision (R11). We used mAP metric $AP_{3D}$ for 3D object detection task and mAP metric $AP_{BEV}$ for BEV object detection task.
TABLE II
PERFORMANCE COMPARISONS OF 3D OBJECT DETECTORS ON nuScenes test set.

| Method                  | Lidar-based | Lidar-Camera based |
|-------------------------|-------------|--------------------|
|                         | mAP  | NDS   | Car   | Truck | C.V.  | Ped.  | Motor | Bicycle | Barrier |
| PointPillars [13]       | 54.11| 51.31| 76.00| 54.02| 11.5  | 64.00| 34.2  | 14.00  | 56.4    |
| CenterPoint [14]        | 60.03| 67.31| 85.52| 53.51| 20.00 | 84.60| 59.5  | 30.70  | 71.1    |
| TransFusion [14]        | 65.50| 70.21| 86.62| 56.71| 28.20 | 86.10| 68.3  | 44.20  | 78.2    |
| 3D-CVF [13]             | 52.70| 62.30| 83.00| 45.00| 15.90 | 74.20| 51.2  | 30.40  | 65.9    |
| PointPainting [15]      | 46.42| 58.11| 77.92| 35.82| 15.82 | 73.30| 41.5  | 24.10  | 60.2    |
| MoCA [15]               | 66.60| 70.92| 86.72| 58.62| 32.62 | 87.10| 67.8  | 52.00  | 72.3    |
| AutoAlign [15]          | 65.80| 70.92| 85.92| 55.32| 29.62 | 86.40| 71.5  | 51.50  | -       |
| FusionPainting [14]     | 68.10| 71.62| 87.12| 60.82| 30.00 | 88.30| 73.7  | 53.50  | 71.8    |
| TransFusion [14]        | 68.90| 71.72| 87.12| 60.02| 33.12 | 88.40| 73.6  | 52.90  | 78.1    |
| Focals Conv [40]        | 67.80| 71.82| 86.52| 57.52| 31.22 | 87.30| 76.4  | 52.50  | 73.0    |
| VVF [28]               | 68.40| 72.42| 86.82| 58.12| 32.12 | 87.10| 78.5  | 52.90  | 73.9    |
| AutoAlign2 [44]         | 68.40| 72.42| 87.00| 59.00| 33.10 | 87.60| 72.9  | 52.10  | 78.0    |
| BEVFusion [45]          | 70.20| 72.92| 88.60| 60.10| 39.30 | 89.20| 74.1  | 51.00  | 80.0    |

3D Dual-Fusion (C)        | 67.90| 71.50| 87.50| 58.80| 31.60 | 86.90| 76.6  | 54.90  | 67.3    |
3D Dual-Fusion (T)        | 70.60| 73.10| 87.70| 62.00| 35.60 | 88.80| 78.0  | 58.00  | 74.4    |

2) nuScenes Dataset: nuScenes dataset is a large-scale benchmark dataset for autonomous driving. The dataset provides the samples from a synced 32-beam LiDAR, six cameras, and five radars covering 360 degrees. It contains 1,000 scenes collected from Boston and Singapore, which consists of 700 training scenes, 150 validation scenes, and 150 testing scenes. Each scene was recorded over 20 seconds at 2Hz and 10 classes of objects were annotated. Models were evaluated mainly using the nuScenes detection score (NDS) [17] and mAP. For the ablation study, 3D Dual-Fusion was trained on 1/7 of the training data and evaluated on the full validation dataset.

3) Implementation Details: For the KITTI dataset, we used the backbone network and detection head of Voxel R-CNN [34] for 3D Dual-Fusion. The range of the point cloud data was within $[0,70.4] \times [-40,40] \times [-3.1]_m$ on $(x,y,z)$ axis. The point clouds were voxelized with each voxel size of $0.05 \times 0.05 \times 0.1m$. The number of Dual-Fusion layers, $M$ was set to 4. We used DeepLabV3 [32] pre-trained on the COCO data for the camera backbone. We used the data augmentation in Voxel R-CNN [34]. In addition, we used the point-image GT sampling augmentation [43].

For the nuScenes dataset, we used two baseline networks: CenterPoint [14] and TransFusion [14]. Note that CenterPoint is LiDAR-only based detector and TransFusion is camera-LiDAR fusion-based detector. We integrated the backbone network and detection head of these methods to 3D Dual-Fusion. The detection head of TransFusion performs proposal-level feature fusion. We applied the detection head of TransFusion to the dense feature maps produced by 3D Dual-Fusion. 3D Dual-Fusion with CenterPoint baseline is called 3D Dual-Fusion (C) and 3D Dual-Fusion with TransFusion baseline is called 3D Dual-Fusion (T). For the ablation study, 3D Dual-Fusion layers $M$ was set to 2. Point-image GT sampling was not enabled in the last four epochs.

B. Main Results

1) KITTI: We evaluate the performance of the proposed 3D Dual-Fusion on KITTI test set. Table I compares the detection accuracy of 3D Dual-Fusion with that of other top ranked 3D object detectors in the leaderboard. We included both LiDAR-only and camera-LiDAR fusion-based methods in our comparison. We did not include the methods using multi-frame sequence data. The performance of other 3D detection methods was obtained from the official KITTI leaderboard. Table I shows that for 3D object detection task, 3D Dual-Fusion achieves the best performance among candidates in all the easy, moderate, and hard categories. For the BEV object detection task, 3D Dual-Fusion achieves the best performance only in the moderate category and competitive performance in other categories. In particular, 3D Dual-Fusion sets a new state-of-the-art performance surpassing the current best method, Focals Conv. Compared to the LiDAR-only based baseline, Voxel R-CNN, the proposed sensor fusion method...


| Method                  | Img Backbone | LiDAR Backbone | mAP   | NDS   |
|-------------------------|--------------|----------------|-------|-------|
| LiDAR-based             |              |                |       |       |
| SECOND [2]              | -            | VoxelNet       | 51.9  | 62.0  |
| CenterPoint [42]        | -            | VoxelNet       | 59.6  | 66.8  |
| FocalsConv [40]         | -            | VoxelNet-FocalsConv | 61.2 | 68.1  |
| TransFusion-L [14]     | -            | VoxelNet       | 65.1  | 70.1  |
| LiDAR-Camera based      |              |                |       |       |
| FUTR3D [46]            | R101         | VoxelNet       | 64.5  | 68.3  |
| FocalsConv [40]         | R50          | VoxelNet-FocalsConv | 65.6 | 70.4  |
| MVP [47]                | DLA34        | VoxelNet       | 67.1  | 70.8  |
| AutoAlignv2 [43]        | CSPNet       | VoxelNet       | 67.1  | 71.2  |
| TransFusion [14]       | R50          | VoxelNet       | 67.5  | 71.3  |
| BEVFusion [45]         | Swin-Tiny    | VoxelNet       | 67.9  | 71.0  |
| BEVFusion [48]         | Swin-Tiny    | VoxelNet       | 68.5  | 71.4  |
| 3D Dual-Fusion (C)     | R50          | VoxelNet       | 67.3  | 71.1  |
| 3D Dual-Fusion (T)     | R50          | VoxelNet       | 69.3  | 72.2  |

**TABLE V**

NDS performance vs. distance on nuScenes 1/7. "3D DF" indicates 3D Dual-Fusion (C).

| Method | <15m | 15-30m | >30m |
|--------|------|--------|------|
| Baseline [42] | 71.7 | 62.1  | 38.4 |
| 3D DF  | 73.8 (+2.1) | 67.6 (+5.5) | 43.9 (+5.5) |

**TABLE VI**

Comparison of different positional encoding methods on nuScenes 1/7. "DDA" indicates dual-query deformable attention.

| Method     | PE method | mAP   | NDS   |
|------------|-----------|-------|-------|
| Baseline [42] | -        | 53.1  | 61.3  |
| DDA        | Image coordinates | 60.6  | 64.6  |
|            | Depth     | 62.1  | 66.2  |

C. Performance Analysis

1) Ablation Study: Table III evaluates the contributions of each module of 3D Dual-Fusion to its overall performance. We provide the evaluation results on both KITTI and nuScenes datasets. CenterPoint [42] is the baseline for the nuScenes dataset and Voxel R-CNN [34] is the baseline for the KITTI dataset. We consider the following components: (a) naive fusion, (b) DDA, (c) 3D local self-attention (3D-LSA), and (d) AGFN. First, we consider a naive fusion method that brings the camera features associated with the non-empty voxels and fuses them with LiDAR voxel features through summation. We see the naive fusion improves mAP in the moderate category by 0.62% on KITTI and NDS by 1.9% on nuScenes. Then, we incorporate the proposed fusion method, DDA. DDA yields 0.47% additional mAP gain in the moderate category on KITTI and 2.0% NDS gain on nuScenes. DDA achieves a considerable performance improvement over the baseline, which demonstrate the effectiveness of the proposed dual-domain fusion method. Next, we add 3D-LSA on top of DDA. 3D-LSA offers 0.29% mAP gain in the moderate category on KITTI and 0.7% NDA gain on nuScenes. The performance gains due to 3D-LSA are relatively small, but they contribute to surpassing other methods. Finally, the addition of AGFN provides an additional gain of 0.3% NDA on nuScenes, which is not negligible. AGFN achieved a 0.34% improvement in the
Dual-Fusion achieves 5.5% gain in NDS for (15 of LiDAR-based CenterPoint baseline [42]. We categorized We compare the performance of 3D Dual-Fusion with that query outperforms the single camera query.

dual queries to aggregate the relevant camera and LiDAR features through multiple attention layers. We designed a novel dual-query cross attention to enable simultaneous dual-domain feature fusion by decoding dual-queries. We also added 3D local self-attention and adaptive gated fusion network for further performance enhancement. Our evaluation conducted on KITTI and nuScenes datasets confirmed that the proposed ideas offered a significant performance boost over the baseline methods and that the proposed 3D Dual-Fusion achieved the state-of-the-art performance in the benchmarks.

V. CONCLUSIONS

In this paper, we proposed a novel camera-LiDAR fusion method for 3D object detection, called 3D Dual-Fusion. We introduced the concept of dual-domain feature fusion, in which the camera and LiDAR features are aggregated in both voxel and camera domains in an interactive manner. To this end, we designed a 3D Dual-Fusion Transformer, which employed dual queries to aggregate the relevant camera and LiDAR features through multiple attention layers. We designed a novel dual-query cross attention to enable simultaneous dual-domain feature fusion by decoding dual-queries. We also added 3D local self-attention and adaptive gated fusion network for further performance enhancement. Our evaluation conducted on KITTI and nuScenes datasets confirmed that the proposed ideas offered a significant performance boost over the baseline methods and that the proposed 3D Dual-Fusion achieved the state-of-the-art performance in the benchmarks.

REFERENCES

[1] 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, 2018, pp. 4490–4499.

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

[3] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, “Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers,” arXiv preprint arXiv:2203.17270, 2022.

[4] V. A. Sindagi, Y. Zhou, and O. Tuzel, “Mxv-net: Multimodal voxelnet for 3D object detection,” in 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 7276–7282.

[5] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, “Pointpainting: Semantic segmentation for multi-view 3D object detection,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 4604–4612.

[6] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, “Ipd: Intensive point-based object detector for point cloud,” arXiv preprint arXiv:1812.05276, 2018.

[7] D. Xu, D. Anguelov, and A. Jain, “Pointfusion: Deep sensor fusion for 3D bounding box estimation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 244–253.

[8] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Fustnet: Pointnet++ for 3D object detection from rgbd data,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 918–927.

[9] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, “Multi-task multi-sensor fusion for 3D object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7345–7353.

[10] M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 641–656.

[11] C. Wang, C. Ma, M. Zhu, and X. Yang, “Pointaugmenting: Cross-modal augmentation for 3D object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 11 794–11 803.

[12] T. Huang, Z. Liu, X. Chen, and X. Bai, “Epnet: Enhancing point features with image semantics for 3D object detection,” in European Conference on Computer Vision. Springer, 2020, pp. 35–52.

[13] J. H. Yoo, Y. Kim, J. Kim, and J. W. Choi, “3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3D object detection,” in European Conference on Computer Vision. Springer, 2020, pp. 720–736.
[14] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai, “Transfusion: Robust lidar-camera fusion for 3d object detection with transformers,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1090–1099.

[15] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, F. Zhao, B. Zhou, and H. Zhao, “Autoalign: Pixel-instance feature aggregation for multi-modal 3D object detection,” arXiv preprint arXiv:2201.06493, 2022.

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “The kitti vision benchmark suite,” URL http://www.cvlibs.net/datasets/kitti, vol. 2, 2015.

[17] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “muscenes: A multimodal dataset for autonomous driving,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11 621–11 631.

[18] 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/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12 697–12 705.

[19] 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.

[20] S. Shi, X. Wang, and H. Li, “Pointrcnn: 3d object proposal generation and detection from point cloud,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 770–779.

[21] Z. Yang, Y. Sun, S. Liu, and J. Jia, “3dssd: Point-based 3d single stage object detector,” in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11 040–11 048.

[22] H. Sheng, S. Cai, Y. Liu, B. Deng, J. Huang, X.-S. Hua, and M.-J. Zhao, “Improving 3d object detection with channel-wise transformer,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2743–2752.

[23] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, “Voxel transformer for 3d object detection,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3164–3173.

[24] X. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, “Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection,” in 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, 2021, pp. 3047–3054.

[25] K. Shin, Y. P. Kwon, and M. Tomizuka, “Roarnet: A robust 3d object detection based on region approximation refinement,” in 2019 IEEE intelligent vehicles symposium (IV). IEEE, 2019, pp. 2510–2515.

[26] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.

[27] J. Ku, M. Moziﬁan, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3d proposal generation and object detection from view aggregation,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1–8.

[28] Y. Li, X. Qi, Y. Chen, L. Wang, Z. Li, J. Sun, and J. Jia, “Voxel field fusion for 3d object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1120–1129.

[29] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le et al., “Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17182–17191.

[30] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, “Deformable detr: Deformable transformers for end-to-end object detection,” arXiv preprint arXiv:2010.04159, 2020.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

[32] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 801–818.

[33] 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/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 10 529–10 538.

[34] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, “Voxel r-cnn: Towards high performance voxel-based 3d object detection,” arXiv preprint arXiv:2012.15712, vol. 1, no. 2, p. 4, 2020.

[35] J. Mao, M. Niu, H. Bai, X. Liang, H. Xu, and C. Xu, “Pyramid r-cnn: Towards better performance and adaptability for 3d object detection,” in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2723–2732.

[36] Y. Zhang, Q. Hu, G. Xu, Y. Ma, J. Wan, and Y. Guo, “Not all points are equal: Learning highly efﬁcient point-based detectors for 3d lidar point clouds,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 18 953–18 962.

[37] J. S. 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, 2022, pp. 8469–8478.

[38] S. Pang, D. Morris, and H. Radha, “Clods: Camera-lidar object candidates fusion for 3d object detection,” in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 10 386–10 393.

[39] X. Li, B. Shi, Y. Hou, X. Wu, T. Ma, Y. Li, and L. He, “Homogeneous multi-modal feature fusion and interaction for 3d object detection,” in European Conference on Computer Vision. Springer, 2022, pp. 691–707.

[40] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, “Focal sparse convolutional networks for 3d object detection,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 5428–5437.

[41] J. Kim, J. Koh, Y. Kim, J. Choi, Y. Hwang, and J. W. Choi, “Robust deep multi-modal learning based on gated information fusion network,” in Asian Conference on Computer Vision. Springer, 2018, pp. 90–106.

[42] 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, 2021, pp. 11 784–11 793.

[43] W. Zhang, Z. Wang, and C. C. Loy, “Exploring data augmentation for multi-modality 3d object detection,” arXiv preprint arXiv:2012.12741, 2020.

[44] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, “Autoalignv2: Deformable feature aggregation for dynamic multi-modal 3d object detection,” arXiv preprint arXiv:2207.10316, 2022.

[45] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, “BEVFusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation,” arXiv preprint arXiv:2205.13542, 2022.

[46] X. Chen, T. Zhang, Y. Wang, and H. Zhao, “FUTR3D: A unified sensor fusion framework for 3d detection,” arXiv preprint arXiv:2203.10642, 2022.

[47] T. Yin, X. Zhou, and P. Krahenbuhl, “Multimodal virtual point 3d detection,” Advances in Neural Information Processing Systems, vol. 34, pp. 16 494–16 507, 2021.

[48] 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,” arXiv preprint arXiv:2205.13790, 2022.