DeepInteraction: Exploring Multi-modal Interaction for 3D Object Detection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Existing multi-modal 3D object detectors typically consider a unilateral association strategy with a biased inclination on 3D LiDAR point clouds whilst treating the 2D multi-camera images as an auxiliary information source. As a result, those useful high-resolution information unique with the images is rigidly thrown away in modality association. In essence, the intrinsic complementary nature between the two modalities is fully overlooked by prior arts. In this work, we introduce a novel 3D object detection architecture, dubbed as `DeepInteraction`, characterized by bilateral interaction and association throughout both representation encoding and decoding, in order to maximally exploit the inter-modal complementary property. Extensive experiments verify the accuracy superiority of `DeepInteraction` over the state-of-the-art methods by large margin on the large scale nuScenes benchmark.

1 Introduction

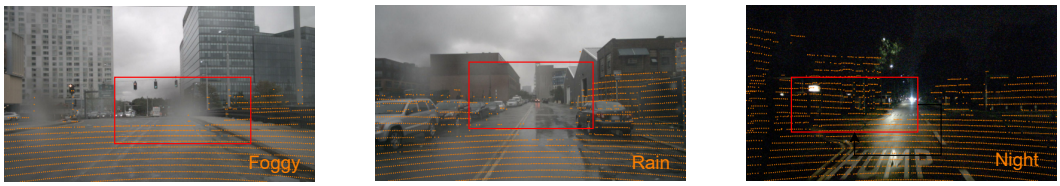


Figure 1: Visualization of **the complementary property** between the coupled LiDAR 3D point clouds (orange colored) and multi-camera 2D images. As highlighted in red boxes, often one modality (either the point clouds or image) is favored over the other depending on the specific situations. Neither modality is always preferred, which means the two modalities are strongly complementary.

3D object detection is critical for autonomous driving by localizing and recognizing decision-sensitive objects in the 3D space. For reliable object detection, LiDAR and camera sensors have been systematically deployed to provide low-resolution point clouds with depth information and high-resolution visual images, respectively. The two modalities exhibit naturally strong complementary effects due to such different perception characteristics. However, it is non-trivial to associate and jointly exploit both modalities due to the intrinsic sensor misalignment challenge [38]. As highlighted in the red boxes of Figure 1, one modality would be favored over the other depending on which unconstrained driving conditions. Neither modality is always preferred throughout, which means the two modalities are strongly complementary. As a specific example, due to the low-resolution nature with LiDAR sensor, small objects will be significantly under-represented with only one point

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

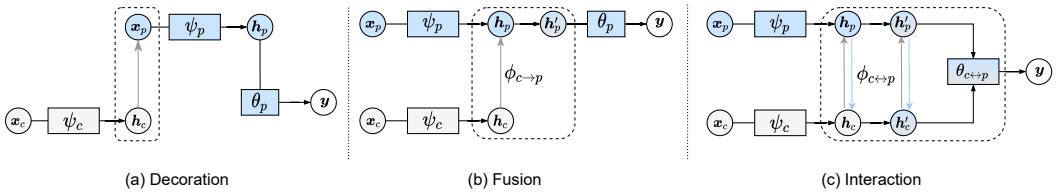


Figure 2: Comparing multi-modal association strategies for 3D object detection. (a) The 2D backbone ϕ_c extracts features $\boldsymbol{h}_c$ of the image $\boldsymbol{x}_c$ to decorate the LiDAR points $\boldsymbol{x}_p$ at the input-level of 3D backbone ψ_p. (b) The association is conducted at the feature-level by a fusion module $\phi_{c\rightarrow p}$ with the image $\boldsymbol{x}_c$ as the auxiliary information source. Both (a) and (b) are unilateral. (c) We propose a bilateral interaction paradigm $\phi_{c\leftrightarrow p}$ realized by the multi-modal module $\theta_{c\leftrightarrow p}$.

or very few points; In this case, the coupled image will be useful by offering more visual appearance information thanks to its high-resolution property.

Existing top-performing multi-modal 3D object detection methods fall into two categories. The *first* category of methods (*e.g.*, PointPainting [29] and PointAugmentation [30]) decorate the LiDAR points at the input level (Figure 2(a)). Specifically, the image features are used to enrich the low-resolution point cloud information (*e.g.*, shape and depth). Such decorated point clouds are limited in improving the performance due to the big inter-modality discrepancy. To address this issue, the *second* category of methods (*e.g.*, TransFusion [1]) take a feature-level fusion paradigm (Figure 2(b)). Technically, this strategy maximizes the correlation of LiDAR and camera data in the representation space. Commonly, both paradigms treat the coupled images as an ***auxiliary*** modality with a biased inclination on the LiDAR modality. As a consequence, those useful information unique with the 2D image data is simply thrown away during the *unilateral* fusion step. That being said, the intrinsic complementary property of the two modalities as clearly exhibited in the raw data is not properly and fully leveraged.

To overcome the aforementioned architectural limitations of existing solutions, in this work we introduce a novel multi-modal 3D object detection architecture capable of conducting *bilateral* interaction between 3D point clouds and multi-camera visual images (Figure 1), dubbed as `DeepInteraction`. Specifically, the whole pipeline of `DeepInteraction` has three stages: (1) In the feature extraction stage, 3D point clouds and 2D multi-view images are encoded into intermediate representations using two modality-specific encoders, respectively; (2) In the encoding stage, the two representations intensively are interacted and fused with each other with a multi-modal encoder to build more expressive embedding; (3) In the decoding stage, these embeddings are further alternatively aggregated with a multi-modal dynamic decoder for generating predictions. In design, the two modalities are associated and interacted in both directions so that modality-specific strengths can be fully leveraged for maximally capitalizing their complementary effects.

The **contributions** of this work are summarized as follows: **(i)** With in-depth assessment into the model design, we reveal an architectural limitation of existing multi-modal 3D object detection methods that multi-camera 2D images with fine visual appearance information is only exploited unilaterally as an auxiliary modality. This ends up with *insufficient exploitation* of the multi-modality perception data coming with intrinsically strong complementary nature. **(ii)** To solve this limitation, we introduce a novel `DeepInteraction` architecture based on bilateral interaction and association across modalities, through both representation encoding and decoding in both directions. **(iii)** Extensive experiments on the nuScenes dataset show that our `DeepInteraction` yields new state of the art for multi-modality 3D object detection, often with a large margin over the alternatives.

2 Related work

3D object detection with single modality Automated driving vehicles are generally equipped with both LiDAR and multiple surround-view cameras. Many previous methods preform 3D object

detection by exploiting data captured from only a single form of sensors. For camera-based 3D object detection, since depth information is not directly accessible from RGB images, some previous works [17, 31, 27] lift 2D features into a 3D space by conducting depth estimation, followed by performing object detection in the 3D space. Another line of works [32, 23, 21] resort to the detection Transformer [5] architecture. They leverage 3D object queries and 3D-2D correspondence to incorporate 3D computation into the detection pipelines. Despite the rapid progress of camera-based approaches, the state-of-the-art of 3D object detection is still dominated by LiDAR-based methods. Most of LiDAR-based detectors quantify point clouds into regular grid structures such as voxels [39, 35], pillars [20] or range images [2, 15, 7] before processing them. Due to the sampling characteristics of LiDAR, these grids are naturally sparse and hence fit the Transformer design. So, a number of approaches [24, 14] have applied the Transformer for point cloud feature extraction. Differently, several methods use the Transformer decoder or its variants as their detection head [1, 33]. In contrast, we present an encoder-decoder architecture as an unified framework for both feature interaction and prediction.

Multi-modality fusion for 3D object detection Leveraging the perception data from both camera and LiDAR sensors typically leads to improved performance, which hence emerges as a promising direction. Existing 3D detection methods typically perform multi-modal association/fusion at one of the three points: raw input, intermediate feature, and object proposal. For example, PointPainting [29] is the pioneering input fusion method [30, 18, 12]. The main idea is to decorate the 3D point clouds with the category scores or semantic features from the 2D instance segmentation network. Whilst 4D-Net [25] placed the fusion module in the point cloud feature extractor for allowing the point cloud features to dynamically attend to the image features. The proposal based fusion methods [19, 8] keep the feature extraction of two modalities independently and aggregate multi-modal features via proposals or queries at the detection head. The first two categories of methods take a unilateral fusion strategy with bias to 3D LiDAR modality due to the superiority of point clouds in distance and spatial perception. Instead, the last category fully ignores the intrinsic association between the two modalities in representation. As a result, all these previous methods fail to fully exploit both modalities with strong complementary nature. In this work, we address this limitation with all previous solutions by developing a bilateral interaction and association based architecture for exploring the inter-modal complementary effect.

3 DeepInteraction

We adopt DETR [6] as the backbone architecture due to no need of hand-crafted components like anchor generation and non-maximum suppression. At the presence of multi-modal inputs, extending the standard DETR, originally designed for single-modal 2D object detection, becomes non-trivial. This is primarily due to an extra challenge of reliably associating the 3D LiDAR points and 2D image pixels without perfect correspondence. We hence introduce a novel bilateral association framework (`DeepInteraction`) for multi-modal 3D object detection, characterized by deep interaction between 3D point clouds and 2D multi-camera images. In contrast to all prior arts, we treat the 3D LiDAR and 2D camera modalities *equally* and conduct multi-modal interaction through both representation encoding and decoding. An overview of our `DeepInteraction` is shown in Figure 3(a).

In the following, we will first provide the preliminary of 3D object detection with Transformers in Section 3.1. We then describe how to perform multi-modal interaction in both the encoder (Section 3.2) and decoder (Section 3.3).

3.1 Preliminary: 3D object detection with Transformers

Given multi-modal input including 3D LiDAR point clouds $\boldsymbol{x}_p$ and surrounding 2D RGB images $\boldsymbol{x}_c$, the model need to predict a set of 3D bounding boxes along with category labels $\hat{\boldsymbol{y}}$. We start with the context representation extraction per modality: $\boldsymbol{h}_p = f_{\psi_p}(\boldsymbol{x}_p)$ for 3D point cloud by a 3D backbone and $\boldsymbol{h}_c = f_{\psi_c}(\boldsymbol{x}_c)$ for 2D image by a 2D backbone. Adopting the original structure

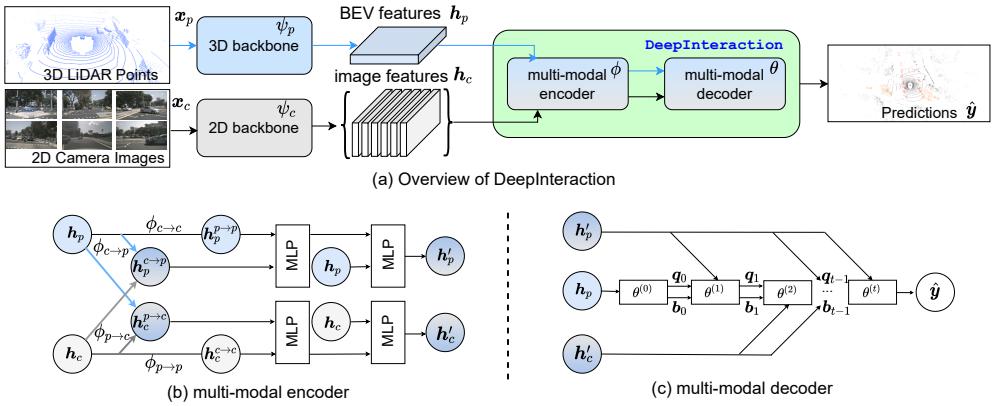


Figure 3: Overview of `DeepInteraction` for LiDAR-camera sensors based 3D object detection.

of the Transformer, our model consists of two modules: (i) the multi-modal encoder, and (ii) the multi-modal decoder. Taking the context representations $\boldsymbol{h}_p$ and $\boldsymbol{h}_c$ as input, the encoder outputs an enhanced multi-modal embeddings:

$$\boldsymbol{h} = f_\phi(\mathcal{A}(\boldsymbol{h}_p, \boldsymbol{h}_c)), \tag{1}$$

where $\mathcal{A}(\cdot)$ is an association function that combines the heterogeneous inputs from two modalities. This $\mathcal{A}(\cdot)$ can be instantiated as addition, concatenation, or a learnable module such as the multi-head cross-attention as in Transfusion [1]. The decoder needs to make the final predictions by taking the multi-modal representations $\boldsymbol{h}$ as input:

$$\hat{\boldsymbol{y}} = f_\theta(\boldsymbol{h}). \tag{2}$$

Following DETR, the optimal bipartite matching is computed for matching the set of predictions $\hat{\boldsymbol{y}}$ with the ground-truth $\boldsymbol{y}$ annotations. During optimization, the model is trained to minimize the matching cost.

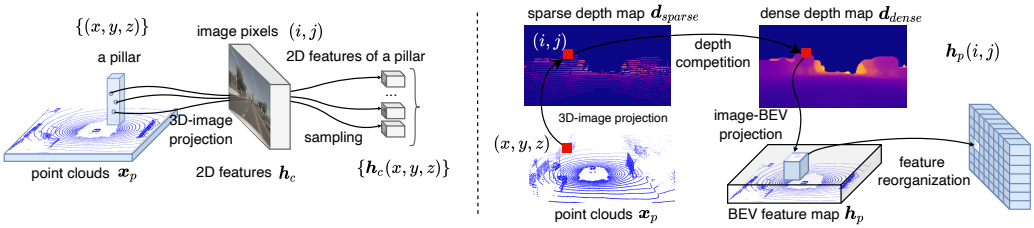
Remarks As described above, the multi-modal association in Transfusion [1] by design is unilateral, conducted before the encoder input (Eq. 1). Thus, the LiDAR-camera data is not fully interacted and exploited for final prediction (Eq. 2). Instead of employing a stand-alone and unilateral association, we formulate $\mathcal{A}(\cdot)$ within both encoder and decoder, so that the two modalities can interact deeply for maximally capitalizing their complementary nature.

3.2 Multi-modal interaction in encoding

Multi-modal bilateral mapping To enable multi-modal interaction in the encoder, we first need to build the correspondence relationship between the context representations $\boldsymbol{h}_p$ and $\boldsymbol{h}_c$. To that end, we build a bilateral mapping between the 3D LiDAR point cloud and 2D camera images: $\mathcal{M}_{p\to c}$ and $\mathcal{M}_{c\to p}$, to locate and sample the spatially corresponding features between the two modalities.

For the 3D LiDAR to 2D camera mapping $\mathcal{M}_{p\to c}$ (Figure 4(a)), we first obtain the point coordinates $\{(x, y, z)\}$ of a pillar. Then, we project the 3D locations into a set of pixel coordinates $\{(i, j)\}$ on the corresponding image views, according to the extrinsic and intrinsic camera matrices. Also, the bilinear interpolation is used to sample 2D image features $\{\boldsymbol{h}_c(x, y, z)\}$ located at $\{(i, j)\}$ for the pillar from multi-view image context representations $\boldsymbol{h}_c$.

For the 2D camera to 3D LiDAR mapping $\mathcal{M}_{c\to p}$ (Figure 4(b)), we project each point (x, y, z) on 3D point cloud to the spatially corresponding image location (i, j) to obtain the sparse depth maps $\boldsymbol{d}_{sparse}$. Then, we complete sparse depth maps $\boldsymbol{d}_{sparse}$ into dense depth map $\boldsymbol{d}_{dense}$ and project the depth to 3D coordinates (x, y, z), which is used to locate a BEV feature pillar $\boldsymbol{h}_p(i, j)$ at the location of (x, y, z) from $\boldsymbol{h}_p$ for the 2D image location (i, j). All located BEV feature pillars are reorganized into a 2D shape.



(a) feature mapping from 3D to 2D

(b) feature mapping from 2D to 3D

Figure 4: The bilateral mappings between the 3D LiDAR and 2D camera representations.

Multi-modal feature interactions We aim to construct multilateral interaction, enabling information to flow in different modality. We introduce inner-modal interaction (*i.e.*, $f_{\phi_{p\to p}}$ and $f_{\phi_{c\to c}}$) to explore the intrinsic connections of the inner-modal modalities themselves. Besides, to associate the spatial and semantic information on the same object, we also design the cross-modal interaction (*i.e.*, $f_{\phi_{c\to p}}$ represents the interaction from image to point clouds and $f_{\phi_{p\to c}}$ vice versa). In the following, we detail the four multi-modal feature interactions (Figure 3(b)).

For all types of interaction module, we use local attention [26] to capture the local correlation. This can be formulated analogously as:

$$\boldsymbol{v}' = \sum_{\boldsymbol{k},\boldsymbol{v}\in\mathcal{N}_q} \text{softmax}\left(\frac{\boldsymbol{q}\boldsymbol{k}}{\sqrt{d_k}}\right)\boldsymbol{v}, \tag{3}$$

where $\boldsymbol{q}$ is the query feature and $\mathcal{N}_q$ denotes a group of features in the spatial neighborhood of $\boldsymbol{q}$.

The inner-modal interaction of point clouds ($\boldsymbol{h}_p^{p\to p} = f_{\phi_{p\to p}}(\boldsymbol{h}_p)$) and images ($\boldsymbol{h}_c^{c\to c} = f_{\phi_{c\to c}}(\boldsymbol{h}_c)$) allow interaction within the neighboring feature pixels using Eq. 3. In this case, we define $\mathcal{N}_q$ as the inner-modal features in 2D spatial $k \times k$ neighborhood of query feature pixel, where $k = 9$. We transform each pixel of input feature into $\boldsymbol{q}$, and transform $\mathcal{N}_q$ into $\boldsymbol{k}$ and $\boldsymbol{v}$. In doing so, the final features that are weighted by the attention map (*i.e.*, $\boldsymbol{h}_p^{p\to p}$ and $\boldsymbol{h}_c^{c\to c}$) are more likely to focus on locally important information.

Cross-modal interactions take the same spirit of locality, with a slight modification on the definition of $\mathcal{N}_q$ in Eq. 3.

For the interaction from point clouds to images ($\boldsymbol{h}_c^{p\to c} = f_{\phi_{p\to c}}(\boldsymbol{h}_p, \boldsymbol{h}_c)$), the LiDAR BEV feature map $\boldsymbol{h}_p$ is first reorganized into the same form with multi-view image feature maps $\boldsymbol{h}_c$ via 3D LiDAR to 2D camera mapping $\mathcal{M}_{p\to c}$. Then, for each query $\boldsymbol{q} = \boldsymbol{h}_c[i, j]$ in Eq. 3, the neighborhood $\mathcal{N}_q$ is a set of feature pixels within a square patch centered around (i, j) on reorganized LiDAR BEV feature map, with spatial extent $k = 9$.

On the contrary, the interaction from images to point clouds ($\boldsymbol{h}_p^{c\to p} = f_{\phi_{c\to p}}(\boldsymbol{h}_c, \boldsymbol{h}_p)$) is proposed to spread visual signals from multi-view image feature maps $\boldsymbol{h}_c$ to LiDAR BEV feature map $\boldsymbol{h}_p$. Specifically, given $\boldsymbol{q} = \boldsymbol{h}_p[i, j]$ as query, its neighborhood $\mathcal{N}_q$ is defined as a group of image features sampled by $\mathcal{M}_{c\to p}$.

In the above processes, four interacted multi-modal feature maps (*i.e.*, $\boldsymbol{h}_p^{p\to p}$, $\boldsymbol{h}_c^{c\to c}$, $\boldsymbol{h}_p^{c\to p}$, and $\boldsymbol{h}_c^{p\to c}$) are generated by multi-modal feature interactions described above respectively. Then, a pair of MLP layers (denoted as MLP$(\cdot)$ in Eq. 4,5) are conducted in succession to fuse the augmented features before and after interactions into the final enriched representation (Figure 3(b)). The computation of each encoder layer can be formulated as:

$$\boldsymbol{h}_p' = \text{MLP}(\text{MLP}(\boldsymbol{h}_p^{p\to p},\ \boldsymbol{h}_p^{c\to p}),\ \boldsymbol{h}_p), \tag{4}$$

$$\boldsymbol{h}_c' = \text{MLP}(\text{MLP}(\boldsymbol{h}_c^{c\to c},\ \boldsymbol{h}_c^{p\to c}),\ \boldsymbol{h}_c). \tag{5}$$

Furthermore, we can construct more expressive representations by stacking more encoder layers, where each subsequent layer takes the multi-modal enriched representations $\boldsymbol{h}_p'$ and $\boldsymbol{h}_c'$ from the previous layer as input.

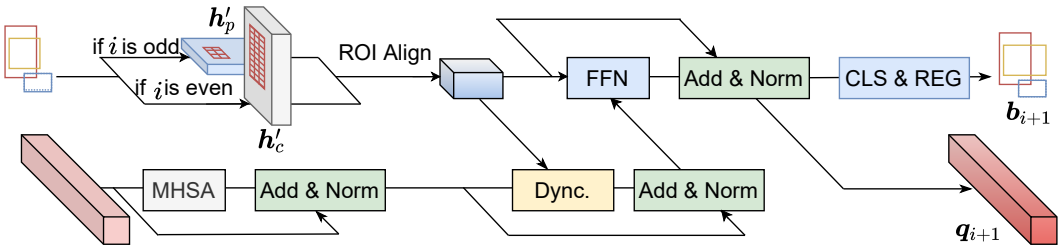


Figure 5: The illustration of the multi-modal decoder interaction layer.

3.3 Multi-modal interaction in decoding

After obtaining the interacted representations $\boldsymbol{h}'_p$ and $\boldsymbol{h}'_c$ for LiDAR and camera, we further consider to maximize the complementary effect in the decoder. As single-modal representations alone may not cover enough useful information for making robust predictions. Therefore, we propose a multi-modal alternate query refinement strategy to perform multi-modal interaction in the decoder.

Multi-modal alternate query refinement As illustrated in Figure 3(c), the proposed multi-modal decoder contains N alternate decoder layers which iteratively refine the bounding box predictions by aggregating features from two modalities. Specifically, we follow Transfusion [1] to obtain the initial queries $\boldsymbol{q}$ of decoder from combining LiDAR interacted features $\boldsymbol{h}'_p$ and original LiDAR context features $\boldsymbol{h}_p$ to speed up convergence. To enable more flexible interaction between modalities, we employ the dynamic instance interaction (Dync.) head [28] as the decoder interaction layer, as shown in Figure 5, in which we compute the predictions by alternately taking the interacted representations $\boldsymbol{h}'_p$ and $\boldsymbol{h}'_c$ of two modalities as input. In doing so, the decoder queries $\boldsymbol{q}$ are refined iteratively by fully exploiting both LiDAR and camera information.

3.4 Optimization

During training, we utilize a set-to-set loss [6] between predicted and ground-truth objects:

$$\mathcal{L}_{\text{hungarian}}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \sum_{i=1}^{m} \left(\mathcal{L}_{\text{cls}}\left(\boldsymbol{c}_i, \hat{\boldsymbol{c}}_{\sigma(i)}\right) + \mathbb{1}_{\{\boldsymbol{c}_i \neq \emptyset\}} \mathcal{L}_{\text{reg}}\left(\boldsymbol{b}_i, \hat{\boldsymbol{b}}_{\sigma(i)}\right) \right), \tag{6}$$

where $\sigma(\cdot)$ denotes the optimal bipartite matching between y and $\hat{y}$ found by Hungarian algorithm, $\mathcal{L}_{\text{cls}}$ is the focal loss for classification, and $\mathcal{L}_{\text{reg}}$ is the ℓ_1 loss for bounding box regression. The matching cost $\mathcal{L}_{\text{match}}$ used for computing optimal assignment $\sigma(\cdot)$ is the weighted sum of $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{reg}}$ and the negative 3D IoU. We also compute the auxiliary loss with the same form as Eq. 6 for the predictions of each layer to stabilize convergence. In addition, we provide direct supervision for heatmap prediction in query initialization by adding an extra Guassian Focal Loss [36].

4 Experiments

4.1 Experimental setup

Dataset We evaluate our approach on the nuScenes dataset [3]. It provides 32-beam point clouds from LiDAR and images with resolution of 1600×900 from 6 surrounding cameras. The total of 1000 scenes, where each sequence is roughly 20 seconds long and annotated every 0.5 second, is officially split into `train/val/test` set with 700/150/150 scenes. For the 3D object detection task, 1.4M objects in scenes are annotated with 3D bounding boxes and classified into 10 categories: car, truck, bus, trailer, construction vehicle, pedestrian, motorcycle, bicycle, barrier, and traffic cone. We follow the common practice to transform points from the past 9 sweeps to the current frame.

Metric Mean average precision (mAP) [13] and nuScenes detection score (NDS) [3] are used as the evaluation metric of 3D detection performance. The final mAP is computed by averaging over

Table 1: Comparison with state-of-the-art methods on the nuScenes `test` set. 'C.V.', 'Ped.', and 'T.C.','M.T.' and 'T.L.' are short for construction vehicle, pedestrian, traffic cone, motor and trailer respectively. '`L`' and '`C`' represent LiDAR and camera, respectively. For FusionPainting [34], we report the results on the nuScenes website, which are better than what they reported in their paper. † denotes test-time augmentation is used, § denotes that both test-time augmentation and model ensemble are applied for testing.

Method	Mod.	mAP	NDS	Car	Truck	C.V.	Bus	T.L.	B.R.	M.T.	Bike	Ped.	T.C.
CenterPoint [36]†	L	60.3	67.3	85.2	53.5	20.0	63.6	56.0	71.1	59.5	30.7	84.6	78.4
Focals Conv [9]	L	63.8	70.0	86.7	56.3	23.8	67.7	59.5	74.1	64.5	36.3	87.5	81.4
TransFusion-L [1]	L	65.5	70.2	86.2	56.7	28.2	66.3	58.8	78.2	68.3	44.2	86.1	82.0
PointAug. [30]†	L+C	66.8	71.0	87.5	57.3	28.0	65.2	60.7	72.6	74.3	50.9	87.9	83.6
MVP [37]	L+C	66.4	70.5	86.8	58.5	26.1	67.4	57.3	74.8	70.0	49.3	89.1	85.0
FusionPainting [34]	L+C	68.1	71.6	87.1	60.8	30.0	68.5	61.7	71.8	74.7	53.5	88.3	85.0
AutoAlign [10]	L+C	68.4	72.4	87.0	59.0	33.1	69.3	59.3	78.0	72.9	52.1	87.6	85.1
TransFusion [1]	L+C	68.9	71.7	87.1	60.0	33.1	68.3	60.8	78.1	73.6	52.9	88.4	86.7
Focals Conv-F [9]†	L+C	70.1	73.6	87.5	60.0	32.6	69.9	**64.0**	71.8	81.1	59.2	89.0	85.5
DeepInteraction	L+C	**70.8**	73.4	**87.9**	60.2	**37.5**	**70.8**	63.8	**80.4**	75.4	54.5	**91.7**	**87.2**
DeepInteraction†	L+C	**72.5**	**74.4**	**88.4**	**61.5**	**38.9**	**72.4**	63.6	**81.7**	80.3	59.2	**90.3**	**87.0**
DeepInteraction§	L+C	**73.5**	**74.8**	**88.8**	**63.0**	**41.7**	**73.0**	62.1	**82.7**	**81.4**	**62.4**	**91.8**	**88.1**

distance thresholds of 0.5m, 1m, 2m, 4m across 10 classes. NDS is a weighted average of mAP and other attributes metrics, including translation, scale, orientation, velocity, and other box attributes.

4.2 Implementation details

Our implementation is based on the public code base *mmdetection3d* [11]. For image branch, we initialize our backbone and FPN [22] from the instance segmentation model Cascade Mask R-CNN [4] with resnet-50 [16] pretrained on nuImage [3]. To save the computation cost, we rescale the input image to 1/2 of its original size before feeding into the network, and freeze the weights of image branch during training. The voxel size is set to $(0.075m, 0.075m, 0.2m)$, and the detection range is set to $[-54m, 54m]$ for for X and Y axis and $[-5m, 3m]$ for Z axis.

For data augmentation, we follow TransFusion [1] and adopt random rotation with a range of $r \in [-\pi/4, \pi/4]$, random scaling with a factor of $r \in [0.9, 1.1]$, random translation with standard deviation 0.5 in three axis, and random horizontal flipping. We also use the class-balanced resamping in CBGS [40] to balance the class distribution for nuScenes. We use TransFusion-L [1] as our LiDAR-only baseline. For training schedule, we use Adam optimizer with one-cycle learning rate policy, with max learning rate 1×10^{-3}, weight decay 0.01 and momentum 0.85 to 0.95, following CBGS [40]. Our LiDAR-only baseline is trained for 20 epochs and LiDAR-camera fusion for 6 epochs with batch size of 16 using 8 NVIDIA V100 GPUs.

4.3 Comparison to the state of the art

We compare with state-of-the-art alternatives on the nuScenes test set. We test single model with and without test-time augmentation respectively, as well as model ensemble. Following the common practice [36] for model ensemble we merge the results from multiple models with voxel size of 0.05m, 0.075m, 0.1m, 0.125m and 0.15m. As shown in Table 1, our model achieves 70.8% mAP and 73.4% NDS on the nuScenes test set without bells and whistles, outperforming alternative Transfusion [1] by 1.9% mAP and 1.7% NDS. Our model beats the closest rival Focals Conv-F [9] by a considerable margin under the same configuration, which clearly verifies the effectiveness of our multi-modal interaction approach. To the best of our knowledge, `DeepInteraction` surpasses all the published LiDAR-camera methods on the nuScenes 3D object detection test set leaderboard.

Table 2: Ablation on bilateral interaction encoder (denoted as 'Encoder') and multi-modal interaction decoder (denoted as 'Decoder') on nuScenes `val` set. We use the TransFusion-L [1] as baseline.

Encoder	Decoder	Mod.	mAP	NDS	mATE	mASE	mAOE	mAVE	mAAE
		L	65.1	70.2	0.275	**0.249**	**0.274**	**0.247**	0.189
	✓	L+C	66.4	70.7	0.277	0.254	0.279	0.253	**0.187**
✓	✓	L+C	**69.9**	**72.7**	**0.267**	0.250	0.276	0.248	0.189

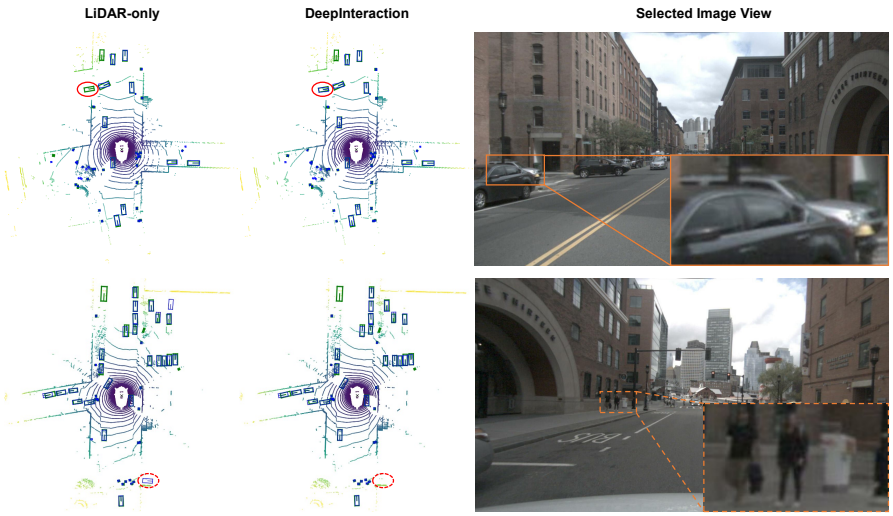


Figure 6: Qualitative comparison between LiDAR-only baseline and our DeepInteraction. Blue boxes and green boxes are the predictions and ground-truth, respectively. Solid eclipse indicates false negative, and dashed eclipse indicates false positive.

4.4 Ablation studies

3D Backbone As shown in Table 3e, we examine our framework's compatibility with different 3D backbones: PointPillars [20] and VoxelNet [39]. For PointPillars backbone, we set the voxel size to (0.2m, 0.2m) while keeping all the other settings the same as other experiments. To be fair, we have kept the same number of queries as TransFusion [1]. Benefiting from deep feature interactions, our model exhibits consistent improvements over LiDAR-only baseline methods (see Figure 6) on both backbones (by 5.5% mAP for voxel-based backbone, and 4.4% mAP for pillar-based backbone). This result demonstrates the proposed `DeepInteraction` is generic to different point cloud encoding manners.

Where does the improvement come from? To investigate exactly where these improvements come from, we conduct extensive experiments on the nuScenes `val` dataset to test the effectiveness of each component. From Table 2, we can see that the alternative interaction in the decoder brings a gain of 0.7% mAP and 0.5% NDS compared to the lidar-only baseline [1]. Further gains of 3.5% mAP and 2.7% NDS are achieved by the bidirectional multi-modal interactions before detection. This suggests that both our bilateral interaction encoder and multi-modal interaction decoder are helpful, and that interaction in the early stage (encoder) appears more critical than those in the decoder.

Bilateral interaction For an in-depth understanding of bilateral interaction encoder, we present a performance comparison about inner-modal interaction (denoted as 'Inner-Int.') and cross-modal interaction (denoted as 'Cross-Int.') in Table 3a. With only inner-modal interaction, there is a performance improvement of 1.7% mAP and 1.2% NDS. When only cross-modal interaction is introduced, the performance improves significantly by 3.1% mAP and 1.8% NDS. With both inner-modal and cross-modal interactions are applied, our method achieves state-of-the-art performance with 69.9% mAP and 72.7% NDS.

Table 3: Ablation studies on the encoder (a) (c), decoder (b) (d), detection backbone (e) and number of queries (f) of our model. 'L' and 'C' represent LiDAR and camera respectively.

Inner-int.	Cross-int.	mAP	NDS
		66.4	70.7
✓		68.1	71.9
	✓	69.5	72.5
✓	✓	69.9	72.7

(a) Ablation study on the bilateral interaction encoder.

Int. of LiDAR	Int. of Camera	mAP	NDS
Transformer	Transformer	68.6	71.6
Transformer	Dync.	69.3	72.1
Dync.	Dync.	69.9	72.6

(b) Ablation study on the multi-modal interaction decoder.

number of encoder layers	mAP	NDS
w/o	66.4	70.7
1	67.7	71.2
2	69.9	72.6
3	69.6	72.3

(c) Ablation study on the number of bilateral interaction encoder layers.

number of head layers	mAP	NDS	FPS
2	69.5	72.3	1.53
3	69.7	72.5	1.46
4	69.8	72.5	1.45
5	69.9	72.6	1.42
6	69.7	72.1	1.38

(d) Ablation study on the number of decoder layers.

Methods	Modility	mAP	NDS
PointPillars [20]	L	46.2	59.1
+ CenterPoint [36]	L	50.3	60.2
+ Transfusion-L [1]	L	54.5	62.7
+ Transfusion [1]	L+C	58.3	64.5
+ DeepInteraction	L+C	60.0	65.6
VoxelNet [39]	L	52.6	63.0
+ CenterPoint [36]	L	59.6	66.8
+ Transfusion-L [1]	L	65.1	70.1
+ Transfusion [1]	L+C	67.5	71.3
+ DeepInteraction	L+C	69.9	72.6

(e) Ablation study on the 3D detection backbone.

train	inference	mAP	NDS
	200	69.9	72.6
200	300	70.1	72.7
	400	70.0	72.6
	200	69.7	72.5
300	300	69.9	72.6
	400	70.0	72.6

(f) Ablation study on the number of queries for training and inference. We use 200 and 300 queries in training and inference respectively as the default setting in our model.

Alternative interaction We study the impact of multi-modal interaction in the decoder. As shown in Table 3d, increasing the number of decoder layers can consistently improve the performance except that 6 layers experience a little performance drop. Table 3b shows the comparison between different interaction candidates, including Transformer decoder layer [1] and dynamic instance interaction. The second row of Table 3b presents the mixing strategy, *i.e.*, using Transformer decoder layer for augmented LiDAR BEV features and Dync. for the augmented image features, which leads to a boost of 0.7% mAP compared with the pure Transformer counterparts. The best result comes from using Dync. for both modalities. This suggests that our multi-modal decoder with dynamic instance interaction is capable of exploiting the local context for specific object query. Since object queries are non-parametric and input-dependent [1], we can modify the number of queries during inference. In Table 3f, we report the results under different combinations of query numbers in the training and testing. As seen, the best performance is achieved when training the model with 200 queries and testing with 300 queries.

5 Conclusion

In this work, we have presented a novel 3D object detection architecture `DeepInteraction` for exploring the intrinsic inter-modal complementary nature. This key idea is to conduct bilateral interaction and association across modalities in representation learning. This strategy is designed particularly to resolve the fundamental limitation of existing alternatives that 2D multi-camera images are insufficiently exploited due to their auxiliary source role treatment. Extensive experiments on the nuScenes benchmark demonstrate that `DeepInteraction` surpasses the state-of-the-art methods by a large margin, yielding new state of the art.

References

[1] Xuyang Bai, Zeyu Hu, Xinge Zhu, Qingqiu Huang, Yilun Chen, Hongbo Fu, and Chiew-Lan Tai. Transfusion: Robust lidar-camera fusion for 3d object detection with transformers. In *CVPR*, 2022.

[2] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv preprint*, 2020.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

[4] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: high quality object detection and instance segmentation. *IEEE TPAMI*, 2019.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[7] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Drago Anguelov. To the point: Efficient 3d object detection in the range image with graph convolution kernels. In *CVPR*, 2021.

[8] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *arXiv preprint*, 2022.

[9] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. *arXiv preprint*, 2022.

[10] Zehui Chen, Zhenyu Li, Shiquan Zhang, Liangji Fang, Qinghong Jiang, Feng Zhao, Bolei Zhou, and Hang Zhao. Autoalign: Pixel-instance feature aggregation for multi-modal 3d object detection. *arXiv preprint*, 2022.

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

[12] Zhuangzhuang Ding, Yihan Hu, Runzhou Ge, Li Huang, Sijia Chen, Yu Wang, and Jie Liao. 1st place solution for waymo open dataset challenge–3d detection and domain adaptation. *arXiv preprint*, 2020.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[14] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. *arXiv preprint*, 2021.

[15] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and ZhaoXiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *ICCV*, 2021.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint*, 2021.

[18] Tengteng Huang, Zhe Liu, Xiwu Chen, and Xiang Bai. Epnet: Enhancing point features with image semantics for 3d object detection. In *ECCV*, 2020.

[19] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *IROS*, 2018.

[20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019.

[21] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint*, 2022.

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017.

[23] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint*, 2022.

[24] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *CVPR*, 2021.

[25] AJ Piergiovanni, Vincent Casser, Michael S Ryoo, and Anelia Angelova. 4d-net for learned multi-modal alignment. In *CVPR*, 2021.

[26] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019.

[27] Cody Reading, Ali Harakeh, Julia Chae, and Steven L. Waslander. Categorical depth distributionnetwork for monocular 3d object detection. In *CVPR*, 2021.

[28] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *CVPR*, pages 14454–14463, 2021.

[29] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *CVPR*, 2020.

[30] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *CVPR*, 2021.

[31] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *CVPR*, 2019.

[32] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *CoRL*, 2022.

[33] Yue Wang and Justin M. Solomon. Object dgcnn: 3d object detection using dynamic graphs. In *NeurIPS*, 2021.

[34] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Bin Zhou, and Liangjun Zhang. FusionPainting: Multimodal fusion with adaptive attention for 3d object detection. *ITSC*, 2021.

[35] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 2018.

[36] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *CVPR*, 2021.

[37] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *NeurIPS*, 2021.

[38] Lin Zhao, Hui Zhou, Xinge Zhu, Xiao Song, Hongsheng Li, and Wenbing Tao. Lif-seg: Lidar and camera image fusion for 3d lidar semantic segmentation. *arXiv preprint*, 2021.

[39] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *CVPR*, 2018.

[40] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint*, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Section 4.
   (b) Did you describe the limitations of your work? [Yes] See supplementary material.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See supplementary material.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]