Cross Modal Transformer: Towards Fast and Robust 3D Object Detection

Junjie Yan Yingfei Liu ✉ Jianjian Sun Fan Jia Shuailin Li
Tiancai Wang Xiangyu Zhang
MEGVII Technology

Abstract

In this paper, we propose a robust 3D detector, named Cross Modal Transformer (CMT), for end-to-end 3D multi-modal detection. Without explicit view transformation, CMT takes the image and point clouds tokens as inputs and directly outputs accurate 3D bounding boxes. The spatial alignment of multi-modal tokens is performed by encoding the 3D points into multi-modal features. The core design of CMT is quite simple while its performance is impressive. It achieves 74.1% NDS (state-of-the-art with single model) on nuScenes test set while maintaining faster inference speed. Moreover, CMT has a strong robustness even if the LiDAR is missing. Code is released at https://github.com/junjie18/CMT.

1. Introduction

Multi-sensor fusion has shown its great superiority in autonomous driving system [31, 8, 22, 1, 27]. Different sensors usually provide the complementary information for each other. For instance, the camera captures information in a perspective view and the image contains rich semantic features while point clouds provide much more localization and geometry information. Taking full advantage of different sensors helps reduce the uncertainty and makes accurate and robust prediction.

Sensor data of different modalities usually has large discrepancy in distribution, making it hard to merge the multi-modalities. State-of-the-art (SoTA) methods tend to fuse the multi-modality by constructing unified bird's-eye-view (BEV) representation [31, 27, 22] or querying from tokens [1, 8]. For example, BEVFusion [31] explores a unified representation by BEV transformation for BEV feature fusion (see Fig. 1(a)). TransFusion [1] follows a two-stage pipeline and the camera images in second stage provide supplementary information for prediction refinement (see Fig. 1(b)). However, exploring a truly end-to-end pipeline for multi-sensor fusion remains to be a question.

✉ Corresponding author.

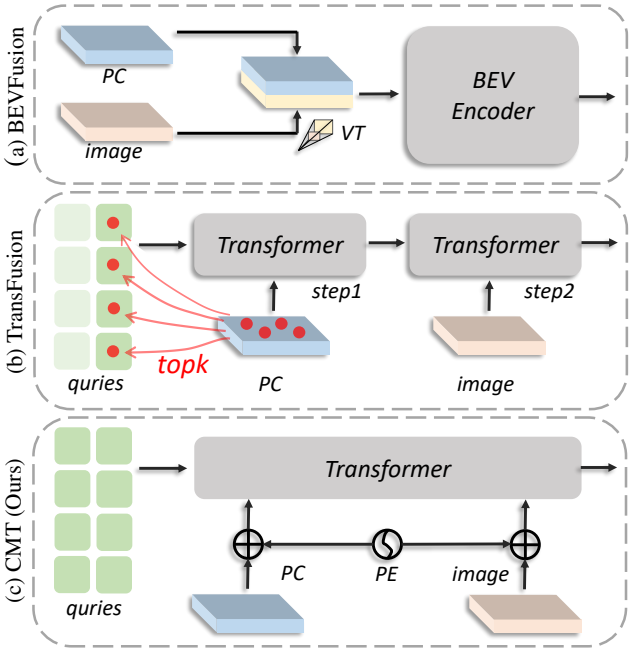

Figure 1: Comparison between BEVFusion, TransFusion, and our proposed CMT. (a) In BEVFusion, the camera features are transformed into BEV space by view transform. Two modality features are concatenated in BEV space and the BEV encoder is adopted for fusion. (b) TransFusion first generates the queries from the high response regions of LiDAR features. After that, object queries interact with point cloud features and image features separately. (c) In CMT, the object queries directly interact with multi modality features simultaneously. Position encoding (PE) is added to the multi-modal features for alignment. "VT" is the view transformation from image to 3D space.

Recently, the effectiveness of end-to-end object detection with transformer (DETR) [3, 60] has been proved in many perception tasks, such as instance segmentation [13, 15], multi-object tracking [55, 33] and visual 3D detection [47, 29, 30]. The DETR architecture is simple yet effective thanks to the object queries for representing different

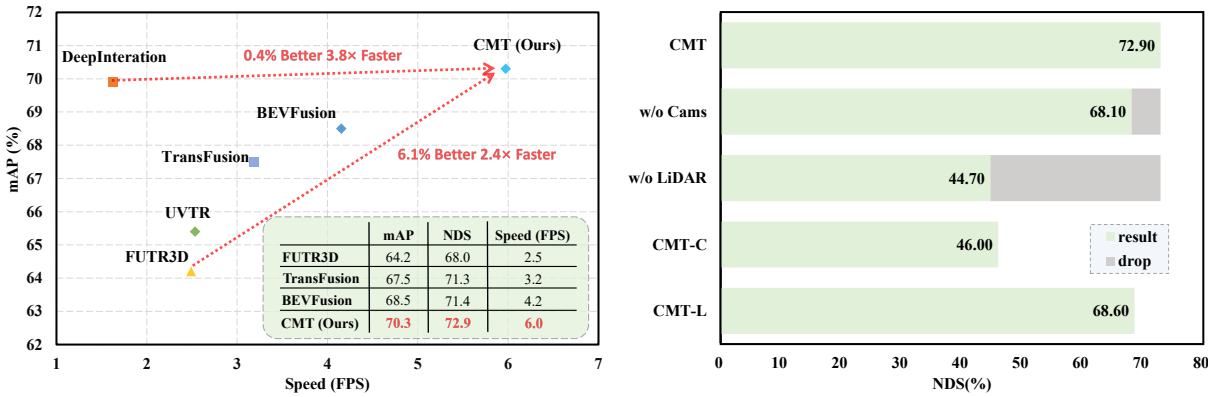

Figure 2: **Left**: Performance comparison between CMT and existing methods. All speed statistics are measured on a single Tesla A100 GPU using the best model of official repositories. **Right**: Performance evaluation of CMT under sensor missing. During inference, CMT achieves vision-based performance when LiDAR is missing, showing strong robustness.

instances and bipartite matching for one-to-one assignment.

Inspired by DETR, we aim to build an elegant end-to-end pipeline for multi-modal fusion in 3D object detection. In DETR, object queries directly interact with the image tokens through cross-attention in transformer decoder. For 3D object detection, one intuitive way is to concatenate the image and point cloud tokens together for further interaction with object queries. However, the concatenated tokens are disordered and unaware of their corresponding locations in 3D space. Therefore, it is necessary to provide the location prior for multi-modal tokens and object queries.

In this paper, we propose Cross-Modal Transformer (CMT), a simple yet effective end-to-end pipeline for robust 3D object detection (see Fig. 1(c)). First, we propose the Coordinates Encoding Module (CEM), which produces position-aware features, by encoding 3D points set implicitly into multi-modal tokens. Specifically, for camera images, 3D points sampled from frustum space are used to indicate the probability of 3D positions for each pixel. While for LiDAR, the BEV coordinates are simply encoded into the point cloud tokens. Next, we introduce the position-guided queries. Each query is initialized as a 3D reference point following PETR [29]. We transform the 3D coordinates of reference points to both image and LiDAR spaces, to perform the relative coordinates encoding in each space.

The proposed CMT framework brings many advantages compared to existing methods. Firstly, our method is a simple and end-to-end pipeline and can be easily extended. The 3D positions are encoded into the multi-modal features implicitly, which avoids introducing the bias caused by explicit cross-view feature alignment. Secondly, our method only contains basic operations, without the feature sampling or complex 2D-to-3D view transformation on multi-modal features. It achieves state-of-the-art performance and shows obvious superiority compared to existing approaches, as shown in the left graph of Fig. 2. Thirdly, the robustness of our CMT is much stronger than other existing approaches. Extremely, under the condition of LiDAR miss, our CMT with only image tokens can achieve similar performance compared to those vision-based 3D object detectors [29, 26] (see the right graph of Fig. 2).

To summarize, our contributions are:

- we propose a fast and robust 3D detector, which is a truly end-to-end framework without any post-process. It overcomes the sensor missing problem.
- The 3D positions are encoded into the multi-modal tokens, without any complex operations, like grid sampling and voxel-pooling.
- CMT achieves state-of-the-art 3D detection performance on nuScenes dataset. It provides a simple baseline for future research.

2. Related Work

2.1. Camera Based 3D Object Detection

Camera-based 3D object detection is one of the basic tasks in computer vision. Early works [45, 44] mainly follow the dense prediction pipeline. They first localize the objects on image plane and then predict their relevant 3D attributes, such as depth, size and orientation. However, with the surrounding cameras, the perspective-view based design requires elaborate post-processes to eliminate the redundant predictions of the overlapping regions. Recently, 3D object detection under the BEV has attracted increasing attention. The BEV representation provides a unified coordinate to fuse information from multiple camera views. LSS [35], BEVDet [17] and BEVDepth [24] predict the depth distribution to lift the image features to 3D frustum meshgrid. Besides, inspired by DETR [4], DETR3D [47] and BEVFormer [26] project the predefined BEV queries onto images and then employ the transformer attention to model

the relation of multi-view features. The above methods explicitly project the local image feature from 2D perspective view to BEV. Different from them, PETR [29, 30] and SpatialDETR [12] adopt the positional embedding that depends on the camera poses, allowing the transformer to implicitly learn the projection from image views to 3D space.

2.2. LiDAR Based 3D Object Detection

LiDAR-based 3D object detection aims to predict 3D object bounding boxes using the point clouds captured from LiDAR. Existing methods process the point cloud into different representations. Point-based methods [36, 37, 38, 39, 25, 53] directly extract features from raw point clouds and predict 3D bounding boxes. PointNet [37] is the first architecture to process the point cloud in an end-to-end manner, which preserves the spatial characteristics of the point cloud. Other methods project the unordered, irregular LiDAR point clouds onto a regular feature space such as 3D voxels [58, 51, 9, 10], feature pillars [19, 46, 54] and range images [14, 41]. Then the features are extracted in the BEV plane using a standard 2D backbone. VoxelNet [58] first divides the raw point clouds into regular voxel grids, and then uses PointNet network to extract features from the points in each voxel grid.

2.3. Multi-modal 3D Object Detection

Multi-sensor fusion in 3D detection has gained great attention in recent years. State-of-the-art (SoTA) methods tend to find a unified representation for both modalities, or define object queries to fuse the features for further prediction. For example, BEVFusion[31, 27] applies a lift-splat-shoot (LSS) operation to project image feature onto BEV space and concatenates it with LiDAR feature. UVTR[22] generates a unified representation in the 3D voxel space by deformable attention[60]. While for query-based methods, FUTR3D[8] defines the 3D reference points as queries and directly samples the features from the coordinates of projected planes. TransFusion[1] follows a two-stage pipeline. The proposals are generated by LiDAR features and further refined by querying the image features.

2.4. Transformer-based Object Detection

The pioneering work DETR [3] proposes a transformer-based detector paradigm without any hand-craft components, and has achieved state-of-the-arts in both 2D and 3D detection [57, 6, 26, 30]. However, DETR-like methods usually suffer from the slow convergence. To this end, many works [60, 56, 28, 21, 57, 5, 18] are proposed to improve the training efficiency from various aspects. Other improvements in 2D detection mainly focus on modifying the transformer layers[60, 56], designing informative object queries[28, 21, 57], or exploring the label assignment mechanism[5, 18]. Deformable DETR[60] proposes the deformable attention, which only attends to sampling points of local regions. SAM-DETR[56] presents a semantic aligner between object queries and encoded features to accelerate the matching process. To alleviate the instability of bipartite matching, DAB-DETR[28] formulates the object queries as dynamic anchor boxes, while DN-DETR[21] auxillarily reconstructs the ground-truths from the noisy ones. Based on them, DINO[57] further improves the denoising anchor boxes via a contrastive way.

3. Method

The overall architecture of the proposed CMT is illustrated in Fig. 3. Multi-view images and LiDAR points are fed into two individual backbones to extract multi-modal tokens. The 3D coordinates are encoded into the multi-modal tokens by the *coordinates encoding*. The queries from the *position-guided query generator* are used to interact with the multi-modal tokens in transformer decoder and then predict the object class as well as the 3D bounding boxes. The whole framework is learned in a fully end-to-end manner and the LiDAR backbone is trained from scratch without pretraining.

3.1. Coordinates Encoding Module

The coordinates encoding module (CEM) is used to encode the 3D position information into multi-modal tokens. It generates both the camera and BEV position encodings (PEs), which are added to image tokens and point cloud tokens respectively. With the help of CEM, multi-modal tokens can be implicitly aligned in 3D space.

Let $P(u, v)$ be the 3D points set corresponding to the feature map $F(u, v)$ of different modalities. Here (u, v) indicates the coordinate in the feature map. Specifically, F is the image feature for camera while BEV feature for LiDAR. Suppose the output position embedding of CEM is $\Gamma(u, v)$, its calculation can be formulated as:

$$\Gamma(u, v) = \psi(P(u, v)) \tag{1}$$

where ψ is a multi-layer perception (MLP) layer.

CE for Images. Since the image is captured from a perspective view, each pixel can be seen as an epipolar line in 3D space. Inspired by PETR [29], for each image, we encode a set of points in camera frustum space to perform the coordinates encoding. Given the image feature F_{im}, each pixel can be formulated as a series of points $\{p_k(u, v) = (u * d_k, v * d_k, d_k, 1)^T, k = 1, 2, ..., d\}$ in the camera frustum coordinates. Here, d is the number of points sampled along the depth axis. The corresponding 3D points can be calculated by:

$$p_k^{im}(u, v) = T_{c_i}^l K_i^{-1} p_k(u, v) \tag{2}$$

where $T_{c_i}^l \in R^{4\times4}$ is the transformation matrix from the i-th camera coordinate to the LiDAR coordinate. $K_i \in 4 \times 4$

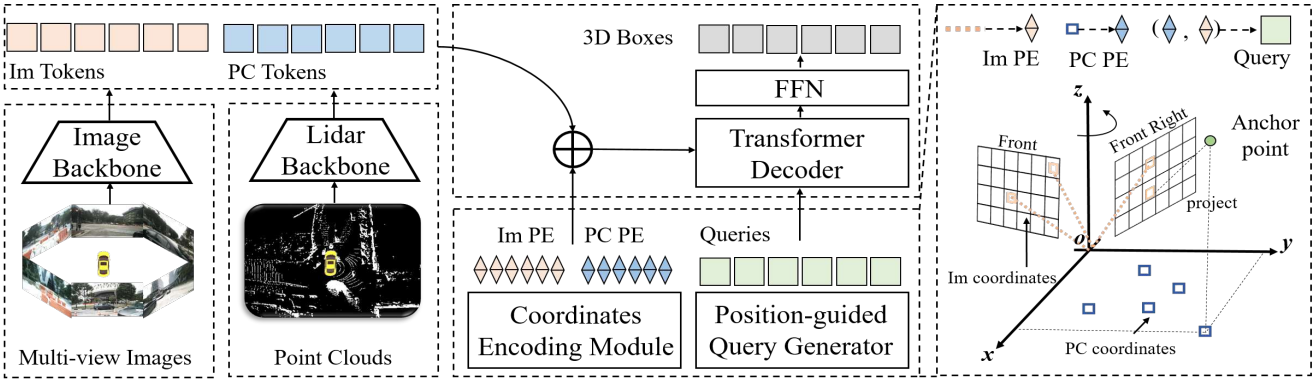

Figure 3: The architecture of Cross-Modal Transformer (CMT) paradigm. The multi-view images and point clouds are input to two backbone networks to extract feature tokens. In coordinates encoding module, coordinates of camera rays and BEV positions are transformed into the image position encoding (Im PE) and point cloud position encoding (PC PE), respectively. The queries are generated by the position-guided query generator. In query generator, 3D anchor points are projected to different modalities and the relative coordinates are encoded (see the right part). Multi-modal tokens further interact with queries in the transformer decoder. The updated queries are further used to predict the 3D bounding boxes.

is the intrinsic matrix of i-th camera. The position encoding of pixel (u, v) for image is formulated as:

$$\Gamma_{im}(u,v) = \psi_{im}(\{p_k^{im}(u,v), \quad k = 1, 2, ..., d\}) \tag{3}$$

CE for Point Clouds. We choose VoxelNet[51, 58] or PointPillar[19] as backbone to encode the point cloud tokens F_{pc}. Intuitively, the point set P in Eq. (1) can be sampled along the Z-axis. Suppose (u, v) is the coordinates in BEV feature map, the sampled point set is then $p_k(u, v) = (u, v, h_k, 1)^T$, where h_k indicates the height of k-th points and $h_0 = 0$ as default. The corresponding 3D points of BEV feature map can be calculated by:

$$p_k^{pc}(u,v) = (u * u_d, v * v_d, h_k, 1) \tag{4}$$

where (u_d, v_d) is the size of each BEV feature grid. To simplify, we only sample one point along the height axis. It is equivalent to the 2D coordinate encoding in BEV space. The position embedding of point cloud can be obtained by:

$$\Gamma_{pc}(u,v) = \psi_{pc}(\{p_k^{pc}(u,v), \quad k = 1, 2, ..., h\}) \tag{5}$$

3.2. Position-guided Query Generator

Following Anchor-DETR [48] and PETR [29], we firstly initialize the queries with n anchor points $A = \{a_i = (a_{x,i}, a_{y,i}, a_{z,i}), i = 1, 2, ..., n\}$ sampled from uniform distribution between $[0, 1]$. Then these anchor points are transformed into 3D world space by linear transformation:

$$\begin{cases} a_{x,i} = & a_{x,i} * (x_{max} - x_{min}) + x_{min} \\ a_{y,i} = & a_{y,i} * (y_{max} - y_{min}) + y_{min} \\ a_{z,i} = & a_{z,i} * (z_{max} - z_{min}) + z_{min} \end{cases} \tag{6}$$

where $[x_{min}, y_{min}, z_{min}, x_{max}, y_{max}, z_{max}]$ is the region of interest (RoI) of 3D world space. After that, we project the 3D anchor points A to different modalities and encode the corresponding point sets by CEM. Then the positional embedding Γ_q of object queries can be generated by:

$$\Gamma_q = \psi_{pc}(A_{pc}) + \psi_{im}(A_{im}) \tag{7}$$

where A_{pc} and A_{im} are the point set projected on BEV plane and image plane, respectively. The positional embedding Γ_q are further added with the query content embedding to generate the initial position-guided queries Q_0.

3.3. Decoder and Loss

As for the decoder, we follow the original transformer decoder in DETR [48] and use L decoder layers. For each decoder layer, the position-guided queries interact with the multi-modal tokens and update their representations. Two feed-forward networks (FFNs) are used to predict the 3D bounding boxes and the classes using updated queries. We formulate the prediction process of each decoder layer as follows:

$$\hat{b}_i = \Psi^{reg}(Q_i), \hat{c}_i = \Psi^{cls}(Q_i), \tag{8}$$

where Ψ^{reg} and Ψ^{cls} respectively represent the FFN for regression and classification. Q_i is the the updated object queries of the i-th decoder layer.

For set prediction, the bipartite matching is applied for one-to-one assignment between predictions and ground-truths. We adopt the focal loss for classification and $L1$ loss for 3D bounding box regression:

$$L(y, \hat{y}) = \omega_1 L_{cls}(c, \hat{c}) + \omega_2 L_{reg}(b, \hat{b}) \tag{9}$$

Table 1: Performance comparison on the nuScenes **test** set. "L" is LiDAR and "C" is camera.

Methods	Modality	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓
BEVDet [17]	C	0.488	0.424	0.524	0.242	0.373	0.950	0.148
DETR3D [47]	C	0.479	0.412	0.641	0.255	0.394	0.845	0.133
PETR [29]	C	0.504	0.441	0.593	0.249	0.383	0.808	0.132
CenterPoint [54]	L	0.673	0.603	0.262	0.239	0.361	0.288	0.136
UVTR [22]	L	0.697	0.639	0.302	0.246	0.350	0.207	0.123
TransFusion [1]	L	0.702	0.655	0.256	0.240	0.351	0.278	0.129
PointPainting[42]	LC	0.610	0.541	0.380	0.260	0.541	0.293	0.131
PointAugmenting[43]	LC	0.711	0.668	0.253	0.235	0.354	0.266	0.123
MVP[7]	LC	0.705	0.664	0.263	0.238	0.321	0.313	0.134
FusionPainting[50]	LC	0.716	0.681	0.256	0.236	0.346	0.274	0.132
UVTR [22]	LC	0.711	0.671	0.306	0.245	0.351	0.225	0.124
TransFusion [1]	LC	0.717	0.689	0.259	0.243	0.359	0.288	0.127
BEVFusion [31]	LC	0.729	0.702	0.261	0.239	0.329	0.260	0.134
DeepInteration [52]	LC	0.734	0.708	0.257	0.240	0.325	0.245	0.128
CMT-C	C	0.481	0.429	0.616	0.248	0.415	0.904	0.147
CMT-L	L	**0.701**	**0.653**	0.286	0.243	0.356	0.238	0.125
CMT	LC	**0.741**	**0.720**	0.279	0.235	0.308	0.259	0.112

Table 2: Performance comparison on the nuScenes **val** set. "L" is LiDAR and "C" is camera.

Methods	modality	NDS↑	mAP↑
FUTR3D [8]	L	0.655	0.593
UVTR [22]	L	0.676	0.608
TransFusion [1]	L	0.701	0.651
FUTR3D [8]	LC	0.683	0.645
UVTR [22]	LC	0.702	0.654
TransFusion [1]	LC	0.713	0.675
BEVFusion [31]	LC	0.714	0.685
DeepInteration [52]	LC	0.726	0.699
CMT-C	C	0.460	0.406
CMT-L	L	0.686	0.624
CMT	LC	**0.729**	**0.703**

where ω_1 and ω_2 are the hyper-parameter to balance the two loss terms. Note that for positive and negative queries in query denoising, the loss is calculated in the same way.

3.4. Masked-Modal Training for Robustness

Security is the most important concern for autonomous driving systems. An ideal system requires solid performance even if part of them fails, as well as not relying on any input of a specific modality. Recently, BEVFusion [27] has explored the robustness of LiDAR sensor failure. However, the exploration is limited to restricted scan range and model need be retrained. In this paper, we try more extreme failures, including single camera miss, camera miss and LiDAR miss, as shown in Fig. 4. It is consistent with the actual scene and ensures the safety of autonomous driving.

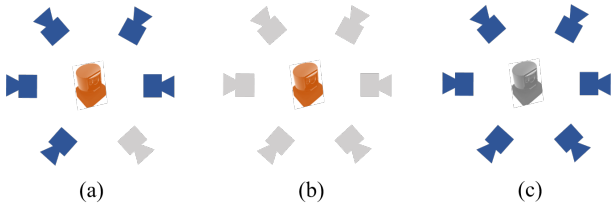

Figure 4: We analyze the system robustness of CMT at test period under three simulated sensor errors: (a) single camera miss, (b) all camera miss and (c) LiDAR miss.

To improve the robustness of the model, we propose a training strategy, called masked-modal training. In training process, we randomly use only a single modality for training, such as camera or LiDAR, with the ratio of η_1 and η_2. This strategy ensures that the model are fully trained with both single modal and multi-modal. Then the model can be tested with single modal or multi-modal, without modifying the model weight. The experimental results show that masked-modal training will not affect the performance of our fusion model. Even if LiDAR is damaged, it can still achieve similar performance compared to the SoTA vision-based 3D detectors [29, 17] (see Tab. 3-4).

Table 3: Quantitative results on the nuScenes val with LiDAR or camera miss. With the masked-modal training, the efficiency and robustness of our CMT is significantly improved, especially when the LiDAR camera is missed.

Metric	Vanilla training			Masked-modal training		
	CMT	only LiDAR	only Cams	CMT	only LiDAR	only Cams
NDS ↑	0.726	0.603	0.073	0.729 (↑0.3%)	0.681 (↑**7.8%**)	0.447 (↑**37.4%**)
mAP ↑	0.691	0.487	0.000	0.703 (↑1.2%)	0.617 (↑**13.0%**)	0.383 (↑**38.3%**)

Table 4: NDS/mAP comparison on nuScenes val with sensor miss. BEVFusion is trained with mask-modal strategy. * means our reproduced result.

Model	Test modal		
	Both	only LiDAR	only Cams
TransFusion[1]	0.71/0.67	0.70/0.65	None
BEVFusion[31]*	0.72/0.68	0.68/0.63	0.40/0.32
CMT	0.73/0.70	0.68/0.62	**0.45/0.38**

Table 5: CDS/AP comparison on Argoverse2 **val** set. "L" is LiDAR and "C" is camera.

Model	Modality	AP	CDS
VoxelNeXt[11]	L	0.307	-
FSF[23]	LC	0.332	0.255
CMT	LC	**0.361**	**0.278**

3.5. Discussion

CMT shares similar motivation with FUTR3D [8] on the end-to-end modeling. However, both the method and its effectiveness are totally different. FUTR3D repeatly samples the corresponding features from each modal and then performs the cross-modal fusion. CMT conducts the position encoding for both multi-view images and point clouds, which are simply added with corresponding modal tokens, removing the repeated projection and sampling processes. It keeps more end-to-end spirits in original DETR framework. Moreover, CMT achieves much better performance compared to the FUTR3D (see comparison in Tab. 1), showing its superior effectiveness. We think CMT provides a better end-to-end solution for multi-modal object detection.

4. Experiments

4.1. Datasets and Metrics

We evaluate our method on open datasets, including nuScenes [2] and Argoverse 2 [49].

NuScenes [2] is a large-scale multi-modal dataset, which is composed of data from 6 cameras, 1 LiDAR and 5 radars. The dataset has 1000 scenes totally and is divided into 700/150/150 scenes as train/validation/test sets, respectively. Each scene has $20s$ video frames with 12 FPS. 3D bounding boxes are annotated every $0.5s$. We only use these key frames. In each frame, nuScenes provides images from six cameras. NuScenes provides a 32-beam LiDAR with 20 FPS. The key frames are also annotated every 0.5s, the same as cameras. We follow the common practice to transform the points from the past 9 frames to the current frame for training and evaluation. We follow the nuScenes official metrics.

We report the nuScenes Detection Score (NDS), mean Average Precision (mAP), mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error(mAOE), mean Average Velocity Error (mAVE) and mean Average Attribute Error (mAAE).

Argoverse 2(AV2) [49] contains 1000 sequences in total, 700/150/150 for train/validation/test similar as nuScenes. AV2 privides a long perceptron range up to 200 meters, covering an area of $400m \times 400m$, which is much larger than nuScenes. We report mean Average Precision(mAP), Composite Detection Score(CDS).

4.2. Implementation Details

We use ResNet[16] or VoVNet[20] as image backbone to extract the 2D image features. The C5 feature is upsampled and fused with C4 feature to produce P4 feature. We use VoxelNet [58] or PointPillars [19] as the backbone to extract the point-cloud features. All the feature dimension is set to 256, including the LiDAR feature, image feature and query embedding. Six decoder layers are adopted in transformer decoder.

Our model is trained with the batch size of 16 on 8 A100 GPUs. It is trained for total 20 epochs with CBGS[59]. We adopt the AdamW[32] optimizer for optimization. The initial learning rate is 1.0×10^{-4} and we follow the cycle learning rate policy[40]. The mask ratios η_1 and η_2 are both set to 0.25 for masked-modal training. The GT sample augmentation is employed for the first 15 epochs and closed for the rest epochs. As for the loss weights, we follow the default setting in DETR3D [47] and set the ω_1 and ω_2 to 2.0 and 0.25, respectively. For fast convergence, we introduce the point-based query denoising strategy based on DN-DETR [21]. Different from it, we generate the noisy anchor points by center shifting since the box scale is not that important in 3D object detection.

Figure 5: Some qualitative detection results on the surrounding views and BEV space in the nuScenes test set. Bounding boxes with different colors represent vehicles(●), pedestrians(●), Bus(●) and Truck(●).

Figure 6: Visualization of attention maps on multi-view images. The blue points (●) are initial anchor points while red points (●) are the centers of box predictions. It shows that high response regions of attention maps mainly focus on the foreground objects, which are close to the anchor points.

On AV2, our model is trained 6 epochs, following common practice[11, 23].

4.3. State-of-the-Art Comparison

As shown in Tab. 1, CMT achieves state-of-the-art performance compared to existing methods on nuScenes test set. Our LiDAR-only baseline, named CMT-L, achieves 70.1% NDS, which is a nearly SoTA performance among all existing LiDAR-only methods. Our multi-modal CMT achieves 74.1% NDS and 72.0% mAP, outperforming all existing SoTA approaches, such as BEVFusion [31] and DeepInteration [52]. We also compare the performance with other SoTA methods on nuScenes val set (see Tab. 2). It shows that our proposed CMT with multi-modal fusion outperforms the BEVFusion by 1.8% mAP. CMT introduces large performance improvements compared to our LiDAR-only CMT-L by 4.0%/6.7% and 4.3%/7.9% NDS/mAP on test and validation set, respectively. In comparison, TransFusion only brings 1.5%/3.4% NDS/mAP on test set, compared to the LiDAR-only TransFusion. It shows that the multi-view images bring much more complementary information to the point clouds in CMT framework. We think the end-to-end modeling of CMT relatively improves the importance of image tokens. Fig. 5 shows some qualitative detection results on the nuScenes test set.

On AV2 dataset, CMT also outperforms existing SoTA methods, including VoxelNeXt[11] and FSF[23], as shown in Tab. 5.

4.4. Strong Robustness

We evaluate the robustness of our framework under various harsh environments, including LiDAR miss and camera miss. Tab. 3 shows the results when the sensor miss occurs, by simulating the scenarios of any modality totally broken. The performance is compared between the vanilla training and masked-modal training. It validates the effect of masked-modal training. Note that the model are only trained with multi-modality and evaluated without any finetune process. With vanilla training, the model fails to predict anything meaningful (only Cams with mAP=0) when LiDAR is missing. With masked-modal training, the absence of LiDAR or camera modalities lead to 4.8% and 28.2% NDS drop compared to CMT, respectively. It is observed that losing one modality still remains similar results compared to single-modal training settings. It overcomes the drawback that multi-modal method usually rely on one major modality and performance would degrade significantly if losing the major modality. Especially, for the case of LiDAR missing, the performance is still comparable to the SoTA camera-only method PETR [29], validating the strong robustness of our method. We further evaluate the performance of TransFusion and BEVFusion under sensor miss (see Tab. 4). TransFusion fails to work when LiDAR is missing due to the two-stage design. With the masked-modal training, BEVFusion achieves the decent performance (40% NDS and 32% mAP), while showing large inferiority compared to CMT.

Moreover, we also investigate the case when any one of cameras fails. Experimental result shows slight performance drop, indicating the tolerable to single camera miss

Table 6: The ablation studies of different components in the proposed CMT.

Im	PC	NDS	mAP	mATE	mASE	mAOE
✓		0.595	0.554	0.515	0.258	0.429
	✓	0.665	0.626	**0.372**	**0.255**	**0.347**
✓	✓	**0.669**	**0.641**	0.377	0.254	0.375

(a) Position encoding for query.

PQD	NDS	mAP	mATE	mASE	mAOE
	0.626	0.584	0.429	0.259	0.420
✓	**0.669**	**0.641**	**0.377**	**0.254**	**0.375**

(b) Point-based query denoising.

Voxel size	NDS	mAP	mATE	mASE	mAOE
0.075	0.669	**0.641**	**0.377**	0.254	0.375
0.1	**0.671**	0.638	0.378	**0.252**	**0.334**
0.125	0.655	0.624	0.396	0.255	0.397

(c) Voxel size of LiDAR backbone.

Backbone	NDS	mAP	mATE	mASE	mAOE
ResNet-50	0.658	0.623	**0.376**	**0.253**	0.399
ResNet-101	0.664	0.629	0.383	0.254	**0.363**
VoV-99	**0.669**	**0.641**	0.377	0.254	0.375

(d) Image backbone.

Image size	NDS	mAP	mATE	mASE	mAOE
800×320	0.654	0.609	**0.374**	0.256	0.389
1600×640	**0.669**	**0.641**	0.377	**0.254**	**0.375**

(e) Input size of image backbone.

Backbone	NDS	mAP	mATE	mASE	mAOE
PointPillars	0.628	0.598	0.430	**0.252**	0.455
VoxelNet	**0.669**	**0.641**	**0.377**	0.254	**0.375**

(f) Lidar backbone

of our method. Six sensors brings an average decrease of 0.7% NDS, no more than 1% performance of the oracle version. The front and back sensor relatively play the important role among camera sensors, with 1.1% and 0.8% decrease respectively, due to their distant or large field of view. Compared to the camera-only setting, our multi-modal framework facilitate the compensation between LiDAR and image domains, thus presenting a robust performance.

4.5. Ablation Study

We present ablation studies in Tab. 6. All experiments are conducted for 20 epochs without CBGS[34]. We first ablate the effect of Im PE and PC PE on the generation of position-guided queries. It shows that removing PC PE introduces a $7.4\%/8.70\%$ NDS/mAP performance drop, which is much larger than the drop of removing Im PE $0.4\%/1.5\%$. Next, we explore the effectiveness of point-based query denoising (PQD) introduced in Sec. 4.2. We can easily find that PQD can greatly improve the overall performance by $4.3\%/5.7\%$ NDS/mAP. With PQD, the training convergence can be boosted, which is similar to the practice in DN-DETR [21]. Further, we also illustrate the effect of scaling up the CMT model as well as the input size. Overall, CMT can benefit from the scaling model size. Interestingly, we find increasing the voxel number (smaller voxel size) and image size achieves similar improvements $\approx 1.5\%$ in NDS. While scaling the image size increases more mAP than the voxel number(+3.2% *vs.* +1.7%). When increasing the image size from 800×320 to 1600×640, we find the performance improvements are mainly from these small objects, such as pedestrian and motorcycle. We also conduct experiments on replacing image and LiDAR backbones, we use VoV-99[20] and ResNet[16] as our image backbones. Experiments show that our proposed CMT can benefit from larger backbones. For image, VoV-99 backbone achieves the best result and outperforms the ResNet-50 by $1.1\%/1.8\%$ in NDS/mAP. While for LiDAR, VoxelNet outperforms the PointPillar by $4.1\%/4.3\%$ in NDS/mAP.

4.6. Analysis

For better understanding on querying from multi-modal tokens, we visualize the attention map of cross-attention on the multi-view images (see Fig. 6). We can clearly find that the attention maps have higher response on the regions that includes foreground objects. It shows that our method can implicitly achieve the cross-modal interaction. We visualize the initial anchor points and the center points of predictions. Most anchor points focus on the closest foreground objects. After the interaction with multi-modal tokens in the transformer decoder, anchor points are updated and gradually access the accurate center points.

5. Conclusions

In this paper, we propose a fully end-to-end framework for multi-modal 3D object detection. It implicitly encodes the 3D coordinates into the tokens of images and point clouds. With the coordinates encoding, the simple yet effective DETR pipeline can be adopted for multi-modal fusion and end-to-end learning. With masked-modal training, our multi-modal detector can be learned with strong robustness, even if one of multi-modalities are missed. We hope such a simple pipeline design could provide more insights on the end-to-end 3D object detection.

Acknowledgements: This research was supported by National Key R&D Program of China (No. 2017YFA0700800) and Beijing Academy of Artificial Intelligence (BAAI).

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1090–1099, 2022. 1, 3, 5, 6

[2] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1, 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[5] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 3

[6] Qiang Chen, Jian Wang, Chuchu Han, Shangang Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaodi Wang, Shumin Han, Gang Zhang, Haocheng Feng, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Group detr v2: Strong object detector with encoder-decoder pretraining. 2022. 3

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 5

[8] Xuanyao Chen, Tianyuan Zhang, Yue Wang, Yilun Wang, and Hang Zhao. Futr3d: A unified sensor fusion framework for 3d detection. *arXiv preprint arXiv:2203.10642*, 2022. 1, 3, 5, 6

[9] Yukang Chen, Yanwei Li, Xiangyu Zhang, Jian Sun, and Jiaya Jia. Focal sparse convolutional networks for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5428–5437, 2022. 3

[10] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Largekernel3d: Scaling up kernels in 3d sparse cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13488–13498, 2023. 3

[11] Yukang Chen, Jianhui Liu, Xiangyu Zhang, Xiaojuan Qi, and Jiaya Jia. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21674–21683, 2023. 6, 7

[12] Simon Doll, Richard Schulz, Lukas Schneider, Viviane Benzin, Markus Enzweiler, and Hendrik Lensch. Spatialdetr: Robust scalable transformer-based 3d object detection from multi-view camera images with global cross-sensor attention. In *European Conference on Computer Vision*, pages 230–245. Springer, 2022. 3

[13] Bin Dong, Fangao Zeng, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Solq: Segmenting objects by learning queries. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[14] Lue Fan, Xuan Xiong, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Rangedet: In defense of range view for lidar-based 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2918–2927, 2021. 3

[15] Yuxin Fang, Shusheng Yang, Xinggang Wang, Yu Li, Chen Fang, Ying Shan, Bin Feng, and Wenyu Liu. Instances as queries. *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 8

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 5

[18] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. *arXiv preprint arXiv:2207.13080*, 2022. 3

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 3, 4, 6

[20] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13906–13915, 2020. 6, 8

[21] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 3, 6, 8

[22] Yanwei Li, Yilun Chen, Xiaojuan Qi, Zeming Li, Jian Sun, and Jiaya Jia. Unifying voxel-based representation with transformer for 3d object detection. *arXiv preprint arXiv:2206.00630*, 2022. 1, 3, 5

[23] Yingyan Li, Lue Fan, Yang Liu, Zehao Huang, Yuntao Chen, Naiyan Wang, Zhaoxiang Zhang, and Tieniu Tan. Fully sparse fusion for 3d object detection. *arXiv preprint arXiv:2304.12310*, 2023. 6, 7

[24] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*, 2022. 2

[25] Zhichao Li, Feng Wang, and Naiyan Wang. Lidar r-cnn: An efficient and universal 3d object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7546–7555, 2021. 3

[26] 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 arXiv:2203.17270*, 2022. 2, 3

[27] Tingting Liang, Hongwei Xie, Kaicheng Yu, Zhongyu Xia, Zhiwei Lin, Yongtao Wang, Tao Tang, Bing Wang, and Zhi Tang. Bevfusion: A simple and robust lidar-camera fusion framework. *arXiv preprint arXiv:2205.13790*, 2022. 1, 3, 5

[28] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 3

[29] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *arXiv preprint arXiv:2203.05625*, 2022. 1, 2, 3, 4, 5, 7

[30] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. 1, 3

[31] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv2205.13542*, 2022. 1, 3, 5, 6, 7

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 6

[33] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. Trackformer: Multi-object tracking with transformers. *arXiv preprint arXiv:2101.02702*, 2021. 1

[34] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6181–6189, 2018. 8

[35] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 2

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 3

[37] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas 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*, pages 652–660, 2017. 3

[38] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 3

[39] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 3

[40] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. 6

[41] Pei Sun, Weiyue Wang, Yuning Chai, Gamaleldin Elsayed, Alex Bewley, Xiao Zhang, Cristian Sminchisescu, and Dragomir Anguelov. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2021. 3

[42] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020. 5

[43] Chunwei Wang, Chao Ma, Ming Zhu, and Xiaokang Yang. Pointaugmenting: Cross-modal augmentation for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11794–11803, 2021. 5

[44] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*, pages 1475–1485. PMLR, 2022. 2

[45] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. 2

[46] Yue Wang, Alireza Fathi, Abhijit Kundu, David A Ross, Caroline Pantofaru, Tom Funkhouser, and Justin Solomon. Pillar-based object detection for autonomous driving. In *European Conference on Computer Vision*, pages 18–34. Springer, 2020. 3

[47] Yue Wang, Guizilini Vitor Campagnolo, Tianyuan Zhang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *In Conference on Robot Learning*, pages 180–191, 2022. 1, 2, 5, 6

[48] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. *arXiv preprint arXiv:2109.07107*, 2021. 4

[49] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 6

[50] Shaoqing Xu, Dingfu Zhou, Jin Fang, Junbo Yin, Zhou Bin, and Liangjun Zhang. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3047–3054. IEEE, 2021. 5

[51] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 3, 4

[52] Zeyu Yang, Jiaqi Chen, Zhenwei Miao, Wei Li, Xiatian Zhu, and Li Zhang. Deepinteraction: 3d object detection via modality interaction. *arXiv preprint arXiv:2208.11112*, 2022. 5, 7

[53] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 3

[54] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11784–11793, 2021. 3, 5

[55] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. Motr: End-to-end multiple-object tracking with transformer. In *European Conference on Computer Vision*, pages 659–675. Springer, 2022. 1

[56] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 949–958, June 2022. 3

[57] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[58] Yin Zhou and Oncel 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*, pages 4490–4499, 2018. 3, 4, 6

[59] Benjin Zhu, Zhengkai Jiang, Xiangxin Zhou, Zeming Li, and Gang Yu. Class-balanced grouping and sampling for point cloud 3d object detection. *arXiv preprint arXiv:1908.09492*, 2019. 6

[60] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 3