CenterFormer: Center-Based Transformer for 3D Object Detection

Zixiang Zhou[1,2(✉)], Xiangchen Zhao[1], Yu Wang[1], Panqu Wang[1], and Hassan Foroosh[2]

[1] TuSimple, San Diego, USA
{xiangchen.zhao,yu.wang,panqu.wang}@tusimple.ai
[2] Computational Imaging Lab., University of Central Florida, Orlando, USA
zhouzixiang@knights.ucf.edu, hassan.foroosh@ucf.edu

Abstract. Query-based transformer has shown great potential in constructing long-range attention in many image-domain tasks, but has rarely been considered in LiDAR-based 3D object detection due to the overwhelming size of the point cloud data. In this paper, we propose **CenterFormer**, a center-based transformer network for 3D object detection. CenterFormer first uses a center heatmap to select center candidates on top of a standard voxel-based point cloud encoder. It then uses the feature of the center candidate as the query embedding in the transformer. To further aggregate features from multiple frames, we design an approach to fuse features through cross-attention. Lastly, regression heads are added to predict the bounding box on the output center feature representation. Our design reduces the convergence difficulty and computational complexity of the transformer structure. The results show significant improvements over the strong baseline of anchor-free object detection networks. CenterFormer achieves state-of-the-art performance for a single model on the Waymo Open Dataset, with 73.7% mAPH on the validation set and 75.6% mAPH on the test set, significantly outperforming all previously published CNN and transformer-based methods. Our code is publicly available at https://github.com/TuSimple/centerformer

Keywords: LiDAR point cloud · 3D object detection · Transformer · Multi-frame fusion

1 Introduction

LiDAR is an important sensing and perception tool in autonomous driving due to its ability to provide highly accurate 3D point cloud data of the scanned

Z. Zhou– Work done during an internship at TuSimple.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-19839-7_29.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13698, pp. 496–513, 2022.
https://doi.org/10.1007/978-3-031-19839-7_29

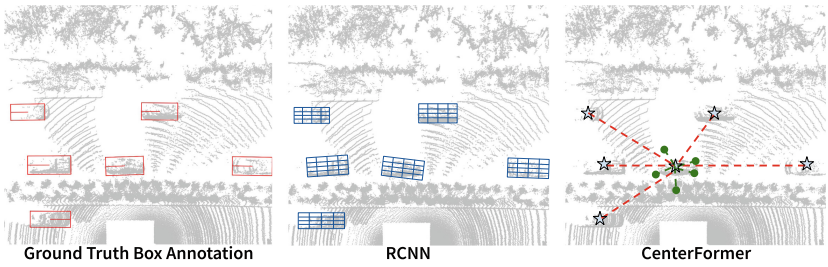

Fig. 1. The comparison of CenterFormer with RCNN-style detector. RCNN aggregates point or grid features in RoI, while CenterFormer can learn object-level contextual information and long range features through an attention mechanism.

environment. LiDAR-based 3D object detection aims to detect the bounding boxes of the objects in the LiDAR point cloud. Compared to image-domain object detection, the scanned points in LiDAR data may be sparse and irregularly spaced depending on the distance from the sensor. Most recent methods rely on discretizing the point clouds into voxels [47,58] or projected bird's eye view (BEV) feature maps [18] to use 2D or 3D convolution networks. Sometimes, it requires a second stage RCNN [11]-style refinement network to compensate for the information loss in the voxelization. However, current two-stage networks [34,51] lack contextual and global information learning. They only use the local features of the proposal (RoI) to refine the results. The features in other boxes or neighboring positions that could also be beneficial to the refinement are neglected. Moreover, the environment of the autonomous driving scene is not stationary. The local feature learning has more limitations when using a sequence of scans.

In the image domain, transformer encoder-decoder structure has become a competitive method for the detection [4,62] and segmentation [2,42] tasks. The transformer is able to capture long-range contextual information in the whole feature map and different feature domains. One of the most representative methods is DETR [4], which uses the parametric query to directly learn object information from an encoder-decoder transformer. DETR is trained end-to-end as a set matching problem to avoid any handcrafting processes like non-maximum suppression (NMS). However, there are two major problems in the DETR-style encoder-decoder transformer network: First, the computational complexity grows quadratically as the input size increases. This limits the transformer to take only low-dimensional features as input which leads to low performance on small objects. Second, the query embedding is learned through the network so that the training is hard to converge.

Can we design a transformer encoder-decoder network for the LiDAR point cloud in order to better perceive the global connection of point cloud data? Considering the sheer size of LiDAR point cloud data, and the relatively small sizes of objects to be detected, voxel or BEV feature map representations need to be large enough to keep the features for such objects to be separable. As a

result, it is impractical to use the transformer encoder structure on the feature map due to the large input size. In addition, if we use a large feature map for the transformer decoder, the query embedding is also difficult to focus on meaningful attention during training. To mitigate these converging problems, one solution is to provide the transformer with a good initial query embedding and confine the attention learning region to a smaller range. In the center-based 3D object detection network [51], the feature at the center of an object is used to capture all object information, hence the center feature is a good substitute for the object feature embedding. Multi-scale image pyramid and deformable convolution [16] are two common methods to increase the receptive field of the feature learning without significantly increasing the complexity. Some recent works [15,62] apply these two methods in the transformer networks.

Taking the aforementioned aspects into consideration, we propose a center-based transformer network, named **Center Transformer (CenterFormer)**, for 3D object detection. Specifically, we first use a standard voxel-based backbone network to encode the point cloud into a BEV feature representation. Next, we employ a multi-scale center proposal network to convert the feature into different scales and predict the initial center locations. The feature at the proposed center is fed into a transformer decoder as the query embedding. In each transformer module, we use a deformable cross attention layer to efficiently aggregate the features from the multi-scale feature map. The output object representation then regresses to other object properties to create the final object prediction. As shown in Fig. 1, our method can model object-level connection and long-range feature attention. To further explore the ability of the transformer, we also propose a multi-frame design to fuse features from different frames through cross-attention. We test CenterFormer on the large scale Waymo Open Dataset [38] and the nuScenes dataset [3]. Our method outperforms the popular center-based 3D object detection networks which are dominant on public benchmarks by a large margin, achieving state-of-the-art performance, with 73.7% and 75.6% mAPH on the waymo validation and test sets, respectively. The contributions of our method can be summarised as follows:

- We introduce a center-based transformer network for 3D object detection.
- We use the center feature as the initial query embedding to facilitate learning of the transformer.
- We propose a multi-scale cross-attention layer to efficiently aggregate neighboring features without significantly increasing the computational complexity.
- We propose using the cross-attention transformer to fuse object features from different frames.
- Our method outperforms all previously published methods by a large margin, setting a new state-of-the-art performance on the Waymo Open Dataset.

2 Related Work

2.1 LiDAR-Based 3D Object Detection

Compared to the well-established point cloud processing networks like PointNet [31] and PointNet++ [32], most recent LiDAR detection and segmentation methods [18,54,58,59,61] voxelize the point cloud in a fixed 3D space into a BEV/voxel representation and use conventional 2D/3D convolutional networks to predict the 3D bounding boxes. Other methods [1,9,39,44] detect the objects on a projected range image. There are also some methods that use hybrid features along with the voxel network [30,34,49], and combine multi-view features in the voxel feature representation [34]. VoxelNet [58] uses a PointNet inside each voxel to encode all points into a voxel feature. This feature encoder later became an essential method in voxel-based point cloud networks. PointPillar [18] proposes the pillar feature encoder to directly encode the point cloud into the BEV feature map so that only 2D convolution is needed in the network.

Similar to image object detection, 3D object detection methods can be divided into anchor-based [18,34,36,47] and anchor-free [10,51] methods. Anchor-based methods detect objects through a classification of all predefined object anchors, while anchor-free methods generally consider objects as keypoints and find those keypoints at the local heatmap maxima. Even though anchor-based methods can achieve a good performance, they rely heavily on hyper-parameter tuning. On the other hand, as the anchor-free methods become more prevailing in image-domain tasks, many 3D and LiDAR works have adopted the same design and show a more efficient and competitive performance. Many works [6,20,24,34] also require an RCNN-style second stage refinement. Feature maps for each bounding box proposal are aggregated through RoIAlign or RoIPool. CenterPoint [51] detects objects using center heatmap and regresses other bounding box information using center feature representation.

Most methods directly concatenate points from different frames based on the ego-motion estimation to use the multi-frame information. This assumes the model can align object features from different frames. However, independently moving objects cause misalignment of features across frames. Recent multi-frame methods [14,50] use an LSTM or a GNN module to fuse the previous state feature with the current feature map. 3D-MAN [48] uses a multi-frame alignment and aggregation module to learn the temporal attention of predictions from multiple frames. The feature of each box is generated from the RoI pooling.

2.2 Vision Transformer

Originally proposed in the Natural Language Processing (NLP) community, transformer [40] is becoming a competitive feature learning module in computer vision. Compared to traditional CNN, the transformer has a bigger receptive field, and feature aggregation is based on the response learned directly from pairwise features. A transformer encoder [7,15,52] usually serves as a replacement for the convolution layer in the backbone network. Meanwhile, the transformer

decoder uses high-level query feature embedding as the input and extracts features from feature encoding through cross-attention, which is more common in detection and segmentation tasks [4,42,57,62]. DETR [4] uses a transformer encoder-decoder structure to predict objects from learned query embedding. Deformable DETR [62] improves the DETR training through a deformable attention layer. Some recent methods [22,26,53] show that DETR is easier to converge using guidance like anchor boxes.

2.3 3D Transformer

An important design in transformer structure is the position embedding due to permutation invariance of the transformer input. However, 3D point clouds already have position information in them, which leads to deviance in the design of 3D transformers. Point transformer [56] proposes a point transformer layer in the PointNet structure, where the position embedding in the transformer is the pairwise point distances. 3DETR [27] and [23] use a DETR-style transformer decoder in the point cloud except that the query embedding in the decoder is sampled from Farthest Point Sampling (FPS) and learned through classification. Voxel Transformer [25] introduces a voxel transformer layer to replace the sparse convolution layer in the voxel-based point cloud backbone network. SST [8] uses a Single-stride Sparse Transformer as the backbone network to prevent information loss in downsampling of the previous 3D object detector. CT3D [33] uses a transformer to learn a refinement of the initial prediction from local points. In contrast to the above methods, our CenterFormer tailors the DETR to work on LiDAR point clouds with lower memory usage and faster convergence. Moreover, CenterFormer can learn both object-level self-attentions and local cross-attentions without requiring a first-stage bounding box prediction.

3 Method

3.1 Preliminaries

Center-Based 3D Object Detection is motivated by the recent anchor-free image-domain object detection methods [17,19]. It detects each object as a center keypoint by predicting a heatmap on the BEV feature map. Given the output of a common voxel point cloud feature encoder $M \in R^{h*w*c}$, where h and w are the BEV map size and c is the feature dimension, center-based LiDAR object detection predicts both a center heatmap $H \in R^{h*w*l}$ and the box regression $B \in R^{h*w*8}$ through two separated heads. Center heatmap H has l channels, one for each object class. In training, the ground truth is generated from the Gaussian heatmap of the annotated box center. Box regression B contains 8 object properties: the grid offset from the predicted grid center to the real box center, the height of the object, the 3D size, and the yaw rotation angle. During the evaluation, it takes the class and regression predictions at the top N highest heatmap scores and uses NMS to predict the final bounding box.

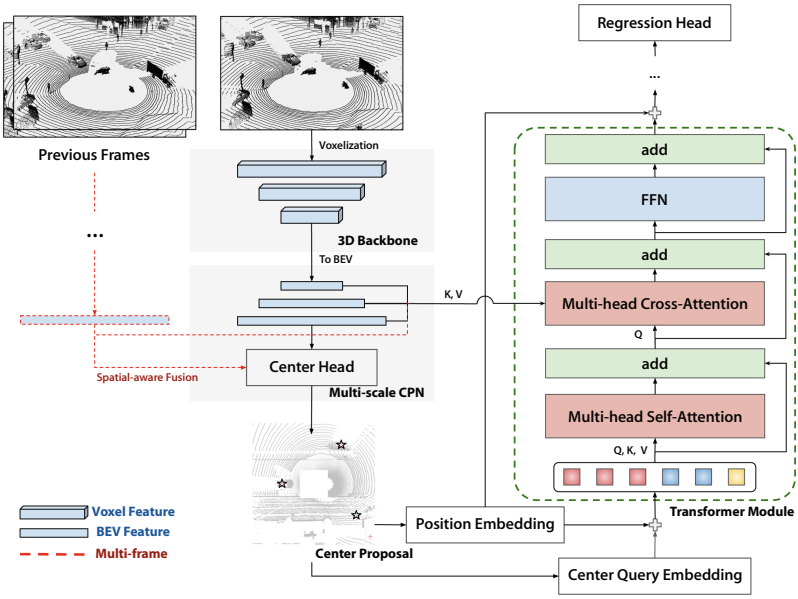

Fig. 2. The overall architecture of CenterFormer. The network consists of four parts: a voxel feature encoder that encodes the raw point cloud into a BEV feature representation, a multi-scale center proposal network (CPN), the center-based transformer decoder, and a regression head to predict the bounding box.

Transformer Decoder aggregates features from the source representation to each query based on the query-key pairwise attention. Each transformer module consists of three layers: A multi-head self-attention layer, a multi-head cross-attention layer, and a feed-forward layer. In each layer, there is also a skip connection that connects the input and the output features and layer normalization. Let f^q and f^k be the query feature and key feature. The multi-head attention can be formulated as:

$$f_i^{out} = \sum_{m=1}^{M} W_m [\sum_{j \in \Omega_j} \sigma(\frac{Q_i K_j}{\sqrt{d}}) \cdot V_j] \tag{1}$$

$$Q_i = f_i^q W_q + E_i^{pos}, K_j = f_j^k W_k + E_j^{pos}, V_j = f_j^k W_v \tag{2}$$

where i and j are the indices of query feature and source feature respectively, m is the head index, Ω_j is the set of attending key features, σ is the softmax function, d is the feature dimension, E^{pos} is the position embedding and W is the learnable weight. In the self-attention layer, the query feature and the key feature come from the same set of query feature embedding, while in the cross-attention layer, the set of key features is the source feature representation.

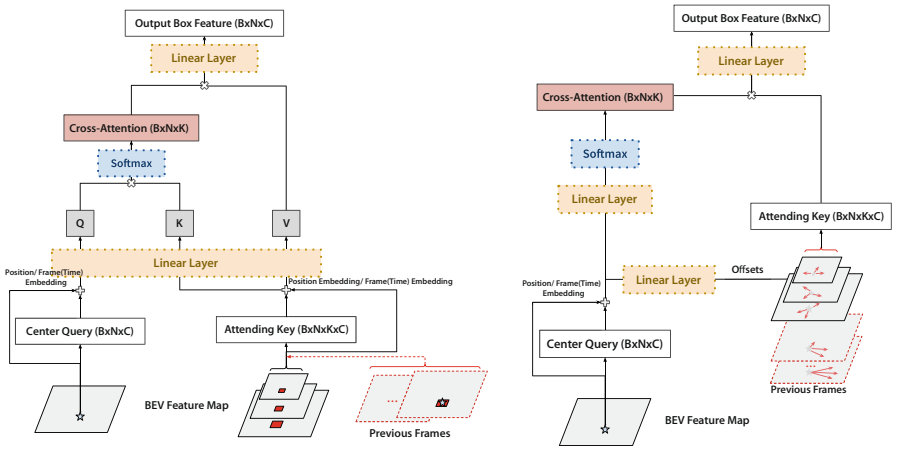

Fig. 3. Illustration of the cross-attention layer. (Left) Multi-scale cross-attention. (Right) Multi-scale deformable cross-attention.

3.2 Center Transformer

The architecture of our model is illustrated in Fig. 2. We use a standard sparse voxel-based backbone network [51] to process each point cloud into a BEV feature representation. We then encode the BEV feature into a multi-scale feature map and predict the center proposals. The proposed centers are then used as the query feature embedding in a transformer decoder to aggregate features from other centers and from multi-scale feature maps. Finally, we use a regression head to predict the bounding box at each enhanced center feature. In our multi-frame CenterFormer, the last BEV features of frames are fused together in both the center prediction stage and the cross-attention transformer.

Multi-scale Center Proposal Network. A DETR-style transformer encoder requires the feature map to be compressed into a small size so that the computation cost is acceptable. This makes the network lose fine-grained features that are crucial for the detection of small objects, which typically occupy $< 1\%$ of the space in the BEV map. Therefore, we propose a multi-scale center proposal network (CPN) to replace the transformer encoder for the BEV feature. In order to prepare a multi-scale feature map, we use a feature pyramid network to process the BEV feature representation into three different scales. At the end of each scale, we add a convolutional block attention module (CBAM) [43] to enhance the feature via channel-wise and spatial attention.

We use a center head on the highest scale feature map $\mathcal{C}$ to predict an l-channel heatmap of object centers. Each channel contains the heatmap score of one class. The location of the top N heatmap scores will be taken out as the center proposals. We used $N = 500$ in our experiments empirically.

Multi-scale Center Transformer Decoder. We extract the features at the proposed center locations as the query embedding for the transformer decoder. We use a linear layer to encode the location of the centers into a position embedding. Traditional DETR decoder initializes the query with a learnable parameter. Consequently, the attention weights acquired in the decoder are almost the same among all features. By using the center feature as the initial query embedding, we can guide the training to focus on the feature that contains meaningful object information. We use the same self-attention layer in the vanilla Transformer decoder to learn contextual attention between objects. The complexity of computing the cross-attention of a center query to all multi-scale BEV features is $O(\sum_{s=1}^{S} H_s W_s K)$. Since the BEV map resolution needs to be relatively large to maintain the fine-grained features for small objects, it is impractical to use all BEV features as the attending keypoints. Alternatively, we confine the attending keypoints to a small 3×3 window near the center location at each scale, as illustrated in Fig. 3. The complexity of this cross-attention is $O(9SK)$, which is more efficient than the normal implementation. Because of multi-scale features, we are able to capture a wide range of features around proposed centers. The Multi-scale cross-attention can be formulated as:

$$\text{MSCA}(p) = \sum_{m=1}^{M} W_m [\sum_{s=1}^{S} \sum_{j \in \Omega_j} \sigma(\frac{Q_i K_j^s}{\sqrt{d}}) \cdot V_j^s], \tag{3}$$

where p denotes the center proposal, Ω_j here is the window around the center, and s is the index of the scale. The feed-forward layer is also kept unchanged.

Multi-scale Deformable Cross-Attention Layer. Inspired by [62], we also used a deformable cross-attention layer to sample the attending keypoints automatically. Figure 3 shows the structure of the deformable cross-attention layer. Compared to the normal multi-head cross-attention layer, deformable cross-attention uses a linear layer to learn 2D offsets Δp of the reference center location p at all heads and scales. The feature at $p + \Delta p$ will be extracted as the cross-attention attending feature through bilinear sampling. We use a linear layer to directly learn the attention scores from the query embedding. Features from multiple scales are aggregated together to form the cross-attention layer output:

$$\text{MSDCA}(p) = \sum_{m=1}^{M} W_m [\sum_{s=1}^{S} \sum_{k=1}^{K} \sigma(W_{msk} \mathcal{C}(p)) x^s (p + \Delta p_{msk})], \tag{4}$$

where x^s is the multi-scale BEV feature, $\mathcal{C}(p)$ is the center feature, and $\sigma(W_{msk}$ $\mathcal{C}(p))$ is the attention weight. We used $K = 15$ in our experiments.

3.3 Multi-frame CenterFormer

Multi-frame is commonly used in 3D detection to improve performance. Current CNN-based detectors cannot effectively fuse features from a fast-moving object,

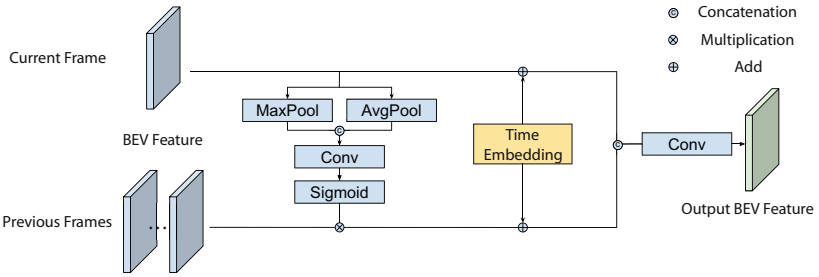

Fig. 4. The network structure of spatial-aware fusion. To focus on the current centers, we use current BEV feature as the reference to learn attention.

while the transformer structure is more suitable for the fusion due to the attention mechanism. To further explore the potential of CenterFormer, we propose a multi-frame feature fusing method using the cross-attention transformer. As shown in Fig. 2, we process each frame individually using the same backbone network. The last BEV feature of the previous frames is transformed to current coordinates and fused with the current BEV feature in both the center head and cross-attention layer.

Due to object movements, the center of an object may shift in different frames. Since we only need to predict the center in the current frame, we use a spatial-aware fusion in the center head to alleviate the misalignment error. As shown in Fig. 4, the spatial-aware module uses a similar spatial attention layer as CBAM [43] to calculate pixel-wise attention based on the current BEV feature. We concatenate the current BEV feature and weighted previous BEV feature and use an additional convolution layer to fuse them together. We also add the time embedding to the BEV features based on their relative time. Finally, we feed the output fused features to the center head to predict the center candidates.

In the cross-attention layer, we use the location of the center proposal to find the corresponding features in the aligned previous frames. The extracted features will be added to the attending keys. Since our normal cross-attention design uses features in a small window close to the center location, it has limited learnability if the object was out of the window area due to fast movement. Meanwhile, our deformable cross-attention is able to model any level of movement, and is more suitable for the long time-range case. Because our multi-frame model only needs the final BEV feature of the previous frame, it is easy to be deployed to the online prediction by saving the BEV feature in a memory bank.

3.4 Loss Functions

Besides the general classification and regression loss functions, we add two additional loss functions to better account for the center-based object detection. First, Inspired by the design in CIA-SSD [45], we move the IoU-aware confidence rectification module from the second stage of other methods to the regression

head. More specifically, we predict an IoU score iou for each bounding box proposal, which is supervised with the highest IoU between the prediction and all ground truth annotations in a smooth L1 loss. During the evaluation, we rectify the confidence score with the predicted IoU score using $\alpha' = \alpha * iou^{\beta}$, where α is the confidence score and β is a hyperparameter controlling the degree of rectification. Second, similar to [13,55], we also added a corner heatmap head alongside the center heatmap head as auxiliary supervision. For each box, we generate the corner heatmap of four bounding box edge centers and the object center using the same methods to draw the center heatmap except that the Gaussian radius is half size. During training, we supervise the corner prediction with an MSE loss on the region where the ground truth heatmap score is above 0.

The final loss used in our model is the weighted combination of the following four parts: $\mathcal{L} = w_{hm}\mathcal{L}_{hm} + w_{reg}\mathcal{L}_{reg} + w_{iou}\mathcal{L}_{iou} + w_{cor}\mathcal{L}_{cor}$. We use focal loss [21] and L1 loss for the heatmap classification and box regression. The weights of heatmap classification loss, box regression loss, IoU rectification loss, and corner classification loss are [1, 2, 1, 1], respectively, in our experiment.

4 Experiments

We present our experimental results on two large-scale LiDAR object detection benchmarks: Waymo Open Dataset [38] and nuScenes dataset [3]. Due to page limitation, the experimental results on the nuScenes dataset, more analysis as well as the details on the choice of network parameters, and additional visualizations are included in the supplementary material.

4.1 Dataset

Waymo Open Dataset (WOD) is a large-scale LiDAR point cloud dataset for the autonomous driving environment. It contains 700, 150, and 150 sequences for training, validation, and testing, respectively. Each sequence is 20-second long, captured by a 10 FPS LiDAR sensor with 64 lines in 360°. WOD provides bounding box annotations for three classes: vehicles, pedestrians, and cyclists. The evaluation metrics used in WOD are mean average precision (mAP) and mAP weighted by heading accuracy (mAPH). The evaluation is split into two levels of difficulty, where LEVEL_1 denotes that there are at least 5 points on the object and LEVEL_2 denotes that there is at least 1 point on the object. We set the range of the 3D voxel space as $[-75.2\text{m}, 75.2\text{m}]$ for the X and Y axes, and $[-2\text{m}, 4\text{m}]$ for the Z axis. The size of each voxel is set to $(0.1\text{m}, 0.1\text{m}, 0.15\text{m})$.

4.2 Implementation Details

We follow the same VoxelNet backbone network design as [34,51,60]. In our center proposal network, we process the output of the BEV feature into three

Table 1. The detection results on WOD validation set. †: Deformable CenterFormer. ‡ Reported by PVRCNN++.

Method	Frame	Vehicle L1 (mAP/APH)	Vehicle L2 (mAP/APH)	Pedestrain L1 (mAP/APH)	Pedestrain L2 (mAP/APH)	Cyclist L1 (mAP/APH)	Cyclist L2 (mAP/APH)	Mean L2 mAPH
StarNet [28]	1	55.1/54.6	48.7/48.3	68.3/60.9	59.3/52.8	–/–	–/–	–
SECOND‡ [47]	1	72.3/71.7	63.9/63.3	68.7/58.2	60.7/51.3	60.6/59.3	58.3/57.1	57.2
LiDAR R-CNN [20]	1	73.5/73.0	64.7/64.2	71.2/58.7	63.1/51.7	68.6/66.9	66.1/64.4	60.1
Part-A^2 [37]	1	77.1/76.5	68.5/68.0	75.2/66.9	66.2/58.6	68.6/67.4	66.1/64.9	63.8
3D-MAN [48]	16	74.5/74.0	67.6/67.2	71.7/67.7	62.6/59.9	–/–	–/–	–
PVRCNN++ [35]	1	78.8/78.2	70.3/69.7	76.7/67.2	68.5/59.7	69.0/67.6	66.5/65.2	64.9
CenterPoint [51]	1	–/–	–/67.9	–/–	–/65.6	–/–	–/68.6	67.4
CenterPoint [51]	2	–/–	–/69.7	–/–	–/70.3	–/–	–/70.9	70.3
CenterFormer	1	75.0/74.4	69.9/69.4	78.0/72.4	73.1/67.7	73.8/72.7	71.3/70.2	69.1
CenterFormer†	1	75.2/74.7	70.2/69.7	78.6/73.0	73.6/68.3	72.3/71.3	69.8/68.8	69.0
CenterFormer	2	77.1/76.6	72.2/71.7	80.9/77.6	76.2/73.0	76.0/75.1	73.6/72.7	72.5
CenterFormer†	2	77.0/76.5	72.1/71.6	81.4/78.0	76.7/73.4	**76.6/75.7**	**74.2/73.3**	72.8
CenterFormer†	4	78.1/77.6	73.4/72.9	81.7/78.6	77.2/74.2	75.6/74.8	73.4/72.6	73.2
CenterFormer†	8	**78.8/78.3**	**74.3/73.8**	**82.1/79.3**	**77.8/75.0**	75.2/74.4	73.2/72.3	**73.7**

scales through one upsample layer and one downsample layer. We set the transformer layer/head numbers to 3/4 when using the normal cross-attention, and to 2/6 when using the deformable cross-attention. During evaluation, we use the NMS IoU threshold of $[0.8, 0.55, 0.55]$ and $\beta = [1, 1, 4]$ for vehicle, pedestrian and cyclist. For our 8 frames model, we use $\beta = [1, 1, 1]$ to get a better result for the cyclist. We also increase the center proposal number N to 1000 in evaluation. We trained our model using AdamW optimizer with the one-cycle policy. We trained the single-frame and multi-frame models on 8 Nvidia A100 GPUs with batch sizes 32 and 16. Due to the memory limitation, 4 frames and 8 frames are first split into two 2 frames and 4 frames mini-batch. The points from frames in each mini-batch will be first concatenated together. Hence our multi-frame model only needs to fuse two BEV features together. We apply the object copy & paste augmentation during training with the same fade strategy in [41]. More details are included in the supplementary material.

4.3 Object Detection Results

In Table 1, we show the results on the validation set. Anchor-free center-based method CenterPoint achieves better performance than the anchor-based methods. PVRCNN++ is the best anchor-based method so far, yet shows weak performance on small objects. This demonstrates the limitation of using manually designed anchors for the detection of objects that have large size variations. Our single frame model outperforms CenterPoint by 1.7%. Using a multi-frame model can significantly increase the mAPH performance. Our 2/4/8 frames model reaches the mAPH of 72.8%, 73.2% and 73.7%, respectively, becoming the new state-of-the-art. The pedestrian class benefits most from the multi-frame since the pedestrian point cloud suffers most from the occlusion and noise, as well as its small size. The overall better performance verifies the effectiveness of our proposed transformer model.

Table 2. The single-model detection result on WOD testing set.

Method	Vehicle L1 (mAP/APH)	Vehicle L2 (mAP/APH)	Pedestrain L1 (mAP/APH)	Pedestrain L2 (mAP/APH)	Cyclist L1 (mAP/APH)	Cyclist L2 (mAP/APH)	Mean L2 mAPH
StarNet [28]	61.5/61.0	54.9/54.5	67.8/59.9	61.1/54.0	–/–	–/–	–
PPBA [5]	64.6/64.1	56.2/55.8	69.7/61.7	63.0/55.8	–/–	–/–	–
M3DETR [12]	77.7/77.1	70.5/70.0	68.2/58.5	60.6/52.0	67.3/65.7	65.3/63.8	61.9
3D-MAN [48]	78.7/78.2	70.4/70.0	70.0/66.0	64.0/60.3	–/–	–/–	–
RSN [1]	80.7/80.3	71.9/71.6	78.9/75.6	70.7/67.8	–/–	–/–	–
PVRCNN++ [35]	81.1/80.6	73.7/73.2	80.3/76.3	74.0/70.2	75.1/73.8	72.4/71.2	71.5
CenterPoint [51]	81.1/80.6	73.4/73.0	80.5/77.3	74.6/71.5	74.6/73.7	72.2/71.3	71.9
SST_3f [8]	81.0/80.6	73.1/72.7	83.3/79.7	76.9/73.5	75.7/74.6	73.2/72.2	72.8
AFDetV2 [13]	81.7/81.2	74.3/73.9	81.3/78.1	75.5/72.4	**76.4/75.4**	**74.1/73.0**	73.1
CenterFormer	**84.7/84.4**	**78.1/77.7**	**84.6/81.8**	**79.4/76.6**	75.5/74.5	73.3/72.4	**75.6**

Table 3. The comparison of single-frame CenterFormer (trained on vehicle only) with other methods on WOD validation set for vehicle class.

	Method	Ref	L2 mAP	L2 mAPH
CNN	RSN [1]	CVPR 2021	66.0	65.5
	Voxel R-CNN [25]	AAAI 2021	66.6	–
	Pyramid R-CNN [24]	ICCV 2021	67.2	66.7
	LiDAR R-CNN [20]	CVPR 2021	68.3	67.9
	CenterPoint [51]	CVPR 2021	–	67.9
	AFDetV2 [13]	AAAI 2022	69.7	69.2
	BtcDet [46]	AAAI 2022	70.1	69.6
	PVRCNN++ [35]	arXiv 2021	70.3	69.7
Transformer	BoxeR-3D [29]	CVPR 2022	63.9	63.7
	Voxel transformer [25]	ICCV 2021	65.9	65.3
	M3DETR [12]	WACV 2022	66.6	66.0
	3D-MAN [48]	CVPR 2021	67.6	67.1
	SST_1f [8]	CVPR 2022	68.0	67.6
	CT3D [33]	ICCV 2021	69.0	–
	CenterFormer	ECCV 2022	**70.5**	**70.0**

In Table 2, we show the single model results on the test set. The prediction result is submitted to the online server for evaluation. Our method outperforms all the previous methods by a large margin. The result on vehicle and pedestrian classes have significant improvements (+3.8% and +3.1% on L2 mAPH) as a result of the long-range contextual information learning of the transformer.

To fairly compare with more recent methods [6,24,25,33] that train the model and report the result on only the vehicle class, we train the single frame CenterFormer only on Vehicle class too. We show the results in Table 3. As shown in the table, even with the simplest design, CenterFormer outperforms both the recent transformer-based methods and CNN-based baselines.

Table 4. The ablation of the LEVEL_2 mAPH result improvement of each component on the validation set using single frame. SA, CA and DCA denote the self-attention layer, cross-attention layer and deformable cross-attention layer. IoU and Corner denote IoU rectification and Corner auxiliary supervision. Fade denotes the fade augmentation strategy.

CPN	SA	CA	Corner	DCA	IoU	Fade	Vehicle	Pedstrain	Cyclist	Mean
							66.4	63.4	67.8	65.9
✓							68.5	62.7	67.0	66.1
✓	✓						68.7	64.5	67.3	66.8
✓		✓					68.7	64.2	66.7	66.5
✓	✓	✓					69.3	64.8	66.8	67.0
✓	✓	✓	✓				69.5	65.3	66.4	67.1
✓	✓	✓	✓		✓		69.5	66.9	69.7	68.7
✓	✓	✓	✓		✓	✓	69.4	67.7	70.2	69.1
✓	✓		✓	✓			69.3	65.1	67.5	67.3
✓	✓		✓	✓	✓		69.2	66.7	69.1	68.3
✓	✓		✓	✓	✓	✓	69.7	68.3	68.8	69.0

4.4 Ablation Study

In Table 4, we investigate the effect of each added component in our method on a single frame. We use the previous center-based method as the baseline. After changing the RPN to multi-scale CPN and separating the detection into the center proposal and box regression, our method reaches a similar performance despite we flatten the regression head to 1D. The transformer self-attention layer and cross-attention layer both can improve the results, and when used together, the result reaches 67.0%. This indicates that the self-attention layer and the cross-attention layer learn features separately. Corner auxiliary supervision can additionally improve the result by 0.1%. On the other hand, deformable cross-attention achieves a better result of 67.3%. When trained with IoU rectification, the results get a significant boost to 68.7% and 68.3% for the models using cross-attention and deformable cross-attention. The fade augmentation strategy can further improve the result by 0.4% and 0.7%. This is because the model can adjust to the real data distribution at the end of the training.

4.5 Analysis

Comparison with Deformable DETR. Deformable DETR [62] aims to speed up the learning speed and reduce the computation cost in the DETR structure. Compared to Deformable DETR, our method has three major differences. First, we completely remove the transformer encoder to enable a larger encoded feature map. Second, we use the center feature rather than the learnable parameter as the query embedding for the transformer decoder. Experiment shows that

using the center feature as the initial query embedding outperforms the parametric embedding by 1.5% mAPH. This is because the center feature already contains object-level information, which makes it easier to learn pairwise attentions. Third, we use a similar training strategy as [51] rather than the end-to-end set matching training strategy. The set matching training is known for being hard to converge. Since we already have an initial center proposal, we can limit the network to learn only when the proposal is close to the ground truth annotation to speed up the training. Experiment shows that if we use the same set matching training strategy in DETR, the mAPH result is 46.3%, which is more than 20% lower than our current training method.

Visualization of the Learned Attention. The visualizations of the learned self- and cross-attention are illustrated in Fig. 5. The self-attention learning is mainly focusing on the feature of the same class or nearby objects that have the same attributes. For instance, the vehicles on the same line or same parking area will have higher attention weight. The offsets learned by the deformable attention layer vary among different scales. The offsets in two lower scales mostly lead to the keypoints inside or around the object, whereas the offsets in the higher scale can sample far-range features.

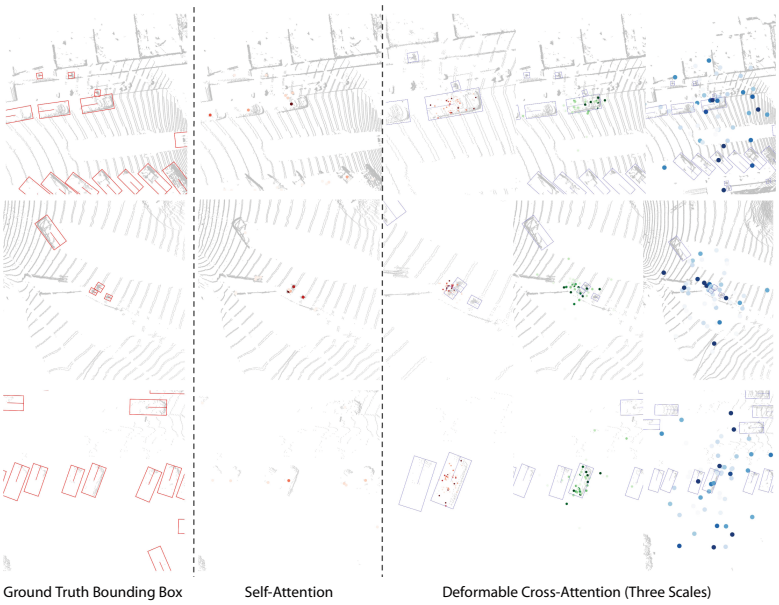

Fig. 5. The visualization of the learned self- and cross-attention weight. The lightness or darkness of the color represents the value of attention weight. The red box denotes the ground truth box and blue box denotes the predicted box. In cross-attention, the sampled keypoints are drawn with red, green and blue for the scale from low to high. Best viewed in color. (Color figure online)

5 Conclusion

In this paper, we propose a novel center-based transformer for 3D object detection. Our method provides a solution to improve the anchor-free 3D object detection network through object-level attention learning. Compared to the DETR-style transformer networks, we use the center feature as the initial query embedding in the transformer decoder to speed up the convergence. We also avoid high computational complexity by focusing the cross-attention learning of each query in a small multi-scale window or a deformable region. Results show that the proposed method outperforms the strong baseline in the Waymo Open Dataset, and reaches state-of-the-art performance when extended to multi-frame. We hope our design will inspire more future work in query-based transformers for LiDAR point cloud analysis.

Acknowledgements. We thank Yufei Xie for his help with refactoring the code for release.

References

1. Bewley, A., Sun, P., Mensink, T., Anguelov, D., Sminchisescu, C.: Range conditioned dilated convolutions for scale invariant 3d object detection. In: CoRL (2021)
2. Bowen, C., Alexander, G.S., Alexander, K.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
3. Caesar, H., et al.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
5. Cheng, S., et al.: Improving 3d object detection through progressive population based augmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12366, pp. 279–294. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58589-1_17
6. Deng, J., Shi, S., Li, P., Zhou, W., Zhang, Y., Li, H.: Voxel r-cnn: Towards high performance voxel-based 3d object detection. In: AAAI (2021)
7. Dosovitskiy, A., et al.: An image is worth 16×16 words: Transformers for image recognition at scale. In: ICLR (2021)
8. Fan, L., et al.: Embracing single stride 3d object detector with sparse transformer. In: CVPR (2022)
9. Fan, L., Xiong, X., Wang, F., Wang, N., Zhang, Z.: Rangedet: In defense of range view for lidar-based 3d object detection. In: ICCV (2021)
10. Ge, R., et al.: Afdet: Anchor free one stage 3d object detection. In: CVPRW (2020)
11. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)

12. Guan, T., et al.: M3detr: Multi-representation, multi-scale, mutual-relation 3d object detection with transformers. In: WACV (2022)
13. Hu, Y., et al.: Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In: AAAI (2022)
14. Huang, R., et al.: An LSTM approach to temporal 3d object detection in lidar point clouds. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12363, pp. 266–282. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58523-5_16
15. Jianwei, Y., et al.: Focal self-attention for local-global interactions in vision transformers. In: NeurIPS (2021)
16. Jifeng, D., et al.: Deformable convolutional networks. In: ICCV (2017)
17. Kaiwen, D., Song, B., Lingxi, X., Honggang, Q., Qingming, H., Qi, T.: Centernet: Keypoint triplets for object detection. In: ICCV (2019)
18. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: PointPillars: Fast encoders for object detection from point clouds. In: CVPR (2019)
19. Law, H., Deng, J.: CornerNet: Detecting objects as paired keypoints. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 765–781. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_45
20. Li, Z., Wang, F., Wang, N.: Lidar r-cnn: An efficient and universal 3d object detector. In: CVPR (2021)
21. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: CVPR (2017)
22. Liu, S., et al.: Dab-detr: Dynamic anchor boxes are better queries for detr. In: ICLR (2021)
23. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3d object detection via transformers. In: ICCV (2021)
24. Mao, J., Niu, M., Bai, H., Liang, X., Xu, H., Xu, C.: Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In: CVPR (2021)
25. Mao, J., et al.: Voxel transformer for 3d object detection. In: CVPR (2021)
26. Meng, D., et al.: Conditional detr for fast training convergence. In: ICCV (2021)
27. Misra, I., Girdhar, R., Joulin, A.: An end-to-end transformer model for 3d object detection. In: CVPR (2021)
28. Ngiam, J., et al.: Starnet: Targeted computation for object detection in point clouds. arXiv (2019)
29. Nguyen, D.K., Ju, J., Booji, O., Oswald, M.R., Snoek, C.G.: Boxer: Box-attention for 2d and 3d transformers. In: CVPR (2022)
30. Noh, J., Lee, S., Ham, B.: Hvpr: Hybrid voxel-point representation for single-stage 3d object detection. In: CVPR (2021)
31. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
32. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: NeurIPS (2017)
33. Sheng, H., et al.: Improving 3d object detection with channel-wise transformer. In: ICCV (2021)
34. Shi, S., et al.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR (2020)
35. Shi, S., et al.: Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. arXiv (2021)
36. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object progposal generation and detection from point cloud. In: CVPR (2019)

37. Shi, S., Wang, Z., Shi, J., Wang, X., Li, H.: From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. In: TPAMI (2020)
38. Sun, P., ., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: CVPR (2020)
39. Sun, P., et al.: Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In: CVPR (2021)
40. Vaswani, A., et al.: Attention is all you need. In: NeurIPS (2017)
41. Wang, C., Ma, C., Zhu, M., Yang, X.: Pointaugmenting: Cross-modal augmentation for 3d object detection. In: CVPR (2021)
42. Wang, H., Zhu, Y., Adam, H., Yuille, A., Chen, L.C.: Max-deeplab: End-to-end panoptic segmentation with mask transformers. In: CVPR (2021)
43. Woo, S., Park, J., Lee, J.-Y., Kweon, I.S.: CBAM: Convolutional block attention module. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11211, pp. 3–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01234-2_1
44. Wu, B., Wan, A., Yue, X., Keutzer, K.: Squeezeseg: Convolutional neural nets with recurrent crf for real-time road-object segmentation from 3d lidar point cloud. In: ICRA (2018)
45. Wu, Z., Weiliang, T., Sijin, C., Li, J., Chi-Wing, F.: Cia-ssd: Confident iou-aware single-stage object detector from point cloud. In: AAAI (2021)
46. Xu, Q., Zhong, Y., Neumann, U.: Behind the curtain: Learning occluded shapes for 3d object detection. In: AAAI (2022)
47. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. In: Sensors (2018)
48. Yang, Z., Zhou, Y., Chen, Z., Ngiam, J.: 3d-man: 3d multi-frame attention network for object detection. In: CVPR (2021)
49. Ye, M., Xu, S., Cao, T.: Hvnet: Hybrid voxel network for lidar based 3d object detection. In: CVPR (2020)
50. Yin, J., Shen, J., Guan, C., Zhou, D., Yang, R.: Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In: CVPR (2020)
51. Yin, T., Zhou, X., Krähenbühl, P.: Center-based 3d object detection and tracking. In: CVPR (2021)
52. Ze, L., et al.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
53. Zhang, H., et al.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv (2022)
54. Zhang, Y., et al.: Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In: CVPR (2020)
55. Zhang, Z., Sun, B., Yang, H., Huang, Q.: H3DNet: 3D object detection using hybrid geometric primitives. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12357, pp. 311–329. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58610-2_19
56. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: CVPR (2021)
57. Zheng, S., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR (2021)
58. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR (2018)
59. Zhou, Z., Zhang, Y., Foroosh, H.: Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In: CVPR (2021)

60. Zhu, B., Jiang, Z., Zhou, X., Li, Z., Yu, G.: Class-balanced grouping and sampling for point cloud 3d object detection. arXiv (2019)
61. Zhu, X., et al.: Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In: CVPR (2021)
62. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. In: ICLR (2020)