Clusterformer: Cluster-based Transformer for 3D Object Detection in Point Clouds

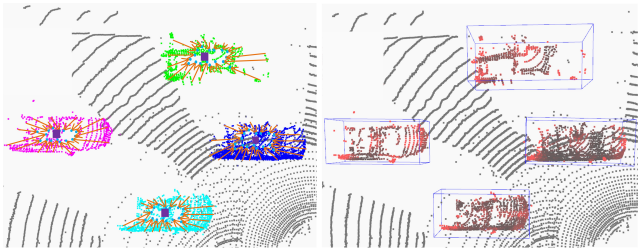
Yu Pei† Xian Zhao† Hao Li Jingyuan Ma Jingwei Zhang Shiliang Pu*
HikVision Research Institute
{peiyu, zhaoxian, lihao85, majingyuan, zhangjingwei6, pushiliang.hri}@hikvision.com

Abstract

Attributed to the unstructured and sparse nature of point clouds, the transformer shows greater potential in point clouds data processing. However, the recent query-based 3D detectors usually project the features acquired from a sparse backbone into the structured and compact Bird's Eye View(BEV) plane before adopting the transformer, which destroys the sparsity of features, introducing empty tokens and additional resource consumption for the transformer. To this end, in this paper, we propose a novel query-based 3D detector called Clusterformer, our Clusterformer regards each object as a cluster of 3D space which mainly consists of the non-empty voxels belonging to the same object, and leverages the cluster to conduct the transformer decoder to generate the proposals from the sparse voxel features directly. Such cluster-based transformer structure can effectively improve the performance and convergence speed of query-based detectors by making use of the object prior information contained in the clusters. Additionally, we introduce a Query2Key strategy to enhance the key and value features with the object-level information iteratively in our cluster-based transformer structure. Experimental results show that the proposed Clusterformer outperforms the previous query-based detectors with a lower latency and memory usage, which achieves state-of-the-art performance on the Waymo Open Datasets and KITTI Datasets.

1. Introduction

LiDAR 3D object detection is a fundamental task in various application fields such as autonomous driving systems and robotics navigation, which has attained wide attention in recent years [27, 39, 34, 26, 14, 41]. Unlike the images with a regular structure in the 2D tasks, the point clouds from LiDAR sensors are unstructured and sparse, making it challenging to directly adopt the convolution neural network (CNN). To tackle this challenge, some detectors [27, 21, 35, 38] extract geometric features directly from point clouds based on the PointNet [22] or PointNet++ [23], while other approaches [34, 26, 14, 25] voxelize the raw point clouds into discrete grids then utilize standard CNN.



(a) (b)

Figure 1. (a): The illustration of our cluster-based query initialization and interaction range. We group the foreground non-empty voxels into different clusters(shown in different colors) based on center voting, and leverage the cluster center(represented by purple rectangle) to initialize the queries. Additionally, we limit the interaction range in the same cluster to make the query only focus on an interest region. (b): Visualization of the attention map in proposed intra-cluster cross-attention(the red color denotes higher attention weight), which shows that the queries can adaptively aggregate crucial voxel features(such as the roof and corner area of a vehicle) in each cluster by the intra-cluster cross-attention.

Witnessing the remarkable research achievements in vision tasks [5, 17, 2, 42], the transformer has also drawn growing attention in point clouds processing recently [18, 7, 12, 41, 1, 6, 20], and shows greater potential compared to CNN and PointNet since it can directly process sparse variable-length point clouds and holds powerful ability in capturing contextual dependencies among points. Among them, [41, 1, 6, 20] mainly leverage the transformer decoder to generate the detections from a set of predefined queries in an end-to-end manner which are also called query-based detectors. Following the currently mainstream grid-based detectors, these query-based detectors also project the sparse features into the BEV plane before adopting the transformer. Although we can obtain meaningful initial queries based on the learned BEV

†Equal contribution, *Corresponding author.

center heatmap [41, 1] which is significant for the query-based detectors, we believe such projection for transformer structure still has the following drawbacks: 1) since the Transformer structure holds powerful ability in processing variable-length sequential data, projecting the sparse and sequential point clouds into the regular and compact BEV plane for transformer structure is unnecessary; 2) the BEV projection destroys the natural sparsity of the features from a sparse convolution backbone, introducing empty "tokens" for the transformer structure and additional resource consumption; Unlike these query-based detectors for outdoor scenes, the 3DETR [19] adopts the transformer decoder on the raw point clouds directly. However the initial queries in 3DETR are acquired from the 3D space by the Farthest Point Sampling (FPS), such query initialization can't achieve satisfactory performance for outdoor scenes due to the fact that point clouds in outdoor scenes are sparse and unevenly distributed compared to indoor scenes.

Based on the above observations, in this paper, we seek to design an effective query-based detector for outdoor scenes to leverage the ability of transformer structures in processing sparse sequence point clouds. To achieve this goal, one crucial factor is to acquire meaningful query point such as object centers [41, 1] from the sparse 3D space. Inspired by the concept of cluster in the 3D panoptic segmentation [15, 40], we propose a novel query-based 3D detector called *Clusterformer*. Our Clusterformer regards each object as a cluster of 3D space which mainly consists of the non-empty voxels belonging to the same object and first obtains different clusters based on center voting. Since the cluster centers are closed to the object centers after center voting, we encode the cluster centers as the initial queries. In this way, our Clusterformer can acquire the initial queries which contain accurate location information of candidates from 3D space in outdoor scenes.

The other significant factor for the query-based detectors is a reasonable interaction range for queries that directly influence the convergence speed [42, 20]. In our Clusterformer, we design an intra-cluster cross-attention to decode the queries into final detections, the cluster-based query initialization allows us to perform intra-cluster interaction between the queries and the grouped voxel features, which can keep the queries only focus on an interest region. We also introduce a Query2Key strategy to enhance the key and value features in the intra-cluster cross-attention with object-level information, which is contained in the query features. Since we have dropped abundant background and empty voxels in the cluster generating process, the cluster-based transformer structure can work in an efficient way.

Extensive experiments are conducted on the Waymo Open Dataset [30] and KITTI dataset [10] to show the state-of-the-art performance of the Clusterformer, the contribution of our Clusterformer can be summarized as follows:

1) A cluster-based transformer called Clusterformer is proposed for outdoor 3D object detection, which applies the transformer on the sparse voxel features to generate proposals directly;

2) We leverage the clusters to acquire the initial queries and perform intra-cluster interaction in our Clusterformer to improve the performance and convergence speed by making use of the object prior information contained in clusters, we also conduct extensive experiments to explore how query initialization strategy and interaction range affect the performance of query-based detectors;

3) A simple but effective strategy is introduced to enhance the key and value features with object-level information during the multiple transformer decoder layers;

4) The proposed Clusterformer has acquired state-of-the-art performance on the large-scale Waymo Open dataset and KITTI dataset.

2. Related work

2.1. Object detection from point clouds

3D object detection can generally be divided into point-based [38, 27, 35, 21, 38] and grid-based methods [39, 34, 14, 26] according to the data representations.

Point-based methods typically adopt PointNet [22] or its various variant networks [23, 16] to extract point-wise geometric features and generate dense predictions directly. Among them, VoteNet [21] first proposes the center voting to aggregate features around the object centers which has been widely used in subsequent point-based works [38, 9, 35]. In contrast, our work mainly focus on proper query initialization from the sparse 3D space with center voting.

Grid-based methods project the raw point clouds into structured voxels [39, 34, 26] or pillars [14, 25], so that the CNN can be adopted directly. Among them, Second [34] introduces the 3D submanifold sparse convolution [11] to extract voxel features while maintaining computational efficiency. Grid-based methods are currently the mainstream detection approach and have achieved promising detection performance. However, the performance and computational complexity are often affected by the voxelization granularity and perception range, since these methods usually project the sparse features into BEV plane.

2.2. Query-based Detectors

DETR [2] is the pioneering work to utilize transformer structure for end-to-end object detection, which regards the objects as a set of learnable queries. However, due to the randomness of the initial queries and the global interaction range, the convergence speed of DETR is extremely slow. Many follow-up works of DETR are proposed to improve the convergence speed and performance with better query initialization strategy [32, 37] and more reasonable interaction range [42, 20].

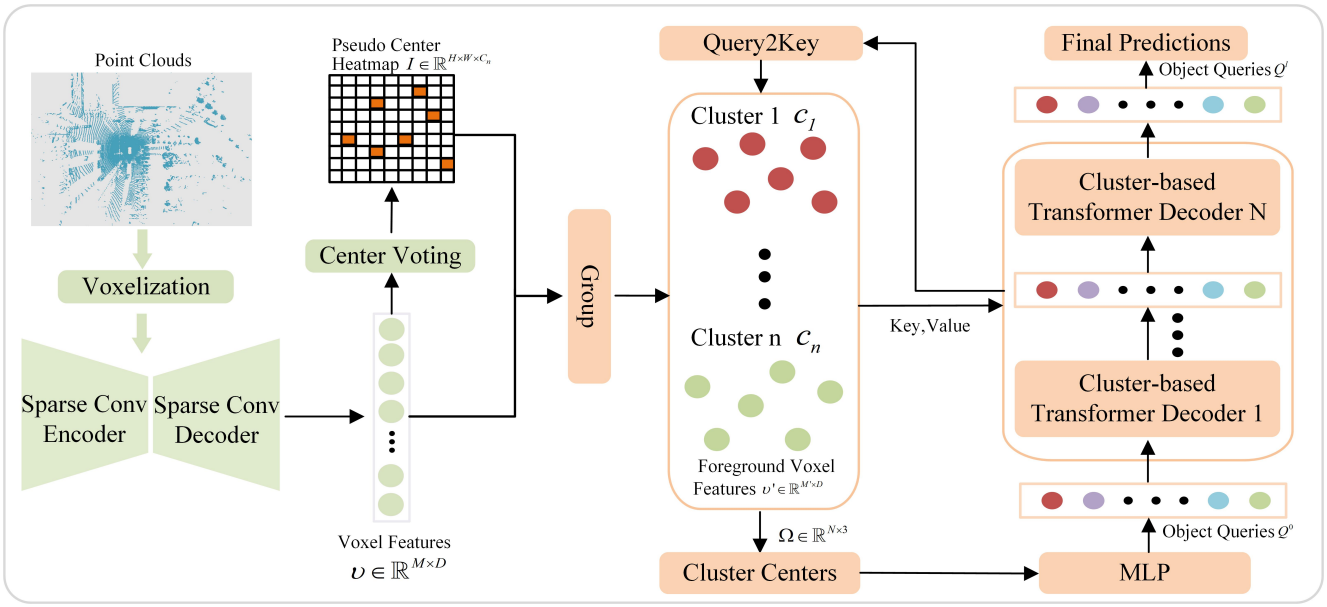


Figure 2. Overall pipeline of our Clusterformer. We first extract the sparse voxel features from a U-net like sparse backbone and group the foreground voxels into different clusters based on center voting and a pseudo center heatmap, then the cluster-based transformer structure with multiple decoder layers is introduced to get the final detections based on the grouped voxel features and a set of cluster-based queries.

3DETR [19] first introduces the DETR into indoor 3D detection task. However, the initial queries in 3DETR are acquired from 3D space by FPS which is not suitable for the outdoor scenes. Different from [19], recently query-based 3D detectors [41, 6, 1, 20] usually project the sparse features into the BEV plane before applying the transformer structure. Among them, Centerformer[41] and Transfusion [1] both utilize the BEV features to initialize the query based on a learned center heatmap. Although, such query initialization can keep the query meaningful, projecting the voxel features into the BEV plane destroys the sparsity of the features, introducing abundant empty tokens and bringing senseless interactions for the transformer structure. Different from these query-based detectors, we group the non-empty voxels into different clusters and utilize the cluster centers to initialize the query. In this way, our Clusterformer can keep the query meaningful which contains accurate location information of object candidates without projecting the voxel features into the BEV plane.

2.3. Point Clouds Panoptic Segmentation

Point clouds panoptic segmentation is generally divided into proposal-based [29] and proposal-free methods [40, 15, 33]. Among them, the proposal-free methods aim to explore cluster-based instance segmentation and drop the additional detection branch. These methods usually adopt center voting and an additional center heatmap to generate the clusters which mainly consist of not-empty voxels belonging to the same instance. In this paper, we leverage such cluster to conduct our cluster-based transformer structure.

3. Methodology

3.1. overview

The overall architecture of the proposed Clusterformer is illustrated in Fig. 2. In our Clusterformer, we first voxelize the input point clouds and utilize a 3D sparse convolution backbone to extract the voxel-wise features(See §3.2). Then, to acquire meaningful object queries from these sparse features, we group the non-empty foreground voxels into different clusters based on center voting and initialize the queries based on the cluster centers(See §3.3). After acquiring these initial queries and grouped voxel features, we further proposed a cluster-based transformer decoder to decode the query by performing intra-cluster interaction with the grouped voxel features(See §3.4). In the end, we adopt a multi-layer perceptron (MLP) to generate the detections from the refined queries with a Centerpoint [36] style box regression objective(See §3.5).

3.2. Sparse feature extraction

Given the input raw point clouds $P \in \mathbb{R}^{n \times 3}$ with n points, we first voxelize P and adopt a U-net like 3D sparse convolution backbone [28] to extract sparse voxel features $\upsilon \in \mathbb{R}^{M \times D}$, where M and D represent the number of non-empty voxels and feature dimension. Different from the previous query-based detectors [41, 1, 6, 20], we do not project the sparse voxel features into the BEV plane, but directly feed them into the transformer structure to leverage the ability of the transformer structure in processing the sparse sequence data.

3.3. Query initialization

Since the query initialization is significant for the query-based detectors [32, 37, 41, 1], we hope to find positions with explicit meaning like object centers in the sparse 3D space to initialize the queries for providing prior information of object candidates. In the proposal-free 3D panoptic segmentation methods, the sparse non-empty voxels are grouped into different clusters based on center voting, we observe that such cluster centers are closed to the object centers after center voting, which can be utilized to initialize the query. At the same time, the cluster can be adopted as a reasonable interaction range for the queries in 3D space. Based on these observations, we first group the non-empty foreground voxels into different clusters and initialize the queries from these clusters.

Cluster generating. To obtain clusters from the sparse voxel features, we first classify the non-empty voxels into different categories and predict the center offset of each foreground voxels by multi-layer perceptron (MLP). Then we shift the voxels closer to their corresponding object centers by adding the predicted offset which is called as center voting [21], and these shifted voxels can be seen as voted centers in VoteNet [21]. Instead of sampling a subset of voted centers by the FPS in VoteNet, we adopt a pseudo heatmap [15] to generate the clusters. Specifically, all the voted centers are projected onto a class-aware BEV map $I \in \mathbb{R}^{H\times W\times C_n}$, where H and W are the BEV map size and C_n represents the number of categories. The number of voted centers in each BEV grid can represent the possibility of the object center falling on the current grid. As a result, the I can be seen as a pseudo center heatmap. We select the position with a local maximum value in I as the object center to avoid generating multiple centers for one object. After acquiring the object centers on the BEV plane, we can simply allocate each shifted voxel to its closest center based on the euclidean distance to generate different clusters $\mathbf{C} = \{c_0, c_1, ...c_i, ..., c_n\}$, where i denotes the unique id of each cluster.

Query initialization from cluster centers. After acquiring the different clusters $\mathbf{C}$, we take the mean position of all the voted centers(shifted voxels) in each cluster c_i as the cluster center. Then we encode the coordinate of each cluster center $\mathbf{\Omega_i} = \{x_i, y_i, z_i\}$ by two linear layers as the initial query which can be formulated as:

$$Q_i^0 = \psi_1(\psi_2(\Omega_i)) \tag{1}$$

where the ψ_1 and ψ_2 represent the two linear layers. $Q^0 \in \mathbb{R}^{N\times D}$ represents the initial query, where N denotes the number of queries. In this way, the acquired queries can contain accurate location information of object candidates which is beneficial for the query decoding. It is worth mentioning that the acquired queries in our Clusterformer are sparse since the queries are only initialized at the possible object centers. Additionally, we do not need to manually set the number of queries in our Clusterformer since it is determined by the cluster number.

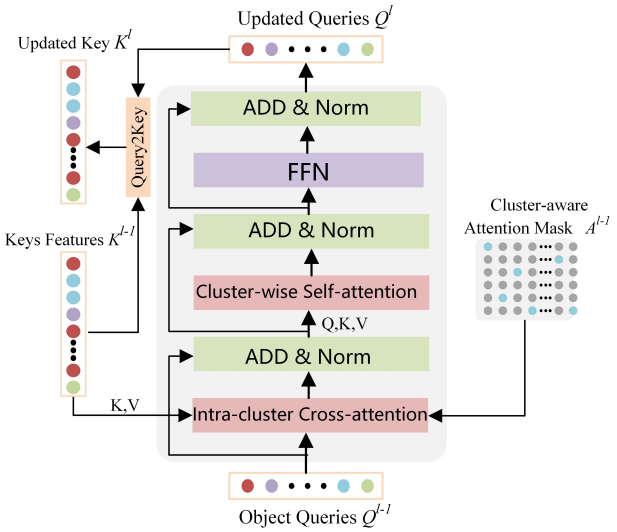


Figure 3. The architecture of our cluster-based transformer decoder, where the intra-cluster cross-attention is proposed to perform the intra-cluster interaction by a cluster-aware attention mask to decode the query. Additionally, we introduce a Query2Key strategy to enhance the key and value features with object-level information after each decoder layer.

3.4. Cluster-based Transformer Decoder

After acquiring the initial queries and the grouped non-empty foreground voxel features, we design a cluster-based transformer decoder to decode the query as shown in Fig. 3. It mainly consists of an intra-cluster cross-attention(ICCA) layer, a self-attention layer and the Query2Key strategy.

Intra-cluster cross-attention. Because the acquired clusters mainly consist of the non-empty voxels belonging to the same object, we naturally think of limiting the interaction range in the same cluster to keep the queries only focus on an interest region.

In our intra-cluster cross-attention, we introduce a cluster-aware attention mask $A^l \in \mathbb{R}^{N\times M'}$ for such intra-cluster interaction, where the l represents the decoder layers. If the i_{th} query and j_{th} key belong to a same cluster, A^l_{ij} would be set as 0 while the other would be set as $-\infty$, which is formulated as Eq. 2.

$$A^l_{i,j} = \begin{cases} 0, & id(Q^l_i) = id(K^l_j); \\ -\infty, & otherwise \end{cases} \tag{2}$$

where the $id(\bullet)$ represents the cluster id. Based on the attention mask $A^l_{i,j}$, the proposed intra-cluster cross-attention layer can be formulated as:

$$Q^l = softmax(\frac{Q^{l-1}K^{l-1}}{\sqrt{D}} + A^{l-1})V^{l-1} \tag{3}$$

We treat the features of each voxel as a token, and set the grouped non-empty voxels features v' as the key $K \in \mathbb{R}^{M' \times D}$ and value $V \in \mathbb{R}^{M' \times D}$ for the intra-cluster cross-attention layer, where M' is the number of foreground non-empty voxels and $M >> M'$. Since we have dropped the numerous background and empty voxels, the intra-cluster cross-attention layer is adopted on a smaller number of tokens with a lower resource consumption, which is significant for outdoor scene. Additionally, unlike the standard cross-attention, the key and value features in our intra-cluster cross-attention are also updated during the multiple decoders by the proposed Query2Key strategy.

Cluster-wise Self-attention. The intra-cluster cross-attention mainly focus on aggregating the information of object candidate in a cluster which lacks of access to the global range information, therefore we further introduce a self-attention layer to perform cluster-wise interaction between the different queries Q^l. Such cluster-wise interaction can provide a global-range view for object localization, additionally, the interaction between the queries can reason the pairwise relation of different candidates.

Query2Key strategy. Since the queries mainly acquire the object candidate information by the cross-attention, the representation ability of the key and value features in a cross-attention layer is essential which directly involves the detection performance. In our intra-cluster cross-attention, the features acquired from the sparse backbone are set as the key and value, which mainly contain local-range information. However, the box regression also needs object-level information for better performance. Therefore, we introduce a simple but effective strategy named Query2Key to enhance the key and value features with object-level information which is similar to the VFE[39]. Specifically, after each decoder layer, we concatenate the updated query features Q^l on the key features K^{l-1} with the same cluster id in a broadcast manner, then use a linear layer to refine the contacted features. Since we generate detections for each decoder layer based on the queries, the updated queries are considered to contain adequate object-level information, such concatenate operation can fuse the local and object-level information to enhance the key features.

Note that since the value features are kept the same with the key features, the value features are also been enhanced as the key features. These enhanced key and value features would be utilized in the next decoder layer.

3.5. Detection head and Loss functions

Detection head. We feed the refined queries into MLP to generate bounding boxes. We also follow the Deformable-DETR [42] to adopt the iterative bounding box refinement with the shared MLP, each decoder would refine the bounding boxes based on the predictions from the previous decoder layer. Additionally, based on the predicted box centers by each decoder, we can correct the cluster id for the key features and update the cluster-aware attention mask.

Loss Functions. The loss functions in our Clusterformer can be divided to cluster generating loss ℓ^c and box prediction loss ℓ^b which are formulated as:

$$\ell = \ell^c_{offset} + \ell^c_{cls} + \sum_{i=1}^{n} \ell^b_{reg} + \sum_{i=1}^{n} \ell^b_{cls} + \sum_{i=1}^{n} \ell^b_{iou} \quad (4)$$

where n represents the number of decoder layers. We adopt L1 loss for the center offset prediction and box regression. For the classification loss, we adopt focal loss for the voxel and box classification. Following [41, 1], the IoU loss is also introduced to improve the detection performance by refining the box classification confidence.

3.6. Discussion

The FSD [9] also acquires clusters by center voting [21] and generates detections based on clusters, while our Clusterformer still has two essential differences from FSD.

1). Unlike FSD, ClusterFormer is essentially a DETR-liked detector, the "clusters" are only utilized to provide object prior information for query decoding, which means that even without clusters, ClusterFormer can still complete the detection task in an end-to-end manner.

2). FSD aggregates group features by dynamic pooling which treats different points equally in a cluster, instead, our ClusterFormer proposes the ICCA to selectively aggregate the critical features, meanwhile, FSD lacks modeling the global context information, in our Clusterformer, cluster-wise self-attention allows for the interaction of features between different clusters to obtain global information.

4. Experiments

In this section, we conduct experiments on the two commonly used datasets, Waymo [30] and KITTI [10] to evaluate the proposed Clusterformer. We first introduce the two datasets and implementation details of the proposed Clusterformer, then compare our approach with the recent state-of-the-art detectors. Finally, ablation studies are conducted to explore the effectiveness of the model design details.

4.1. Waymo open dataset

The Waymo Open dataset [30] consists of 1000 sequences (around 158k samples in total), including 798, 202, and 150 for training, validation, and testing, respectively, which provides 3D bounding box annotations for three categories: vehicles, cyclists, and pedestrians. The commonly used metric 3D Mean Average Precision (mAP) is adopted for WOD evaluation. Additionally, the WOD introduces the mAPH (mAP weighted by heading accuracy) to evaluate the accuracy of the object orientation. The metric is further divided into Level 1(boxes with more than five LiDAR points) and Level 2(boxes with at least one LiDAR points).

Methods	mAP/mAPH L1	mAP/mAPH L2	Vehicle 3DAP/APH L1	Vehicle 3DAP/APH L2	Pedestrian 3DAP/APH L1	Pedestrian 3DAP/APH L2	Cyclist 3D AP/APH L1	Cyclist 3D AP/APH L2
Second [34]	67.2/63.1	61.0/57.2	72.3/71.7	63.9/63.3	68.7/58.2	60.7/51.3	60.6/59.3	58.3/57.0
Part-A2-Net [28]	73.6/70.2	66.9/63.8	77.1/76.5	68.5/68.0	75.2/66.9	66.2/58.6	68.6/67.4	66.1/64.9
CenterPoint-Voxel [36]	74.4/71.7	68.2/65.8	74.2/73.6	66.2/65.7	76.6/70.5	68.8/63.2	72.3/71.1	69.7/68.5
PV-RCNN++ [26]	78.1/75.9	71.7/69.5	79.3/78.8	**70.6/70.2**	81.3/76.3	73.2/68.0	73.7/72.7	71.2/70.2
AFDetV2 [13]	77.2/74.8	71.0/68.8	77.6/77.1	69.7/69.2	80.2/74.6	72.2/67.0	73.7/72.7	71.0/70.1
Pillarnet-34 [25]	77.3/74.6	70.9/68.4	79.0/78.5	70.9/70.4	80.5/74.0	72.2/66.1	72.2/71.2	69.7/68.6
FSD [9]	79.4/77.1	72.7/70.5	79.5/79.0	70.3/69.9	83.6/78.2	74.4/69.4	75.3/74.1	73.3/72.1
SST-Center [7]	75.5/72.3	69.2/66.2	75.1/74.6	66.6/66.1	80.0/72.1	72.3/65.0	71.4/70.2	68.8/67.6
Voxelset [12]	75.5/72.2	69.1/66.2	74.5/74.0	65.9/65.5	80.0/72.4	72.4/65.4	71.5/70.2	68.9/67.7
Votr-Ts [18]	-/-	-/-	74.9/74.2	65.9/65.2	-/-	-/-	-/-	-/-
SWFormer [31]	-/-	-/-	77.8/77.3	69.2/68.8	80.9/72.7	72.5/64.9	-/-	-/-
TransFusion-L [1]	-/-	-/64.9	-/-	-/65.1	-/-	-/63.7	-/-	-/65.9
Centerformer [41]	75.3/72.9	71.7/68.9	75.0/74.4	69.9/69.4	78.6/73.0	73.6/68.3	72.3/71.3	69.8/68.8
Clusterformer(Ours)	**81.4/79.0**	**74.6/72.3**	**79.8/79.3**	70.5/70.1	**84.4/79.0**	**75.7/70.6**	**80.0/78.7**	**77.4/76.2**
SWFormer-3f [31]	-/-	-/-	79.4/78.9	71.1/70.6	82.9/79.0	74.8/71.1	-/-	-/-
SST-3f [7]	-/-	-/-	77.0/76.6	68.5/68.1	82.4/78.0	75.1/70.9	-/-	-/-
Centerformer-4f [41]	78.5/77.0	74.7/73.2	78.1/77.6	73.4/72.9	81.7/78.6	77.2/74.2	75.6/74.8	73.4/72.6
MPPnet-4f [3]	81.1/79.9	75.4/74.2	81.5/81.1	74.1/73.6	84.6/82.0	77.2/74.7	77.2/76.5	75.0/74.4
FSD++-7f [9]	-/-	76.8/75.5	81.4/80.9	73.3/72.9	85.1/82.2	78.2/75.4	81.2/80.3	78.9/78.1
Clusterformer-3f(Ours)	**83.3/81.7**	**77.7/76.2**	**81.4/80.9**	**74.1/73.7**	**85.9/82.7**	**79.2/76.1**	**82.5//81.6**	**79.6/78.7**

Table 1. Performance comparison with state-of-the-art methods on the Waymo dataset with 202 validation sequences (about 40k samples) for single and multi-frame input. The '3f', '4f', and '7f' denote the different input frames for the model. All reported results are from a single model without any test-time augmentations(TTA).

4.2. KITTI dataset

The KITTI dataset consists of 7481 training samples and 7518 testing samples, and the training samples are further split into 3712 and 3769 samples for training and validation respectively. KITTI adopts the 3D mAP with a rotated IoU threshold of 0.7 to evaluate car category, and the mAP is divided into three difficulty levels(easy, moderate, and hard) according to the object size, truncation level, and occlusion.

4.3. Implementation Details

Network architecture. For WOD, the corresponding axis ranges are set as $(-75.2, 75.2)$, $(-75.2, 75.2)$, $(-2, 4)$, and the voxel size is set as (0.1m, 0.1m, 0.15m) in X, Y, Z axis, respectively, for voxelization. For KITTI, the corresponding axis ranges are set as $(0, 70.4)$, $(-40.0, 40.0)$, $(-3, 1)$, and the voxel size is set as (0.1m, 0.1m, 0.1m) in X, Y, Z axis, respectively, for voxelization. Our Clusterformer consists of four transformer decoder layers, and the hidden channels and the number of heads are set as 128 and 4, respectively. Since the query already contains position information, we only adopt position encoding on the key and valuer features in our intra-cluster cross-attention layer by adding the encoded voxel coordinates information. For the multi-frame version Clusterformer, we simply concatenate the multi-frame point clouds and feed them into our Clusterformer without additional structural changes.

Training and inference. We use eight 3090 GPUs to train the proposed Clusterformer with batch-size 16 for 12 epochs and 80 epochs for WOD and KITTI dataset, respectively, unless otherwise specified. The AdamW optimizer and one-cycle learning rate scheduler with a maximal learning rate of 0.001 are adopted. During the training, data augmentations including gt-sample, random flip, and rotation are introduced to improve the performance, meanwhile, we also follow [41, 1] use the fade-strategy to drop the data augmentations in the last epoch for avoiding overfitting. The whole WOD is used to report the final results compared with the other 3D detectors, while only 20% of data is used in the ablation experiment.

4.4. Comparisons on Waymo Open Dataset

As shown in Table 1, we present a comparison between the wide range of state-of-the-art 3D detectors with our Clusterformer on the WOD validation set for the single-frame and multi-frame input. Our Clusterformer achieves state-of-the-art performance on the WOD validation set among the mainstream detectors with only 12 epochs of training. Specifically, as for the single frame input, the Clusterformer acquires 72.3 L2 mAPH, which is 3.2 higher than the previous best query-based 3d detector Centerformer [41]. Our Clusterformer also outperforms the current best two-stage detectors FSD [9] with 1.8 higher L2 mAPH. For the multi-frames input, our Clusterformer with three frames acquires 76.2 L2 mAPH, which still outperforms the precious 3D detectors by a large margin. Our Clusterformer even suppresses the FSD++ [9] by 0.7 L2

Methods	mAP/mAPH L1	mAP/mAPH L2	Vehicle 3DAP/APH L1	Vehicle 3DAP/APH L2	Pedestrian 3DAP/APH L1	Pedestrian 3DAP/APH L2	Cyclist 3D AP/APH L1	Cyclist 3D AP/APH L2
RangeDet [8]	71.7/69.8	65.8/64.1	75.8/75.3	67.1/66.7	74.7/71.0	68.5/65.1	64.5/63.0	61.9/60.4
CenterPoint-Voxel [36]	-/69.0	-/-	81.1/80.6	-/-	76.6/70.5	68.8/63.2	72.3/71.1	69.7/68.5
PV-RCNN++ [26]	77.9/75.6	72.4/70.1	**81.6/81.2**	**73.8/73.4**	78.1/72.0	74.1/69.0	71.9/70.7	69.2/68.1
Pillarnet-34 [25]	78.1/75.9	71.7/69.5	79.3/78.8	70.6/70.2	81.3/76.3	73.2/68.0	73.7/72.7	71.2/70.2
AFDetV2-lite [13]	77.5/75.2	72.2/70.0	80.5/80.0	73.0/72.6	79.8/74.3	73.7/68.6	72.4/71.2	69.8/68.7
Clusterformer(Ours)	**81.1/78.9**	**75.0/73.0**	**81.6/81.2**	73.1/72.8	**83.4/78.3**	**76.6/71.9**	**78.4/77.1**	**75.5/74.3**

Table 2. Performance comparison with the state-of-the-art 3D detectors on the WOD test set, all reported results are from single model without any test-time augmentations for single-frame input.

mAPH with fewer input frames. Such results demonstrate the effectiveness of the proposed cluster-based transformer.

We also submit the prediction results to the official online server for the evaluation result on the test set of Clusterformer. Table 2 shows the comparison with the published results on the WOD test set. Our Clusterformer acquires 78.9 L1 mAPH and 73.0 L2 mAPH for the single-frame input, which outperforms the previous state-of-the-art 3D detectors, especially on the *pedestrian* and *cyclist* category.

4.5. Comparisons on KITTI Dataset

We further conduct experiments on the KITTI dataset, the results are summarized in Table 3. As we can see, our Clusterformer also achieves competitive performance on the KITTI datasets, which suppress the two-stage detectors CT3D and MsSVT-TS by 0.7 and 2.0 mAP on moderate difficult level, additionally, Our Clusterformer achieves the best performance on the hard difficult level with 79.28 mAP. Such results demonstrate the effectiveness and generalization of our Clusterformer on various datasets.

Method	3D Car(IoU=0.7) easy	moderate	hard
Second [34]	88.61	78.62	77.22
PointPillars [14]	86.62	76.06	68.91
VOTR-TSD [18]	89.04	84.04	78.68
CT3D [24]	89.54	86.06	78.99
VoxelSet [12]	89.21	**86.71**	78.56
MsSVT-TS [4]	89.32	84.66	78.94
Clusterformer(ours)	**89.76**	86.69	**79.28**

Table 3. Performance comparison on the KITTI val split with AP calculated by 11 recall points for the Car category.

4.6. ablation study

In this section, we conduct ablative studies on Waymo dataset to investigate key designs in our Clusterformer, Note that we train all the ablation experiments with 24 epochs.

Effects of cluster-based transformer structure. We first investigate the effectiveness of the proposed cluster-based transformer structure. As shown in Table 4, when we remove the cluster-based transformer structure and directly aggregate the features in a cluster by pooling function to make predictions (line one in the table) which is similar to the previous point-based detectors [21, 35, 38], the overall L2 mAPH drops about 2.6. Such a result demonstrates the superiority of the proposed cluster-based transformer structure. Compared with aggregating the features in a cluster directly, the intra-cluster cross-attention can aggregate the meaningful information in a cluster adaptively and obtain 1.4 L2 mAPH improvement. The cluster-wise self-attention layer between the queries can reason the pairwise relation of different candidates, which brings 0.4 L2 mAPH improvement. With the Query2Key strategy, the L2 mAPH has been further improved by 0.8.

ICCA	SA	Query2Key	mAPH/L2 Vehicle	Pedestrian	Cyclist	Overall
			65.9	68.7	71.3	68.6
✓			66.6	69.4	73.9	70.0 ↑ 1.4
✓	✓		67.1	69.7	74.4	70.4 ↑ 0.4
✓	✓	✓	**69.0**	**70.1**	**74.7**	**71.2** ↑ 0.8

Table 4. The ablation results of the proposed cluster-based transformer structure on the WOD validation set. Where ICCA and SA denote the intra-cluster cross-attention and self-attention layer.

Discussion of different query initialization strategy and interaction range. To demonstrate the effectiveness of our cluster-based query initialization strategy and intra-cluster interaction, we conduct experiments with different initialization strategy and interaction range for comparison as shown in Table 5.

Query Initialization	Interaction Range	Vehicle mAPH L1	L2
FPS initialization	Global-wise	51.2	44.2
	Distance-wise	70.5	61.7
	intra-cluster	-	-
Cluster Center initialization	Global-wise	56.8	49.4
	Distance-wise	74.4	65.3
	intra-cluster	**76.0**	**66.9**
Zero initialization	Global-wise	38.9	35.0
	Distance-wise	71.2	62.2
	intra-cluster	75.0	65.9

Table 5. Performance comparison of different query initialization strategies and interaction ranges. The distance-wise interaction range represents the voxel features with a distance less than the threshold from the query point will attend in the interaction, since the distance threshold is sensitive to different categories, we only present the result of *vehicle* category with a 2m distance threshold, which is enough to cover a vehicle. The '-' represents the FPS initialization can not perform intra-cluster interaction.

We discuss the results of Table 5 in terms of different query initialization strategies.

FPS initialization: We encode the point positions acquired from the FPS as the initial queries which is similar to 3DETR. Although the FPS is adopted only on the foreground point clouds, such query initialization strategy still can not get satisfactory results with only 61.7 L2 mAPH as shown in row 2 of Table 5.

Cluster Center initialization: Our cluster center initialization strategy outperforms the FPS initialization strategy by 3.6 L2 mAPH with distance-wise interaction range. Such a result demonstrates the effectiveness of our cluster center initialization strategy. Additionally, the cluster center query initialization strategy allows us to perform intra-cluster interaction, which further brings 1.6 L2 mAPH improvement. We argue that this is because compared with intra-cluster interaction, the distance-wise interaction inevitably introduces features in other objects for queries, which may lead to adverse effects on box regression.

Zero initialization: We also initialize the queries by zero value, and such empty query with intra-cluster interaction still achieves promising results with 65.9 L2 mAPH. This result demonstrates that, based on a reasonable interaction range for queries, we can achieve satisfactory performance without an elaborately designed query initialization strategy. It is worth mentioning that for global-wise interaction, all the query initialization strategies obtain poor results since it is difficult for the model to converge in such a large interaction range in 3D space.

The effect of Query2Key on different decoder numbers. To further investigate the effect of Query2Key strategy with different decoder layers, we conduct the experiments by gradually adding the layers with or without Query2Key strategy.

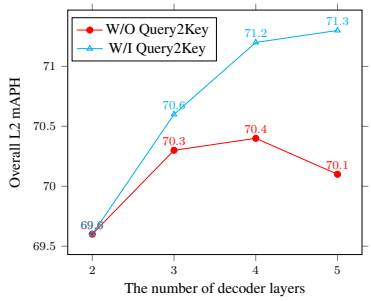


Figure 4. The performance comparison of different decoder numbers with or without the proposed Query2Key strategy.

As shown in Fig. 4, without the proposed Query2Key strategy, the performance of our Clusterformer will slightly be improved with the decoder number increasing from 2 to 4. However, such improvement is gradually attenuated, and when the number of decoders increases from 4 to 5, the performance even drops 0.3 L2 mAPH. In contrast, when the proposed Query2Key strategy is adopted, adding transformer decoder layers can bring more significant improvement. Specifically, when the number of decoders increases from 2 to 4, the L2 mAPH gains 1.6 improvement compared with 0.8 L2 mAPH improvement without Query2Key strategy. We believe such improvement mainly comes from that the updating key and value features contain more object-level information by the Query2Key strategy.

The effect of the local maximum operation. In the cluster generating process, we select the local maximum value in the pseudo heatmap to avoid generating multiple centers for one object which would lead to cluster errors. Table 6 shows the results of different window size settings, we can see that without the local maximum operation(last row in Table 6), the performance drops significantly, especially on the *vehicle* and *pedestrian* category. We also find the window size of (5,3,3) acquires the best performance.

Window size	mAPH/L2			
	Vehicle	Pedestrian	Cyclist	overall
(7,3,5)	68.7	70.0	74.2	70.9
(5,3,3)	**69.0**	**70.1**	**74.7**	**71.2**
(3,1,3)	68.2	68.9	74.5	70.5
(1,1,1)	67.6	68.7	73.6	69.9

Table 6. The performance comparison of different window size setting in the local maximum operation for each category, where the (7,3,5) are for vehicle, pedestrian, and cyclist, respectively.

The effect of NMS with different matching strategies. We experiment with Hungarian matching and Max-IoU matching strategy used in training to investigate the effect of NMS with different matching strategies, the results are summarized in Table 7. We can see that, with different matching strategies, removing NMS will degrade the performance, but still achieves acceptable detection accuracy, this is because the local maximum operation can undertake the role of NMS to a certain extent. On the other hand, the Hungarian matching strategy can further reduce the model's demand for NMS, the Hungarian matching strategy without NMS acquires 70.2 L2 mAPH, which is 1.0 lower than the Max-IoU matching strategy with NMS. Such results demonstrate our Clusterformer can work in an end-to-end manner by removing NMS with a slight performance drop.

Method	mAPH/L2			
	Vehicle	Pedestrian	Cyclist	overall
Hungarian matching W/I NMS	**69.1**	69.8	74.5	71.1
Hungarian matching W/O NMS	**69.1**	68.1	73.4	70.2
Max-IoU matching W/I NMS	69.0	**70.1**	**74.7**	**71.2**
Max-IoU matching W/O NMS	68.2	65.4	70.8	68.1

Table 7. The effect of NMS with different matching strategy.

4.7. The inference speed and memory usage

We also present the latency and inference memory usage of our Clusterformer and make a comparison with other state-of-the-art query-based 3D detectors. Since Centerformer [41] and Transfusion [41] don't report the latency and memory usage, we re-implement them based on their officially released codes to report the results. As shown in

Table 8, compared with the Centerformer [41], our Clusterformer acquires a better detection performance with lower latency, *i.e.*, 103ms *vs.*123ms, and only about half of the inference memory usage.

Methods	Latency	Inference Memory	mAPH/L2
Transfusion-L [1]	118ms	10.8G	64.9
Centerformer [41]	123ms	10.4G	68.9
Clusterformer(Ours)	103ms	5.7G	**72.3**

Table 8. The latency and inference memory comparison with other state-of-the-art query-based 3D detectors on the WOD val set. All the latency results are measured on a NVIDIA GTX 3090 GPU.

5. Conclusion

In this paper, we propose a novel cluster-based transformer structure for 3D object detection called Clusterformer which directly generates the proposals from the sparse voxel features without projecting the voxel feature into the BEV plane. Our Clusterformer leverages the cluster to acquire the initial queries which contain accurate location information of the candidates from the 3D space and perform intra-cluster interaction to decode the queries. Such initial queries and interaction mode can effectively improve the performance and convergence speed of query-based detectors. We also design a Query2Key strategy to enhance the key and value features iteratively. Experimental results show that the proposed Clusterformer outperforms the other state-of-the-art 3D detectors on WOD and KITTI datasets.

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, 2, 3, 4, 5, 6, 9

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 1, 2

[3] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *Proceedings of the European Conference on Computer Vision*, pages 680–697, 2022. 6

[4] Shaocong Dong, Lihe Ding, Haiyang Wang, Tingfa Xu, Xinli Xu, Jie Wang, Ziyang Bian, Ying Wang, and Jianan Li. Mssvt: Mixed-scale sparse voxel transformer for 3d object detection on point clouds. In *Advances in Neural Information Processing Systems*, 2022. 7

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[6] Gopi Krishna Erabati and Helder Araujo. Li3detr: A lidar based 3d detection transformer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4250–4259, 2023. 1, 3

[7] Lue Fan, Ziqi Pang, Tianyuan Zhang, Yu-Xiong Wang, Hang Zhao, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8458–8468, 2022. 1, 6

[8] 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. 7

[9] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3d object detection. *arXiv preprint arXiv:2301.02562*, 2023. 2, 5, 6

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 2, 5

[11] Benjamin Graham and Laurens Van der Maaten. Submanifold sparse convolutional networks. *arXiv preprint arXiv:1706.01307*, 2017. 2

[12] Chenhang He, Ruihuang Li, Shuai Li, and Lei Zhang. Voxel set transformer: A set-to-set approach to 3d object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8417–8427, 2022. 1, 6, 7

[13] Yihan Hu, Zhuangzhuang Ding, Runzhou Ge, Wenxin Shao, Li Huang, Kun Li, and Qiang Liu. Afdetv2: Rethinking the necessity of the second stage for object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 969–979, 2022. 6, 7

[14] 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. 1, 2, 7

[15] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11809–11818, 2022. 2, 3, 4

[16] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8895–8904, 2019. 2

[17] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 1

[18] Jiageng Mao, Yujing Xue, Minzhe Niu, Haoyue Bai, Jiashi Feng, Xiaodan Liang, Hang Xu, and Chunjing Xu. Voxel transformer for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3164–3173, 2021. 1, 6, 7

[19] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 2, 3

[20] Duy-Kien Nguyen, Jihong Ju, Olaf Booij, Martin R Oswald, and Cees GM Snoek. Boxer: Box-attention for 2d and 3d transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4773–4782, 2022. 1, 2, 3

[21] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 2, 4, 5, 7

[22] 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. 1, 2

[23] 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. 1, 2

[24] Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2743–2752, 2021. 7

[25] Guangsheng Shi, Ruifeng Li, and Chao Ma. Pillarnet: Real-time and high-performance pillar-based 3d object detection. In *Proceedings of the European Conference on Computer Vision*, pages 35–52, 2022. 1, 2, 6, 7

[26] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn++: Point-voxel feature set abstraction with local vector representation for 3d object detection. *International Journal of Computer Vision*, pages 1–21, 2022. 1, 2, 6, 7

[27] 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. 1, 2

[28] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020. 3, 6

[29] Kshitij Sirohi, Rohit Mohan, Daniel Büscher, Wolfram Burgard, and Abhinav Valada. Efficientlps: Efficient lidar panoptic segmentation. *IEEE Transactions on Robotics*, 38(3):1894–1914, 2021. 3

[30] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 2, 5

[31] Pei Sun, Mingxing Tan, Weiyue Wang, Chenxi Liu, Fei Xia, Zhaoqi Leng, and Dragomir Anguelov. Swformer: Sparse window transformer for 3d object detection in point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 426–442, 2022. 6

[32] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2567–2575, 2022. 2, 4

[33] Shuangjie Xu, Rui Wan, Maosheng Ye, Xiaoyi Zou, and Tongyi Cao. Sparse cross-scale attention network for efficient lidar panoptic segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2920–2928, 2022. 3

[34] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 6, 7

[35] 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. 1, 2, 7

[36] 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, 6, 7

[37] 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*, pages 949–958, 2022. 2, 4

[38] Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Wan, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18953–18962, 2022. 1, 2, 7

[39] 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. 1, 2, 5

[40] Zixiang Zhou, Yang Zhang, and Hassan Foroosh. Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13194–13203, 2021. 2, 3

[41] Zixiang Zhou, Xiangchen Zhao, Yu Wang, Panqu Wang, and Hassan Foroosh. Centerformer: Center-based transformer for 3d object detection. In *Proceedings of the European Conference on Computer Vision*, pages 496–513, 2022. 1, 2, 3, 4, 5, 6, 8, 9

[42] 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, 2, 5