Li3DeTr: A LiDAR based 3D Detection Transformer

Gopi Krishna Erabati and Helder Araujo
Institute of Systems and Robotics
University of Coimbra, Portugal
{gopi.erabati, helder}@isr.uc.pt

Abstract

*Inspired by recent advances in vision transformers for object detection, we propose Li3DeTr, an end-to-end **LiDAR** based **3D** **De**tection **Tr**ansformer for autonomous driving, that inputs LiDAR point clouds and regresses 3D bounding boxes. The LiDAR local and global features are encoded using sparse convolution and multi-scale deformable attention respectively. In the decoder head, firstly, in the novel Li3DeTr cross-attention block, we link the LiDAR global features to 3D predictions leveraging the sparse set of object queries learnt from the data. Secondly, the object query interactions are formulated using multi-head self-attention. Finally, the decoder layer is repeated L_{dec} number of times to refine the object queries. Inspired by DETR, we employ set-to-set loss to train the Li3DeTr network. Without bells and whistles, the Li3DeTr network achieves 61.3% mAP and 67.6% NDS surpassing the state-of-the-art methods with non-maximum suppression (NMS) on the nuScenes dataset and it also achieves competitive performance on the KITTI dataset. We also employ knowledge distillation (KD) using a teacher and student model that slightly improves the performance of our network.*

1. Introduction

With the advent of deep learning networks for computer vision [16, 37] and large-scale datasets [10] the research on perception systems for scene understanding of autonomous vehicles is growing rapidly. 3D object detection is one of the key processes of autonomous driving, which is a two fold process of classification and localization of the objects in the scene. LiDAR is one of the significant sensors of autonomous vehicles which provides precise 3D information of the scene. Although there is a huge progress in 2D object detection approaches [2, 12, 22, 32, 33, 40], the CNN-based approaches are not well directly adapted to LiDAR point clouds due to their sparse, unordered and irregular nature.

Earlier approaches for 3D object detection on LiDAR data can be divided into two types: point-based and grid-based methods. Point-based methods [27, 36, 49] are based on point operations [29, 30] which detect the 3D objects directly from the point clouds. Grid-based methods either voxelize the points into volumetric grids or project the points to Birds Eye View (BEV) space. The advantage of BEV projection is that it preserves euclidean distance, avoids overlapping of objects and the object size is invariant to distance from ego vehicle which is significant for autonomous driving scenarios. The sparse CNN-based voxel feature extraction [47] is advantageous but it can not extract rich semantic information with limited receptive fields. We mitigate this issue by employing a multi-scale deformable attention [55] encoder to capture global LiDAR feature maps.

Earlier approaches either use two-stage detection pipeline [7, 36] or anchors [19, 53] or anchor-free networks [42, 43, 51] for 3D object detection, but all of them employ post-processing method like NMS to remove redundant boxes. Inspired by Object-DGCNN [45], we formulate the 3D object detection problem as a direct set prediction problem to avoid NMS.

We propose an end-to-end, single-stage LiDAR based 3D Detection Transformer (Li3DeTr) network to predict the 3D bounding boxes for autonomous driving. Firstly, the voxel features are extracted with SECOND [47] by leveraging sparse convolutions [15] and BEV transformation or with PointPillars [19]. Secondly, we employ an encoder module with multi-scale deformable attention [55] to capture rich semantic features and long range dependencies in BEV feature maps to generate LiDAR global features. The LiDAR global features are passed to the decoder module. Finally, we introduce a novel Li3DeTr cross-attention block in the decoder to link the LiDAR global features to the 3D object predictions leveraging the learnt object queries. The object queries interact with each other in multi-head self-attention block [41]. The object queries are iteratively refined and 3D bounding box parameters are regressed in every decoder layer. Inspired by DETR [2], we use set-to-set loss to optimize our network during training.

We conduct experiments on two publicly available au-

tonomous driving benchmarks, nuScenes [1] and KITTI [14] dataset. Our network achieves 61.3% mAP and 67.6% NDS on the nuScenes dataset surpassing the state-of-the-art CenterPoint [51] and Object-DGCNN [45] by 3.3% mAP (and 2.1% NDS) and 2.6% mAP (and 1.6% NDS) respectively.

Our main contributions are as follows:

- We propose an end-to-end, single-stage LiDAR based 3D Detection Transformer (Li3DeTr) for autonomous driving. Our method achieves 61.3% mAP and 67.6% NDS on the nuScenes [1] dataset which surpassed state-of-the-art LiDAR based object detection approaches. Our method achieves competitive performance (without NMS) to other approaches (with NMS) on the KITTI [14] dataset. Similar to DETR [2], our approach does not require NMS, hence it is effective to apply knowledge distillation with teacher and student model to improve the accuracy.
- We introduce a novel Li3DeTr cross-attention block to link the LiDAR global encoded features to 3D object predictions leveraging the learnt object queries. The attention mechanism in encoder and decoder helps to detect large-size objects effectively as shown in Table 3. The ablation study shown in Table 6 justifies our novel Li3DeTr cross-attention block.
- We release our code and models to facilitate further research.

2. Related Work

The LiDAR point cloud based 3D object detection approaches can be divided into two categories: point-based and grid-based, depending on the type of data representation used to predict the 3D bounding boxes.

Point-based methods [28, 35, 36, 49, 50] directly use the sparse and unordered set of points to predict 3D bounding boxes. The point features are aggregated by multi-scale/multi-resolution grouping and set abstraction [29, 30]. PointRCNN [36] employs a two-stage pipeline for 3D object prediction. PVRCNN [35] models point-voxel based set abstraction layer to leverage the advantage of point and voxel based methods. Frustum-PointNet [28] uses 2D object detection to sample a frustum of points to apply PointNet [29] to predict 3D objects. Although point-based methods achieve large receptive fields with set abstraction layer, they are computationally expensive.

Grid-based methods. As the LiDAR point clouds are sparse and unordered set of points, many methods project the points to regular grids such as voxels [47, 53], BEV pillars [19] or range projection [3, 13, 39]. The point clouds are discretized into 3D voxels [38, 53] and 3D CNNs are employed to extract voxel-wise features. However, 3D CNNs are computationally expensive and requires large memory, in order to mitigate this problem [5, 47] use sparse 3D CNNs [15] for efficient voxel processing. The LiDAR point cloud is projected into BEV map in PointPillars [19] and PIXOR [48] and 2D CNNs are employed to reduce the computational cost, however such projection induces 3D information loss. In order to mitigate this issue, some methods [47, 51] compute voxel features using sprase convolutions and then project the voxel features into BEV space, and finally predict the 3D bounding boxes in the BEV space. As this approach takes the advantage of voxel and BEV space, we test our network with SECOND [47] and PointPillars [19] feature extraction networks. In order to achieve large receptive fields similar to point-based methods [29, 30], we model long-range interactions of local LiDAR features using multi-scale deformable attention [55] block in our encoder to obtain LiDAR global features.

Transformer-based methods. Earlier approaches [7, 19, 35, 36, 50, 53] object detection head employ anchor boxes to predict the objects, however anchor boxes involve handcrafted parameter tuning and they are statistically obtained from the dataset. To mitigate this issue, some approaches [5, 43, 48, 51] followed anchor-free pipeline by computing per-pixel or per-pillar prediction. But these approaches use NMS to remove redundant boxes. DETR [2] is the first transformer architecture which formulated 2D detection problem as a direct set prediction to remove NMS. Our network follows similar formulation for 3D object detection. Some approaches [24, 27, 34] used transformer for feature extraction networks. 3DETR [25] is a fully transformer based architecture for 3D object detection using vanilla transformer [41] block with minimal modifications. 3DETR directly operate and attend on points whereas our approach voxelize the points and attend the BEV global voxel features which is computationally efficient for autonomous driving scenarios. 3DETR employs downsampling and set-aggregation operation [30] on the input points of indoor scenarios because the computational complexity of self-attention increases quadratically ($\mathcal{O}(n^2)$) with the number of input points. Moreover, 3DETR is effective on indoor datasets, where the points are dense and concentrated. Object-DGCNN [45] employs a graph-based model for transformer-based 3D object detection for outdoor environments. BoxeR [26] introduces a novel and simple Box-Attention which enables spatial interaction between grid features. BoxeR-2D enables end-to-end 2D object detection and segmentation tasks, which can be extended to BoxeR-3D for end-to-end 3D object detection. VISTA [11] is a plug and play module to adaptively fuse multi-view features in a global spatial context, incorporated with [5, 51]. It introduces dual cross-view spatial attention to leverage the information in BEV and Range View (RV) features. We formulate our model with voxel-BEV based CNN backbone architecture for local feature extraction and an attention-based architecture for global feature extraction to increase the re-

ceptive field size and finally a transformer decoder head to link global features and 3D predictions.

3. Methodology

Our LiDAR based 3D Detection Transformer (Li3DeTr) architecture inputs LiDAR point cloud and predicts 3D bounding boxes in large-scale outdoor environments such as autonomous driving. The network contains two main modules: backbone and transformer encoder-decoder as shown in Figure 1. Inspired by the state-of-the-art 3D object detection approaches [15, 19], our CNN-based backbone (§ 3.1) module learns grid-based local voxel features. Specifically, we employ BEV grid, not only because the 2D grid-like features are a good trade-off between accuracy and efficiency but also very relevant for autonomous driving, as there is possibly one object on every grid cell on the ground plane. The multi-scale deformable attention [55] based encoder (§ 3.2) module learns the multi-scale global voxel features. The encoder module alternates between multi-scale deformable attention block and multi-layer perceptron (MLP) block and is repeated L_{enc} number of times. The novel Li3DeTr cross-attention block in the decoder (§ 3.3) module links the global voxel features to the 3D object predictions leveraging the learnt object queries. The object queries interact with each other in multi-head self-attention [41] block. The decoder module is repeated L_{dec} number of times with alternating multi-head self-attention, Li3DeTr cross-attention and MLP blocks. The refined object queries are transformed into 3D bounding box parameters and the network is trained end-to-end using permutation-invariant loss [2] (§ 3.4).

3.1. Backbone: Local features

Our network inputs LiDAR point cloud $\mathcal{P} = \{p_1, \ldots, p_i, \ldots, p_N\} \subset \mathbb{R}^3$. To accelerate the 3D object detection for large-scale point clouds, we scatter the points into BEV grid and use CNNs to extract local point features. We test two pipeline: 1) We voxelize the point cloud with [0.1, 0.1, 0.2] metres voxel size and employ SparseConv [15] to compute 3D sparse convolutions and obtain local voxel features. The empty voxels are filled with zeros and the sparse voxels are transformed to a BEV 2D grid-like features. 2) We convert the point cloud into a dense BEV pillar map as in PointPillars [19] with [0.2, 0.2, 8] metres pillar resolution. We use pillar feature net to transform the pillar features. Finally, we employ SECOND [47] backbone to extract local voxel features from the sparse voxel or BEV pillar features and further transform them using a feature pyramid network (FPN) [20] to obtain multi-scale local voxel feature maps $\mathcal{F}_1, \mathcal{F}_2, \mathcal{F}_3, \mathcal{F}_4$ where $\mathcal{F}_j \subset \mathbb{R}^{H^j \times W^j \times C^j}$.

3.2. Encoder: Global features

In order to obtain global voxel features from the local voxel feature maps, we employ multi-scale deformable attention [55] mechanism because the conventional attention mechanism [41] leads to unacceptable computational complexities in encoding high resolution feature maps. The multi-scale deformable attention combines the best of the sparse sampling of deformable convolution [9] and long-range relation framework of transformers [41]. The input and output to encoder module are multi-scale feature maps with the same resolution. The deformable attention attends only a small set of key sampling points around a reference point and thereby reduces the computational complexity. The reference point in the deformable self-attention for each query pixel is itself. Each encoder layer consist of multi-scale deformable self-attention and MLP blocks with residual connections and is repeated L_{enc} times. The global voxel features extracted from the encoder is passed to Li3DeTr cross-attention block in the decoder.

3.3. Decoder

The state-of-the-art 3D object detection approaches either formulates the detection head with dense set of anchor boxes or dense per-pillar prediction such as [43, 51], followed by NMS. We remove the need of post-processing step (like NMS) by formulating the detection head to predict a set of bounding boxes instead of per-pillar prediction. This is formulated in the decoder as detailed below.

The decoder inputs a set of object queries $\mathcal{Q}^l = \{q_i^l\}_{i=1}^{N_q} \in \mathbb{R}^d$ (where $l \in \{1, 2, \ldots, L_{dec}\}$, N_q is number of queries and $\mathcal{Q}^1$ are learnt with the model weights) and global voxel feature maps $\{\mathcal{F}_j\}_{j=1}^4$ and it consists of decoder layer that is repeated L_{dec} number of times to refine the object queries.

For the first decoder layer, the 3D reference points are encoded from the object queries using a single-layer fully connected (FC) network and sigmoid normalization as in Equation 1.

$$r_i = \phi_{ref}(q_i), \tag{1}$$

where $r_i \in [0, 1]^3$ and ϕ_{ref} is a FC layer. Each decoder layer consists of Li3DeTr cross-attention block, multi-head self attention block and MLP block with skip connections as shown in Figure 1.

Li3DeTr cross-attention block inputs the object queries $\mathcal{Q} = \{q_i\}_{i=1}^{N_q}$ (we drop the layer index for simplicity), 3D reference points r_i and LiDAR global multi-scale feature maps $\{\mathcal{F}_j\}_{j=1}^4$. The formulation of Li3DeTr cross-attention block is illustrated in Figure 2.

Let $\mathcal{R}_{ji}$ represents the transformation for the projection of reference point r_i into scale j of LiDAR global voxel feature map. The LiDAR BEV feature map at scale j is bilinearly sampled at the location of projection of reference point

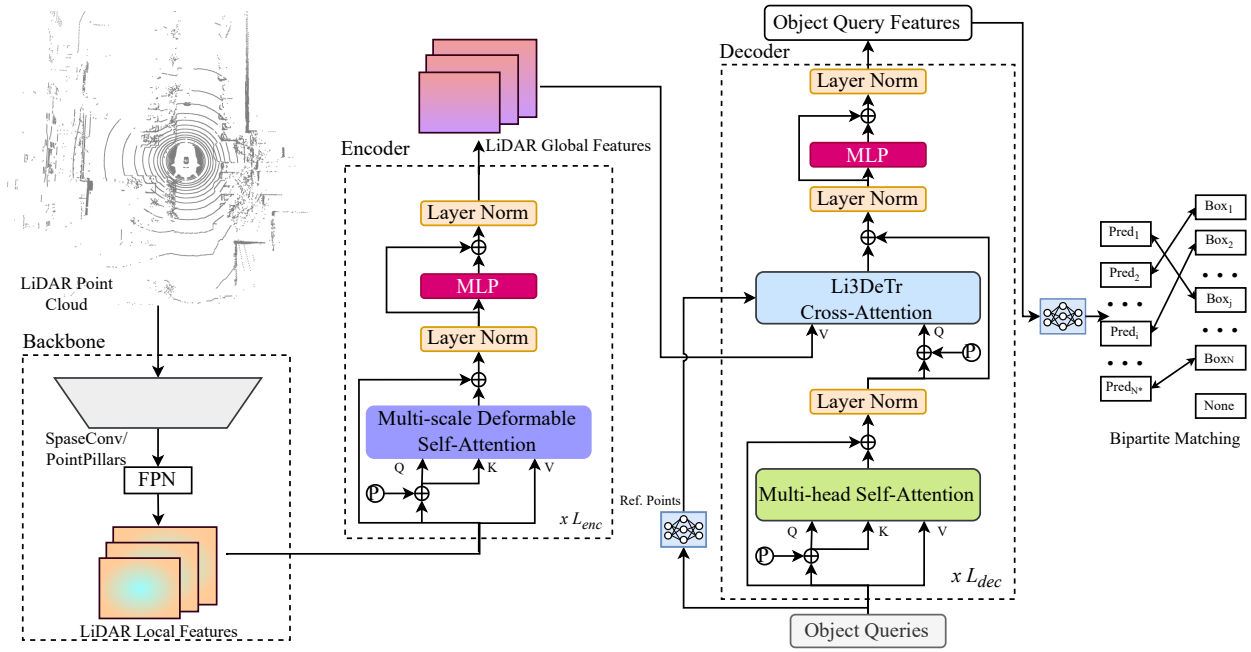

Figure 1: An overview of Li3DeTr architecture. It is an end-to-end, single-stage network which inputs LiDAR point cloud and predicts 3D bounding boxes. The local and global point features are linked to object predictions in transformer encoder-decoder architecture leveraging object queries.

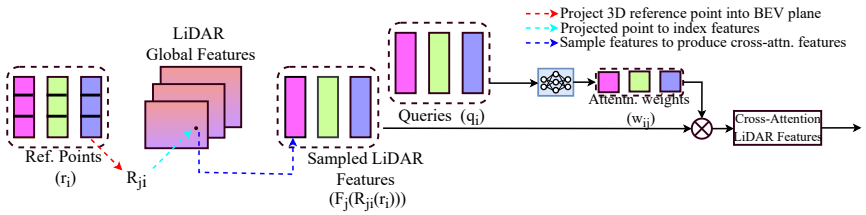

Figure 2: An overview of Li3DeTr cross-attention block

$(\mathcal{R}_{ji}(r_i))$, given by $\mathcal{F}_j(\mathcal{R}_{ji}(r_i))$. The attention weights (w_{ij}) for each query q_i at each sampled feature scale j is computed by a FC layer (ϕ_{samp}) and sigmoid normalization, where $w_{ij} = \phi_{samp}(q_i)$. The sampled features from multi-scale feature maps are added together to obtain cross-attention features $(\mathcal{F}_i^{CAttn})$ for i-th reference point as:

$$\mathcal{F}_i^{CAttn} = \sum_{j=1}^{4} \mathcal{F}_j(\mathcal{R}_{ji}(r_i)).w_{ij} \tag{2}$$

Finally, we update the queries as,

$$q_i = q_i + \mathcal{F}_i^{CAttn} + PE(c_i), \tag{3}$$

where PE is positional encoding of reference point to make the queries location aware. The queries interact with each other in the multi-head self-attention block and we follow skip connection following [41]. The object queries $(\mathcal{Q}^l)$ are updated in each decoder layer.

We employ two FC layers ϕ_l^{reg} and ϕ_l^{cls} to predict offset to box position $\Delta p_i^l \in \mathbb{R}^3$, box size (l_i^l, w_i^l, h_i^l), box orientation $(\sin\theta_i^l, \cos\theta_i^l)$, box velocity (v_{xi}^l, v_{yi}^l) and class label $(\hat{y}_i^l)$ respectively for each object query (q_i^l).

We refine the reference points for each query in each decoder layer except for the first decoder layer (which are encoded using Equation. 1) by using the predictions of box position in the previous layer, as

$$r_i^{l+1} = r_i^l + \Delta p_i^l \tag{4}$$

3.4. Loss

Similar to [2, 44, 45], the error between predictions and ground-truths is calculated by set-to-set loss. Firstly, the one-to-one assignment between predictions and ground-truths is performed by Hungarian algorithm [18]. Secondly, we use bipartite matching to match predictions and ground-truths. Finally, we employ L_1 loss and focal loss [21] to

calculate bounding box regression and classification loss respectively, given the bipartite matching.

4. Experiments

We evaluate Li3DeTr network on the publicly available autonomous driving datasets, nuScenes [1] and KITTI [14]. We introduce the experimental setup (§ 4.1) with dataset details and evaluation metrics (model settings and training details are provided in supplementary), and then present both qualitative and quantitative results (§ 4.2) and analysis on the nuScenes and KITTI dataset. We perform ablation studies (§ 4.3) to study the different configurations of our network.

4.1. Experimental Setup

nuScenes dataset [1] contains 750, 150 and 150 sequences (of $\sim$20s duration) with 28K, 6K and 6K annotated samples for training, validation and testing respectively. Each sample consists of 32-beam LiDAR point cloud with 30K points. The dataset also provides 9 non-key frames (called sweeps) to aggregate to one key-frame, resulting in $\sim$300K points per annotated frame. 10 different categories of objects are available to compute the metrics.

Kitti dataset [14] consists of 7481 and 7518 training and testing samples. The training samples are further divided into 3712 *train* and 3769 *val* splits. Each sample consists of 32-beam LiDAR point cloud. Cars, pedestrians and cyclists are the three main categories for evaluation.

Evaluation metrics. The two main metrics following the official evaluation of nuScenes dataset [1] are mean average precsion (mAP) and nuscenes detection score (NDS). In addition, we also evaluate true positive (TP) metrics: average translation error (ATE), average scale error (ASE), average orientation error (AOE), average velocity error (AVE), average attribute error (AAE). We follow the official evaluation metric of KITTI dataset [14] mean average precision (mAP) with IoU threshold of 0.7 for *car* category and 0.5 for *pedestrian* and *cyclist* categories..

4.2. Results

4.2.1 Quantitative Results

We compare Li3DeTr network with the state-of-the-art methods on the nuScenes [1] *test* dataset as shown in Table 1. Our network surpassed the state-of-the-art CNN-based CenterPoint [51] by 3.3% mAP and 2.1% NDS and transformer-based Object-DGCNN [45] network by 2.6% mAP and 1.6% NDS. Albeit CenterPoint [51] uses post-processing method like NMS, our approach of formulating object detection as a direct set prediction problem inspired by DETR [2] doesn't require NMS to obtain the gain in mAP and NDS. We also compare with the NMS-free state-of-the-art transformer-based network Object-DGCNN [45] with two different backbones: PointPillars [19] and VoxelNet [53]. Our transformer based NMS-free approach surpassed in both *pillar* [19] and *voxel* [53] backbones by 0.6% mAP (and 0.2% NDS) and 2.6% mAP (and 1.6% NDS) respectively. Although our Li3DeTr network outperforms most of the other methods in terms of mAP and NDS, VISTA-OHS [11] performs slightly better than our approach on the nuScenes *test* dataset. VISTA is a plug and play module to fuse multi-view features incorporated with [5] which requires post-processing methods like NMS, whereas our approach is a *standalone* method for 3D object detection without NMS. The perfromance of our approach compared with state-of-the-art approaches on nuScenes *val* dataset is provided in supplementary.

We employ knowledge distillation (KD) with a teacher and student model. The earlier works on 3D object detection involve NMS, so it is not effective to distill those models. As our approach is NMS-free, we can effectively distill the information between models with similar detection heads. We train a teacher model with the loss given in § 3.4 and then we train a student model (with same architecture as teacher model) with the supervision of output of teacher model and the ground-truth. With the KD, we achieve 62.2% mAP and 68.0% NDS which is 0.9% mAP and 0.4% NDS improvement over our model without KD.

We compare the recent works on KITTI [14] dataset for *car* category as shown in Table 2. Our network achieves a competitive performance to the state-of-the-art LiDAR-based approaches like VoxelNet [53], PointPillars [19], TANet [23] and SECOND [47] in terms of AP_{3D} and AP_{BEV} for *easy*, *moderate* and *hard* samples. Our network could not achieve the state-of-the-art performance on KITTI [14] dataset as compared to nuScenes [1] dataset because the transformer network is data hungry and KITTI dataset has 3712 samples for training which is approximately 7.5 times less number of training samples than nuScenes dataset (which has 28K training samples). In addition to this, the nuScenes dataset provides 9 non-key frames (called sweeps) to aggregate to one key frame, resulting in dense LiDAR points but KITTI dataset provides only one LiDAR key frame which results in sparse point cloud. However, our approach obtains competitive performance to the state-of-the-art transformer based architecture VoTr-SSD [24] which uses NMS. To the best of our knowledge this is the first transformer based 3D detection network to report results both on nuScenes and KITTI datasets, which compares the results of detection with an emphasis on training sample size which is significant for transformer based architectures alongside the density of LiDAR point clouds. We further provide comparison of methods in terms of AP_{3D} and AP_{BEV} on the *pedestrian* and *cyclist* categories for *easy*, *moderate* and *hard* samples in the supplementary.

Table 1: Comparison of recent works on nuScenes [1] *test* set.

Method	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓	NMS
PointPillars [19]	55.0	40.1	39.2	26.9	47.6	27.0	10.2	✓
SSN [54]	61.7	51.0	33.9	24.5	42.9	26.6	8.7	✓
CyliNet RG [31]	66.1	57.6	28.3	25.3	29.1	26.8	18.0	✓
CVCNet-ens [4]	66.6	58.2	28.4	24.1	37.2	22.4	12.6	✓
HotSpotNet [5]	66.0	59.3	27.4	23.9	38.4	33.3	13.3	✓
CenterPoint [51]	65.5	58.0	-	-	-	-	-	✓
VISTA-OHS [11]	**69.8**	**63.0**	25.6	23.3	32.1	21.6	12.2	✓
Object-DGCNN (pillar) [45]	62.8	53.2	34.6	26.5	31.6	26.0	19.1	✗
Object-DGCNN (voxel) [45]	66.0	58.7	33.3	26.3	28.8	25.1	19.0	✗
Ours (pillar)	63.0	53.8	35.1	26.4	32.1	26.5	19.0	✗
Ours (voxel)	**67.6**	**61.3**	30.5	25.4	35.2	26.7	12.5	✗

Table 2: Comparison of recent works in terms of AP_{3D} and AP_{BEV} detection on KITTI [14] *val* set. We list results for car category for *easy*, *moderate* and *hard* samples with IoU=0.7.

Method	AP_{3D}			AP_{BEV}			NMS
	Easy	Mod.	Hard	Easy	Mod.	Hard	
RGB & LiDAR							
MV3D [6]	71.2	62.6	56.5	86.5	78.1	76.6	✓
AVOD-FPN [17]	-	73.2	-	-	-	-	✓
F-PointNet [28]	87.3	70.9	63.6	88.1	84.0	76.4	✓
3D-CVF [52]	89.6	79.8	78.4	-	-	-	✓
LiDAR							
VoxelNet [53]	81.9	65.4	62.8	88.0	78.4	71.3	✓
PointPillars [19]	86.6	76.0	68.9	90.1	86.6	82.8	✓
TANet [23]	87.5	76.6	73.8	-	-	-	✓
SECOND [47]	87.4	76.4	69.1	89.4	83.8	78.6	✓
3DSSD [49]	89.7	79.4	78.6	**92.7**	**89.0**	**85.9**	✓
VoTr-SSD [24]	87.8	78.2	76.9	-	-	-	✓
Pointformer [27]	**90.0**	**79.6**	**78.8**	-	-	-	✓
Ours (voxel)	87.6	76.8	73.9	89.6	86.8	83.1	✗

4.2.2 Qualitative Results

The visualization of 3D bounding box predictions of our approach on the nuScenes dataset is shown in Figure 3. Although the LiDAR point clouds are sparse, our approach not only detects small objects like traffic cones but also efficiently detect large size objects like truck, bus, construction vehicle. This is possible with the local and global feature maps of backbone and attention mechanism in encoder alongside the cross-attention in the decoder. Our approach is also able to detect some cars which are not annotated in ground-truth. A short demo video of 3D object predictions of our network projected into BEV map is presented at `https://youtu.be/5pLnLRO_2-U`.

4.2.3 Analysis

The performance analysis of our approach by object category, object distance and object size compared to state-of-the-art LiDAR based CNN and transformer models is detailed below.

Object category. The performance of our network in terms of Average Precision (AP) for each object category compared to other state-of-the-art networks on the nuScenes [1] *val* dataset is shown in Table 3. We compare with Object-DGCNN [45] as it a *standalone* transformer model like ours for fair comparison. The transformer encoder which extracts global LiDAR features leveraging long range interactions using multi-scale deformable attention and decoder cross-attention significantly improves the AP of large

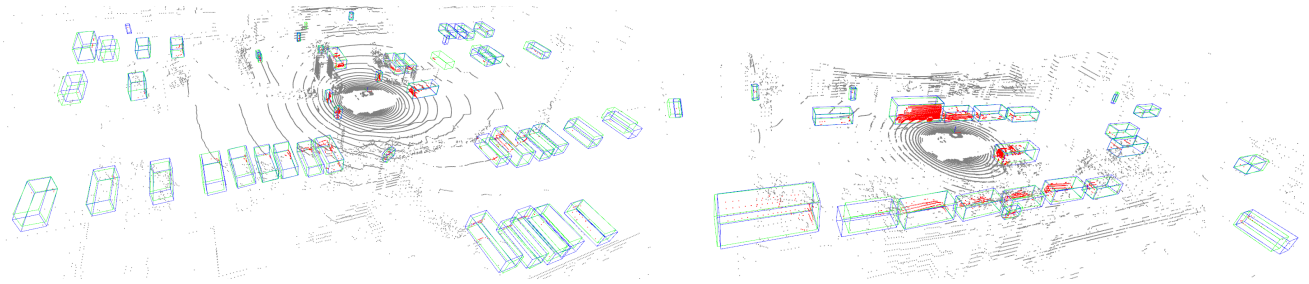

Figure 3: Visualization of results on the nuScenes dataset. Blue and green boxes represent predictions and ground-truth respectively. Points inside the bounding boxes are shown in red. Best viewed in color and zoom-in.

Table 3: Performance of our network in terms of Average Precision (AP) by object category on the nuScenes *val* set. CV - Construction Vehicle, Motor - Motorcycle, Ped - Pedestrian, TC - Traffic Cone, Barr - Barrier. $*$: MMDetection3D [8] implementation. The scores in green indicate the increase in performance with respect to scores in underline.

Method	Car	Truck	Trailer	Bus	CV	Bicycle	Motor	Ped	TC	Barr	mAP
Pointformer [27]	82.3	48.1	43.4	55.6	8.6	22.7	55.0	81.8	72.2	66.0	53.6
CenterPoint [51] $*$	85.1	53.0	35.4	66.8	13.9	34.4	55.2	84.6	66.9	67.5	56.2
VISTA [11]	85.0	57.4	39.9	66.4	21.2	51.7	66.6	84.5	68.5	66.8	60.9
Obj-DGCNN [45]	84.0	54.0	40.4	66.8	20.2	44.7	66.2	81.6	64.7	62.6	58.5
Ours	85.8	56.5 ↑ 2.5	43.0 ↑ 2.6	70.9 ↑ 4.1	22.9 ↑ 2.7	51.6 ↑ 6.9	66.9	83.9	66.8	65.7	**61.4**

size objects like bus, construction vehicle, trailer and truck. Although the number of bicycle category objects are less compared to other objects, our model surpassed Object-DGCNN [45] by 6.9% AP, this is possible with the local and global feature extraction in addition to Li3DeTr cross-attention of decoder. The other object categories obtain a competitive performance. We quantize the point clouds in the backbone of our network and downsample the feature maps into multiple strides to increase the receptive field, but that leads to information loss which thereby makes our network difficult to detect smaller objects like *pedestrians* and *barriers*. The design of point cloud backbones in the future, maintaining the original resolution while increasing the receptive field would solve this problem. The performance of our network in terms of AP for each object category on nuScenes *test* dataset is provided in supplementary.

Object distance. The ground-truth 3D bounding boxes are divided into three subsets: `[0m,20m]`, `[20m,30m]` and `[30m,+∞]` basing on the distance between object centers and ego vehicle. The performance of our network in terms of mAP by object distance on the nuScenes [1] dataset compared to CenterPoint [51] and Object-DGCNN [45] is shown in Table 4. Our approach significantly improves the mAP of objects at object distance greater than `30m` compared to CNN-based CenterPoint [51]. Although the LiDAR point cloud is sparse at far distance from ego vehicle, our attention mechanism in encoder and decoder models long range interactions between the sparse points to predict the objects at far distance.

Table 4: Performance of our network in terms of mAP by object distance on the nuScenes *val* set. $*$: MMDetection3D [8] implementation. The scores in green indicate the increase in performance with respect to scores in underline.

Method	`[0m,20m]`	`[20m,30m]`	`[30m,+∞]`
CenterPoint [51] $*$	71.3	51.5	26.5
Obj-DGCNN [45]	73.2	55.5	30.3
Ours	75.6 ↑ 4.3	56.9	32.7 ↑ 6.2

Object size. The ground-truth 3D bounding boxes are divided into two subsets: `[0m, 4m]` and `[4m,+∞]`, basing on the size of the longer edge of the bounding box. The performance of our model in terms of mAP by object size compared to state-of-the-art approaches on the nuScenes [1] dataset is shown in Table 5. Our transformer based approach predominantly improves the mAP of large size objects than small size objects compared to CNN-based CenterPoint [51]. This proves our hypothesis that long range interactions made possible by attention mechanism improves the detection performance.

4.3. Ablation Studies

Attention blocks. The performance of our network in terms of mAP and NDS with different attention operations for self and cross-attention blocks in the decoder (§ 3.3) is shown in Table 6. We test our approach by employing DGCNN [46] similar to [45] and multi-head self-attention

Table 5: Performance of our network in terms of mAP by object size on the nuScenes *val* set. ∗: MMDetection3D [8] implementation. The scores in green indicate the increase in performance with respect to scores in underline.

Method	[0m, 4m]	[4m, +∞]
CenterPoint [51] ∗	34.9	23.5
Obj-DGCNN [45]	36.0	25.4
Ours	37.9 ↑ 3.0	27.8 ↑ 4.3

[41] to model object query interactions and, deformable cross-attention [55] and our Li3DeTr cross-attention to attend the global voxel features. Our Li3DeTr cross-attention shows improved performance compared to deformable cross-attention [55] for both of the self-attention operations. This proves the effectiveness of our Li3DeTr cross-attention block to effectively link the global voxel features with the 3D object predictions.

Table 6: Ablation study with different attention operations in the decoder on the nuScenes *val* set

Self-attention	Cross-attention	mAP	NDS
DGCNN [46]	Deformable attn. [55]	58.6	66.0
	Li3DeTr (ours)	**59.0**	**66.3**
Multi-head self attn. [41]	Deformable attn. [55]	57.9	65.5
	Li3DeTr (ours)	**61.4**	**67.6**

Number of queries. The performance of our network in terms of mAP and NDS on the nuScenes [1] dataset for different number of queries in the decoder is shown in Table 7. The performance of our network in terms of mAP and NDS slightly increases with increase in number of queries because object queries represent the potential positions of objects. However, the performance has slight impact for over 900 queries. So, we fix number of queries to 900.

Table 7: Ablation study on the number of object queries in decoder on the nuScenes *val* set

	Li3DeTr (pillar)		Li3DeTr (voxel)	
No. of queries	mAP	NDS	mAP	NDS
300	51.9	60.4	59.3	65.1
600	52.5	61.4	60.2	66.1
900	**53.8**	**63.0**	**61.4**	**67.6**
1000	53.1	62.5	60.8	67.2

Backbones. The performance of our network in terms of mAP and NDS on nuScenes dataset for different backbones compared with CenterPoint [51] and Object-DGCNN [45] is shown in Table 8. We test our approach with PointPillars [19] with 0.2m voxel size and VoxelNet [53] with 0.1m voxel size for LiDAR point cloud feature extraction. As shown in Table 8, our approach with VoxelNet feature extraction outperforms the network with PointPillars backbone. However, our architecture is flexible to plugin various backbones depending on the specific requirements of different applications.

Table 8: Ablation study on different backbones on the nuScenes *val* set

Backbone	Method	mAP	NDS
PointPillars [19]	CenterPoint [51]	50.3	60.2
	Object-DGCNN [45]	53.2	62.8
	Ours	**53.8**	**63.0**
VoxelNet [53]	CenterPoint [51]	56.4	64.8
	Object-DGCNN [45]	58.6	66.0
	Ours	**61.4**	**67.6**

5. Conclusion

We present an end-to-end, single-stage LiDAR based 3D Detection Transformer (Li3DeTr) architecture which inputs LiDAR point clouds and predicts 3D bounding boxes. Inspired by DETR [2], we formulate our model with set-to-set loss and thereby remove the need for post processing methods like NMS. We introduce a novel Li3DeTr cross-attention block in the decoder head to link the global LiDAR voxel feature maps (obtained from encoder network) and 3D predictions, leveraged by sparse set of object queries learnt from the data. Without bells and whistles, our network archives 61.3% mAP and 67.6% NDS surpassing the state-of-the-art methods on the nuScenes dataset and achieves competitive performance on the KITTI dataset.

Acknowledgment

This work has been supported by European Union's H2020 MSCA-ITN-ACHIEVE with grant agreement No. 765866, Fundação para a Ciência e a Tecnologia (FCT) under the project UIDB/00048/2020 and FCT Portugal PhD research grant with reference 2021.06219.BD.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

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

[3] Yuning Chai, Pei Sun, Jiquan Ngiam, Weiyue Wang, Benjamin Caine, Vijay Vasudevan, Xiao Zhang, and Dragomir Anguelov. To the point: Efficient 3d object detection in the range image with graph convolution kernels. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2021.

[4] Qi Chen, Lin Sun, Ernest Cheung, and Alan L Yuille. Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization. *Advances in Neural Information Processing Systems*, 33:21224–21235, 2020.

[5] Qi Chen, Lin Sun, Zhixin Wang, Kui Jia, and Alan Yuille. Object as hotspots: An anchor-free 3d object detection approach via firing of hotspots. In *European conference on computer vision*, pages 68–84. Springer, 2020.

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

[7] Yilun Chen, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Fast point r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9775–9784, 2019.

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

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[11] Shengheng Deng, Zhihao Liang, Lin Sun, and Kui Jia. Vista: Boosting 3d object detection via dual cross-view spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8448–8457, 2022.

[12] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6569–6578, 2019.

[13] 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.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[15] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9224–9232, 2018.

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

[17] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018.

[18] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[19] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019.

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[23] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. Tanet: Robust 3d object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11677–11684, 2020.

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

[25] 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.

[26] 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.

[27] Xuran Pan, Zhuofan Xia, Shiji Song, Li Erran Li, and Gao Huang. 3d object detection with pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7463–7472, 2021.

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

[29] 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.

[30] 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.

[31] Meytal Rapoport-Lavie and Dan Raviv. It's all around you: Range-guided cylindrical network for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2992–3001, 2021.

[32] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[34] 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.

[35] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020.

[36] 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.

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[38] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016.

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

[40] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

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

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

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

[45] Yue Wang and Justin M Solomon. Object dgcnn: 3d object detection using dynamic graphs. *Advances in Neural Information Processing Systems*, 34, 2021.

[46] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

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

[48] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018.

[49] 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.

[50] Zetong Yang, Yanan Sun, Shu Liu, Xiaoyong Shen, and Jiaya Jia. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1951–1960, 2019.

[51] 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.

[52] Jin Hyeok Yoo, Yecheol Kim, Jisong Kim, and Jun Won Choi. 3d-cvf: Generating joint camera and lidar features using cross-view spatial feature fusion for 3d object detection. In *European Conference on Computer Vision*, pages 720–736. Springer, 2020.

[53] 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.

[54] Xinge Zhu, Yuexin Ma, Tai Wang, Yan Xu, Jianping Shi, and Dahua Lin. Ssn: Shape signature networks for multi-class object detection from point clouds. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020.

[55] 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.