SEFormer: Structure Embedding Transformer for 3D Object Detection

Xiaoyu Feng[1], Heming Du[2], Hehe Fan[3], Yueqi Duan[1], Yongpan Liu[1]

[1]Tsinghua University,
[2]Australian National University,
[3]National University of Singapore
feng-xy18@mails.tsinghua.edu.cn

Abstract

Effectively preserving and encoding structure features from objects in irregular and sparse LiDAR points is a crucial challenge to 3D object detection on the point cloud. Recently, Transformer has demonstrated promising performance on many 2D and even 3D vision tasks. Compared with the fixed and rigid convolution kernels, the self-attention mechanism in Transformer can adaptively exclude the unrelated or noisy points and is thus suitable for preserving the local spatial structure in the irregular LiDAR point cloud. However, Transformer only performs a simple sum on the point features, based on the self-attention mechanism, and all the points share the same transformation for *value*. A such isotropic operation cannot capture the direction-distance-oriented local structure, which is essential for 3D object detection. In this work, we propose a Structure-Embedding transFormer (SEFormer), which can not only preserve the local structure as a traditional Transformer but also have the ability to encode the local structure. Compared to the self-attention mechanism in traditional Transformer, SEFormer learns different feature transformations for *value* points based on the relative directions and distances to the query point. Then we propose a SEFormer-based network for high-performance 3D object detection. Extensive experiments show that the proposed architecture can achieve SOTA results on the Waymo Open Dataset, one of the most significant 3D detection benchmarks for autonomous driving. Specifically, SEFormer achieves 79.02% mAP, which is 1.2% higher than existing works. https://github.com/tdzdog/SEFormer.

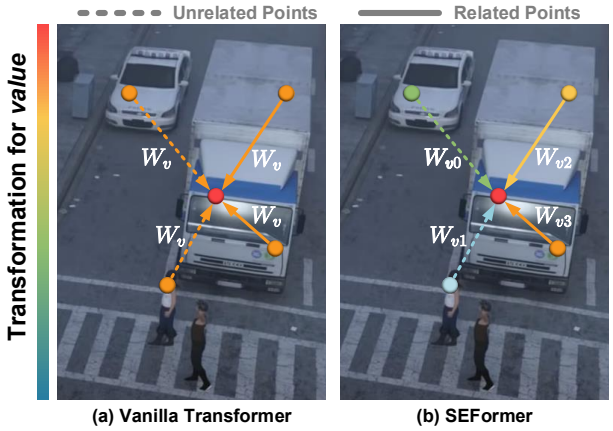


Figure 1: Motivation illustration: (a) The self-attention mechanism in Transformer can adaptively exclude unrelated or noisy points. However, the vanilla Transformer shares the same feature transformation for all *value* points. Such isotropic operation ignores the local structure information between the *key* points and the *query* point. (b) While in SEFormer, we propose to learn different feature transformations for *value* points based on their relative directions and distances to the *query* point. Hence the local structure can be encoded in the Transformer output. For example, SEFormer can differentiate a car's front and tail points.

Introduction

Point cloud-based 3D object detection has attracted more and more attention with the development of autonomous driving and robotics. Due to the lack of texture and color information in the point cloud, 3D object detection highly depends on the structure information of local areas. However, unlike the grid-arranged 2D images, the sparse and irregular nature of LiDAR point clouds makes the local structure often incomplete and noisy. Hence, how to effectively extract the essential structure feature still needs to be solved.

Inspired by the success of 2D object detection (Ren et al. 2015; Wu et al. 2021b,a; Shrivastava, Gupta, and Girshick 2016; Sun et al. 2021b; Chi, Wei, and Hu 2020; Dong et al. 2022), convolution rapidly becomes the mainstream operator in 3D object detection. Traditional convolution-based 3D object detection can be divided into two main trends: point (Qi et al. 2017a,b; Shi, Wang, and Li 2019; Qi et al. 2018; Yang et al. 2020; Chen et al. 2022; He et al. 2022b) and voxel-based solutions (Yan, Mao, and Li 2018; Shi et al. 2020; Deng et al. 2020; Zheng et al. 2021; Lang et al. 2019; Shi et al. 2022; Li et al. 2021b,a; Xu, Zhong, and Neumann 2022).

However, convolution is designed with fixed and rigid kernel sizes and treats all neighboring points equally. Therefore, it inevitably contains unrelated or noisy points from other objects or backgrounds. Recently, Transformer (Vaswani et al. 2017) has shown its effectiveness in 3D vision tasks such as classification, segmentation and object detection (Zheng et al. 2022; Zhao et al. 2021; Wang et al.

2021; Cao et al. 2022). Compared with convolution, the self-attention mechanism in Transformer can adaptively exclude noisy or irrelevant points from other objects. We refer to the such ability of Transformer as **structure preserving**. However, the vanilla Transformer shares the same feature transformation for points *value*. Such isotropic operation ignores the local structure information in the spatial relationships, e.g., directions and distances, from the center point to its neighbors. As illustrated in Fig. 1 (a), the points share the same transformation. If we swap the positions of points, the Transformer output remains the same. It challenges recognizing the object's direction, which is vital to 3D object detection.

In this work, we are motivated by convolution, learning a set of different kernels to embed point features from different distances and directions. Hence, we design a novel Structure-Embedding transFormer (SEFormer), which can encode the direction-distance-oriented local into its output. Compared with the vanilla Transformer, the proposed SEFormer learns different transformations for *value* points from different directions and distances. Hence the change in local spatial structure can be encoded in the output features. We refer to SEFormer's such ability as **structure encoding**. As shown in Fig. 1(b), the points are embedded with different transformations (differently colored arrows). Once the points are swapped, the correspondence between the points and transformations is changed. Hence, the position change can be encoded in the SEFormer output and provide a clue to recognize the object directions accurately. As a Transformer, SEFormer can also adaptively **preserve** the local structure. With the additional structure **encoding** ability, SEFormer can extract better local structure information.

Based on the proposed SEFormer unit, we propose a multi-scale SEFormer network to extract local structure descriptions for 3D object detection. Precisely, we extract *point-* and *object-level* structure features. Based on the multi-scale features extracted by stacked convolution layers, multiple parallel SEFormer blocks, each containing stacked SEFormer units with different neighbor search radii, are utilized to extract richer structure features around each sampled embedding point. Each embedding point provides a point-level structure description of its surrounding local area. Then these point-level embedding features are sent to a SEFormer-based detection head. The network first predicts multiple potential region proposals. Based on stacked SEFormer layers, each proposal integrates its nearby point embedding and outputs an object-level feature embedding. The final bounding boxes are generated based on such object-level embedding. Our main contributions are threefold:

- We propose SEFormer, a new Structure-Embedding transFormer to capture the local point structure. SEFormer can not only preserve the local structure as traditional Transformer but also have additional ability to encode the local direction-distance-oriented structure.
- Based on the proposed SEFormer unit, we design a new multi-scale 3D object detection framework. With multiple SEFormer blocks, we extract point- and object-level structure features for more accurate detection.
- Extensive experiments prove the advantages of our SEFormer. On the Waymo Open dataset, we achieve 79.02% mAP, which is 1.2% higher than existing works.

Related Work

3D Object Detection on Point Cloud. 3D object detection methods on point clouds has made a giant leap recently. According to different input representations, recent research can be categorized into two families: point- and voxel-based.

(1) *Point-based Object Detection*. Many works (Qi et al. 2019; Xie et al. 2021; Chen et al. 2022) propose to directly process raw point cloud data by adopting point-based backbones, such as PointNet (Qi et al. 2017a) and PointNet++ (Qi et al. 2017b). To process a mass of LiDAR points in outdoor environments, *i.e.*, KITTI (Geiger et al. 2013) and Waymo (Sun et al. 2020), previous point-based approaches (Qi et al. 2018; Shi, Wang, and Li 2019; Yang et al. 2019) usually downsample the input point cloud and disturbed by the information loss. (2) *Voxel-based Object Detection*. Voxel-based works (Yan, Mao, and Li 2018; Shi et al. 2020; Deng et al. 2020; Xu, Zhong, and Neumann 2022) transform raw point cloud into compact voxel-grid representation and utilize efficient 3D sparse convolution operator so high-resolution voxelization can be adopted during computation. In this work, we mainly refer to voxel-based ones when talking about convolution-based works.

Transformer for 3D Object Detection. Many researchers are motivated by the recent success of Transformer in point cloud processing (Guo et al. 2021; Fan, Yang, and Kankanhalli 2021; Zhao et al. 2021; Fan, Yang, and Kankanhalli 2022) and have tried to introduce Transformer into 3D object detection. Pointformer (Pan et al. 2021) follows their paradigm and designs three Transformers to extract features from different scales. Voxel Transformer (Mao et al. 2021b) combines Transformer with voxel representation and achieves much higher precision. Fan et al. (2022) propose a single-stride Transformer that improves the detection precision on small objects such as pedestrians and cyclists. However, the vanilla Transformer adopted in such works lacks the ability of local structure encoding. Hence we propose SEFormer to better extract local structure.

Method

In this section, we first introduce the proposed SEFormer unit, including its motivation and corresponding architecture. Then, we present the proposed detection architecture in the following sections.

Structure Embedding Transformer (SEFormer)

Structure Preserving $\&$ **Encoding.** Before introducing SEFormer, we will first state our primary motivation, achieving simultaneous structure-preserving and encoding. Such motivation comes from one critical insight we have on two existing operators, convolution and Transformer.

Convolution is the most famous operator in computer vision tasks because convolution's locality and spatial invariance adapt well to the inductive bias in images. While we propose another essential advantage of convolution is that it

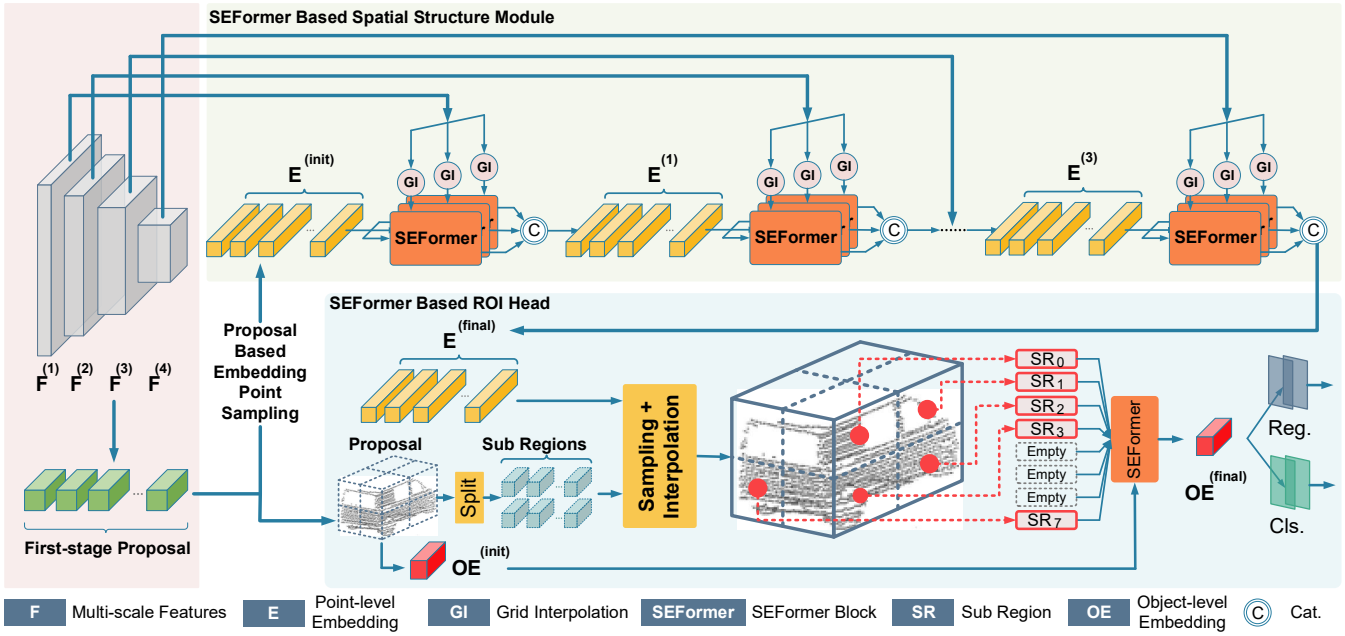


Figure 2: Overview of the proposed multi-scale SEFormer network. Stacked convolution layers are first used to extract multi-scale voxel features. Then a SEFormer-based spatial structure module aggregates the multi-scale features into several point-level embedding features (yellow cube). A SEFormer-based detection head further integrates the point-level embedding with predicted region proposals to generate the object-level embedding features (red cube) for final bounding boxes prediction.

can **encode** the structural information of data. To illustrate such a point, we first formulate convolution as follows:

$$\boldsymbol{f}_{\boldsymbol{p}}^{'} = \sum_{\delta} w_{\delta(\boldsymbol{p})} \cdot \boldsymbol{f}_{\boldsymbol{p}+\delta(\boldsymbol{p})} \tag{1}$$

Here $\boldsymbol{f}, \boldsymbol{f}^{'}$ represents the input and output feature of a convolution layer at center position $\boldsymbol{p}$, while δ denotes the relative position between the neighboring points and the center point. We decompose convolution as a two-step operator, transformation and aggregation. Each point will be multiplied during transformation by its corresponding kernel w_δ. Then these points will be summed with a fixed aggregation coefficient $\alpha = 1$. The kernels are differently learned in convolution based on their directions and distances to the kernel center. Hence convolution can **encode** the local spatial structure into the output. However, in convolution, all neighboring points are equally ($\alpha = 1$) treated during aggregation. The mainstream convolution operator adopts a static and rigid kernel, but the LiDAR point cloud is often irregular and incomplete. Hence convolution inevitably includes irrelevant or noisy points in the output feature.

Compared with convolution, the self-attention mechanism in Transformer provides a more effective method to **preserve** the irregular objects' shapes and boundaries in Point Cloud. For a point cloud with N points, $\boldsymbol{p} = [p_1, \ldots, p_N]$, Transformer computes the response of each point as:

$$\boldsymbol{f}_{p}^{'} = \sum_{\delta} \alpha_{\delta(\boldsymbol{p})} \cdot \boldsymbol{W}^{v} \boldsymbol{f}_{\boldsymbol{p}+\delta(\boldsymbol{p})} \tag{2}$$

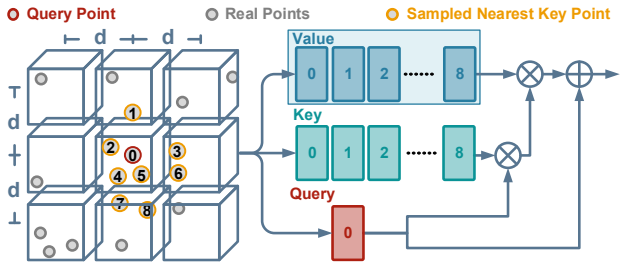
Here α_δ represents the self-attention coefficients among points in the neighboring area while $\boldsymbol{W}^v$ means the value transformation. We can still decompose Eq. 2 into a transformation process with a transformation matrix $\boldsymbol{W}^v$ and an aggregation process with attention coefficients α_δ. The coefficient α_{ij} between point $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ can be calculated as $\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{N} exp(e_{ik})}$. Here $e_{ij} = \frac{(\boldsymbol{W}^q \boldsymbol{f}_i)(\boldsymbol{W}^k \boldsymbol{f}_j)^T}{\sqrt{c}}$ is the scaled dot-produce attention between $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ and $\boldsymbol{W}^q, \boldsymbol{W}^k$ represent the transformation matrix for *query* and *key*. Compared with the static $\alpha = 1$ in convolution, the self-attention coefficients allow the Transformer to adaptively choose the points for aggregation and exclude the influence of unrelated points. We call Transformer's such ability as **structure preserving**. However, according to Eq. 2, the same transformation for *value* is shared among all the points in Transformer. It means that the Transformer misses the **structure encoding** ability, which convolution has.

Given the above discussion, we can find that convolution can **encode** data structure while Transformer can well **preserve** the structure. Hence, the straightforward idea is to design a new operator with both convolution and Transformer advantages. Hence we propose a new Transformer, SEFormer, which can be formulated as follows:

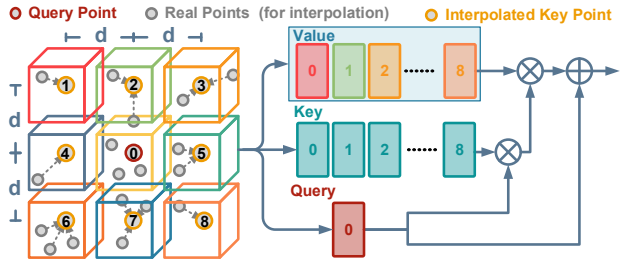
$$\boldsymbol{f}_{\boldsymbol{p}}^{'} = \sum_{\delta} \alpha_{\delta(\boldsymbol{p})} \cdot \boldsymbol{W}_{\delta(\boldsymbol{p})}^{v} \boldsymbol{f}_{\boldsymbol{p}+\delta(\boldsymbol{p})} \tag{3}$$

If we compare Eq. 3 with Eq. 2, we can find the most difference between SEFormer and vanilla Transformer is that different transformations for *value* points are learned based on the relative positions between points.

Architecture of SEFormer. Fig. 3 provides a comparison between the vanilla Transformer and the proposed SE-



(a) Attention of Vanilla Transformer



(b) Attention of SEFormer

Figure 3: Architecture comparison of Transformer and SEFormer. (a) In the vanilla Transformer, nine nearest points (8 key points + 1 center query point) are sampled, and the points share the same transformation for *value*. (b) While in SEFormer, the key points are generated with gird interpolation, and different points have their transformation for *value*.

Former. Given the irregularity of point cloud, we follow the paradigm of Point Transformer (Zhao et al. 2021) to first sample the neighboring points around each query point independently before imported into the Transformer. Unlike the commonly used sampling methods like K Nearest Neighbor (KNN) or Ball Query (BQ), we choose to adopt a grid interpolation to generate the key points. As shown in Fig. 3(b), around the red query point, several (8 in Fig. 3(b)) grid-arranged virtual points are generated. The distance between two grids is a predefined d. Then the virtual points are interpolated with their nearest neighboring points. Compared with traditional sampling like KNN, the advantage of grid sampling is that it can forcibly sample points from different directions. As shown in Fig. 3(a), KNN greedily samples the nearest points. The sampled points show a strong bias from the right corner points. While grid interpolation can avoid such a problem and provide a better description of the local structure. However, grid interpolation uses a fixed grid distance. Hence we adopt a multi-radii strategy in implementation to increase the flexibility of sampling.

During calculation, in Fig. 3 (a), all the anchor points share the same transformation for *key* and *value*. While in Fig. 3 (b), SEFormer constructs a memory pool containing multiple transformation matrices ($\boldsymbol{W}^v$) for *value*. The interpolated key points will be transformed by different $\boldsymbol{W}^v$ based on their relative grid coordinate to the query point. For example, the top left key in Fig. (b) is transformed by the red $\boldsymbol{W}^v$ while the green mW^v transforms the correct key. Hence SEFormer can have the structure encoding ability missed in the vanilla Transformer.

SEFormer Based 3D Object Detection

The whole detection framework is shown in Fig. 2. We first construct a 3D CNN backbone for multi-scale voxel features extraction and initial proposals generation. Then a multi-scale SEFormer network is applied to extract richer local structure features from the voxel features. It contains a SEFormer-based spatial structure module for point-level structure features and a SEFormer-based ROI head for object-level structure features.

3D CNN Backbone. The CNN backbone transforms the input into multiple voxel features with $1\times, 2\times, 4\times$ and $8\times$ downsampling sizes. After the feature extraction, the $8\times$ 3D feature volume will be compressed along the Z-axis and converted into a 2D BEV feature map. Then a center-based approach (Yin, Zhou, and Krahenbuhl 2021) is applied to predict first-stage proposals based on the BEV feature map.

SEFormer Based Spatial Structure module. Then the proposed spatial structure module aggregates the multi-scale features $[\boldsymbol{F}^{(1)}, \boldsymbol{F}^{(2)}, \boldsymbol{F}^{(3)}, \boldsymbol{F}^{(4)}]$ into several point-level embedding $\boldsymbol{E}$. Starting from $\boldsymbol{E}^{init}$, we first integrate the finest-grained features $\boldsymbol{F}^{(1)}$. Each embedding point's corresponding key points are interpolated from $\boldsymbol{F}^{(1)}$. We use m different grid distance d to generate sets of multi-scale key features as $\boldsymbol{F}_1^{(1)}, \boldsymbol{F}_2^{(1)}, \ldots, \boldsymbol{F}_m^{(1)}$. Such a multi-radii strategy can better handle the sparse and irregular point distribution in LiDAR. Then m parallel SEFormer blocks which contain multiple SEFormer units are applied and result in m new embedding $\boldsymbol{E}_1^{(1)}, \boldsymbol{E}_2^{(1)}, \ldots, \boldsymbol{E}_m^{(1)}$. In the end of the block, $\boldsymbol{E}_1^{(1)}, \boldsymbol{E}_2^{(1)}, \ldots, \boldsymbol{E}_m^{(1)}$ are concatenated and transformed into embedding $\boldsymbol{E}^{(1)}$ with a vanilla Transformer. Then $\boldsymbol{E}^{(1)}$ repeats the above process and aggregates $[\boldsymbol{F}^{(2)}, \boldsymbol{F}^{(3)}, \boldsymbol{F}^{(4)}]$ into the final embedding $\boldsymbol{E}^{final}$. Compared with the original voxel features $\boldsymbol{F}$, the embedding $\boldsymbol{E}^{final}$ aggregated contains a richer structural description of the local neighboring area.

SEFormer Based ROI Head. Based on the point-level embeddings $\boldsymbol{E}^{final}$, the proposed head aggregates it into several object-level embeddings to generate final proposals. Specifically, we divide each proposal from the first-stage center-based head into multiple cubic sub-regions and interpolate each sub-region with surrounding point-level embedding features. Due to the sparsity of the point cloud, some sub-regions are often empty. Traditional works simply sum the features from the non-empty parts. However, the car side away from the LiDAR source is often sparse. Hence the relative positions of the empty sub-regions can provide a useful object-level structure feature for direction recognition. In contrast, the proposed SEFormer can utilize such information by embedding both the full and empty sub-regions. As shown in part III of Fig. 2, a SEFormer block takes in both the empty and non-empty sub-regions and integrates their features into a proposal embedding $\boldsymbol{OE}^{final}$. The stronger structure embedding ability of SEFormer can provide a better description of the object-level structure and then generates more accurate 3D proposals.

Methods	LEVEL_1 (IoU=0.7) 3D mAP/mAPH	LEVEL_2 (IoU=0.7) 3D mAP/mAPH	LEVEL_1 3D mAP/mAPH by Distance 0-30m	30-50m	50m-Inf
PointPillars (Lang et al. 2019)*	56.62/-	-/-	81.01/-	51.75/-	27.94/-
MVF (Zhou et al. 2020)	62.93/-	-/-	86.30/-	60.02/-	36.02/-
AFDet (Ge et al. 2020)	63.69/-	-/-	87.38/-	62.19/-	29.27/-
Pillar-OD (Wang et al. 2020)	69.8/-	-/-	88.5/-	66.5/-	42.9/-
CVCNet (Chen et al. 2020)	65.2/-	-/-	86.80/-	62.19/-	29.27/-
SVGA-Net (He et al. 2022b)	73.45/-	66.65/-	92.53	69.44	42.08
VoTr-SSD (Mao et al. 2021b)	68.99/68.39	60.22/59.69	88.18/87.62	66.73/66.05	42.08/41.38
PV-RCNN (Shi et al. 2020)	70.3/69.7	65.4/64.80	91.9/91.3	69.2/68.5	42.2/41.3
VoTr-TSD (Mao et al. 2021b)	74.95/74.25	65.91/65.29	92.28/91.73	73.36/72.56	51.09/50.01
RSN (Sun et al. 2021a) †	75.1/74.6	66.0/65.5	91.8/91.4	73.5/73.1	53.1/52.5
Voxel RCNN (Deng et al. 2020)	75.59/-	66.59/-	92.49/-	74.09/-	53.15/-
SCIR-Net(He et al. 2022c)	75.63/-	66.73/-	92.55/-	72.42/-	-/-
LiDAR RCNN(Li, Wang, and Wang 2021)†	76.0/75.5	68.3/67.9	92.1/91.6	74.6/74.1	54.5/53.4
SST_TS (Fan et al. 2022) †	76.22/75.79	68.04/67.64	-/-	-/-	-/-
CT3D (Sheng et al. 2021)	76.30/-	69.04/-	92.51/-	75.07/-	55.36/-
Pyramid RCNN(Mao et al. 2021a)	76.30/75.68	67.23/66.68	92.67/92.20	74.91/74.21	54.54/53.45
CenterPoint(Yin, Zhou, and Krahenbuhl 2021) †	76.7/68.8	76.2/68.3	-/-	-/-	-/-
PDV (Hu, Kuai, and Waslander 2022)	76.85/76.33	69.30/68.81	93.13/92.71	75.49/74.91	54.75/53.90
Voxel-to-Point(Li et al. 2021b)	77.24/-	69.77/-	**93.23/-**	76.21/-	55.79/-
PV-RCNN++(Shi et al. 2022)	77.32/-	68.62/-	-/-	-/-	-/-
VoxSet (He et al. 2022a)	77.82/-	70.21/-	92.78/-	77.21/-	54.41/-
Ours	**79.02/78.52**	**70.31/69.85**	93.10/**92.66**	**78.07/77.54**	**57.60/56.87**

Table 1: Performance comparison on the Waymo Open Dataset with 202 validation sequences for the 3D vehicle detection. Only one frame is used for training and testing. * is re-implemented by (Zhou et al. 2020). 20% training data are used for most methods. While † denotes methods that use the whole 100% training dataset.

Experiment

In this work, we mainly evaluate the proposed SEFormer on Waymo. Because its large data scale can provide much more convincing evaluation than other benchmarks. We will first introduce Waymo and describe the details of our implementation. Then we will compare with state-of-the-art works on Waymo Open and provide an ablation analysis for the proposed method.

Implementation Details

Waymo Open Dataset. The Waymo dataset contains 1000 LiDAR sequences in total. These sequences are further split into 798 training sequences (including around $158k$ LiDAR samples) and 202 validation sequences (including around $40k$ LiDAR samples). Waymo provides object annotations in the entire 360° field. Its Official evaluation metrics include the standard 3D mean Average Precision (mAP) and mAP weighted by heading accuracy (mAPH). In this work, we present such two metrics mainly from two aspects: difficulty levels and object distance. For the first way, the ground-truth boxes are divided into two groups: LEVEL_1 (number of points is more than five) and LEVEL_2 (the box contains at least one point). For the second way, apart from the overall mAP, we will also show respective mAP for objects located in $0-30m$, $30-50m$, and $>50m$.

Network Architecture. First, the points within the range of $[-75.2,\ 75.2]m$, $[-75.2, 75.2]m$, and $[-2, 4]m$ for the X, Y, and Z-axis are extracted. Then they are voxelized with a $(0.05m, 0.05m, 0.1m)$ step. Our first-stage convolution backbone and the BEV neck follow the same architecture in (Yan, Mao, and Li 2018). The 3D backbone transforms the input into $1\times, 2\times, 4\times$ and $8\times$ downsampled voxel volumes with $16, 32, 64, 64$ dimensions respectively. In the SEFormer based spatial structure module, 4096 query points are selected for each scene. In the SEFormer head, each proposal is divided into $6 \times 6 \times 6$ sub-regions.

Training and Inference. We use 4 RTX 3090 GPUs to train the entire network with batch size 8. We keep most training and inference hyper-parameters same with existing works(Mao et al. 2021a; Shi et al. 2022; Deng et al. 2020; Mao et al. 2021b; Sheng et al. 2021) for a fair comparison. We adopt AdamW optimizer and one-cycle policy(Smith and Topin 2019) with division factor 10 and momentum ranges from 0.95 to 0.85 to train the model. The learning rate is initialized with 0.003. The training time is 40 epochs. Given the large scale of Waymo dataset, we uniformly use 20% training samples for training but use the whole validation set for evaluation.

Detection Results on Waymo Detection Dataset

Fig. 4 illustrates a qualitative presentation of our detection results on Waymo dataset. Table 1 and Table 2 show the 3D and BEV vehicle precision comparison with SOTA works on the Waymo Open Dataset. 0.7 IoU threshold is adopted for both evaluations. We mainly compare the one-frame detection results here.

In Table 1, it can be found further improvement of current 3D object detection has become more and more difficult. However, our SEFormer can still achieve a significant improvement compared with prior works. For the commonly

Diff.	Methods	Overall	0-30m	30-50m	50m-Inf
LV_1	PointPillars(2019)*	75.57	92.10	74.06	55.47
	MVF(2020)	80.40	93.59	79.21	63.09
	Pillar-OD (2020)	87.11	95.78	84.87	72.12
	PV-RCNN (2020)	82.96	97.35	82.99	64.97
	SVGA-Net (2022b)	83.52	97.60	83.14	64.52
	Voxel RCNN (2020)	88.19	97.62	87.34	77.70
	SCIR-Net (2022c)	88.45	97.71	88.41	-
	LiDAR RCNN (2021)	90.1	97.0	89.5	78.9
	Voxel-to-Point(2021b)	88.93	98.05	88.25	79.19
	VoxSet(2022a)	90.31	96.11	88.12	77.98
	Ours	**91.73**	**98.13**	**91.23**	**82.12**
LV_2	PV-RCNN (2020)	77.45	94.64	80.39	55.39
	SVGA-Net (2022b)	80.97	95.54	81.58	60.18
	Voxel RCNN (2020)	81.07	96.99	81.37	63.26
	SCIR-Net (2022c)	81.65	96.88	81.34	-
	LiDAR RCNN (2021)	81.7	94.3	82.3	65.8
	Voxel-to-Point(2021b)	82.18	97.48	82.51	64.86
	VoxSet(2022a)	80.56	96.79	80.44	62.37
	Ours	**85.18**	**97.55**	**85.99**	**69.48**

Table 2: Comparison of BEV vehicle detection on the WOD with 202 validation sequences. * is re-implemented by (Zhou et al. 2020). We only use 20% training data.

used LEVEL_1 mAP/mAPH, we achieve $79.02\%/78.52\%$, which exceeds state-of-the-art works by 1.2% for the LEVEL_1 mAP. For the difficult LEVEL_2 result, we can still get the SOTA results and achieve $70.31\%/69.85\%$ for mAP/mAPH. Such results demonstrate the effectiveness of the proposed SEFormer. To evaluate the influence of object distance, we also provide the range-based LEVEL_1 mAP/mAPH. Although we show lower precision on near objects ($< 30m$), 93.10% mAP vs 93.23% mAP, our improvement for distant objects is much more significant. The improvement for $30-50m$ and $50m-Inf$ targets achieves 0.86% and 3.19% mAP respectively. In most cases, the distant objects are often sparse and only show part of the outline of the objects, which makes extracting useful structure information much more difficult. While SEFormer's structure **preserving** and **encoding** ability alleviates such problem.

In Table 2, SEFormer also outperforms prior works on BEV precision. 91.73% LEVEL_1 and 85.18% LEVEL_2 BEV mAP are achieved. It can be found that the LEVEL_2 improvement is higher. Compared with LEVEL_1, LEVEL_2 contains objects that have very few points. Hence such BEV results also support the above claim that SEFormer has more advantages for sparse objects.

Ablation Study

Comparison between vanilla Transformer and SEFormer. In this work, we propose a new Transformer, SEFormer, to encode the local spatial structure. Hence we compare the performance of the vanilla Transformer and the proposed SEFormer in Table 3. The *T* and *S* represent vanilla Transformer and out SEFormer respectively. In this work, we propose a SEFormer based *spatial structure module* (SSM) and a SEFormer based *head* to extract point- and object-level structure features. Hence we use Transformer based SSM and head as the baseline. It can be found

SSM	Head	LV_1 (IoU=0.7) 3D mAP/mAPH	LV_2 (IoU=0.7) 3D mAP/mAPH
T	T	76.10/75.61	68.24/67.78
S	T	77.54/77.05	68.82/68.38
S	S	**79.02/78.52**	**70.31/69.85**

Table 3: Comparison between vanilla Transformer and SEFormer. Here *SSM* and *Head* respectively denote the spatial structure module and the detection head while *T* and *S* represent vanilla Transformer and SEFormer respectively.

Block Num (m)	LV_1 (IoU=0.7) 3D mAP/mAPH	LV_2 (IoU=0.7) 3D mAP/mAPH
1	78.76/78.25	70.08/69.62
2	**79.02/78.52**	**70.31/69.85**
3	78.81/78.32	70.00/69.55

Table 4: Effects of the number of parallel SEFormer blocks.

vanilla Transformer only achieves $76.10\%/75.61\%$ and $68.24\%/67.78\%$ for LEVEL_1 and LEVEL_2 mAP/mAPH. Replacing Transformer in SSM with SEFormer improves the performance to $77.54\%/77.05\%$ LEVEL_1 mAP/mAPH and $68.82\%/68.38\%$ LEVEL_2 mAP/mAPH. If we further replace the Transformer in the head with SEFormer, we can further get 1.48% LEVEL_1 mAP and 1.49% LEVEL_2 mAP improvement respectively. The results illustrate that the proposed SEFormer has a better ability to capture the structural features of local areas than Transformer. In most Transformer based works, relative position encoding is often used as a method to introduce the relative spatial relationship. Hence we use relative position encoding in both the Transformer based baseline and our SEFormer for a fair comparison. Hence the improvement of SEFormer over Transformer shows that simple relative position encoding cannot fully encode the structure information.

Effect of the number of parallel SEFormer blocks. As noted in Section , multiple parallel SEFormer blocks with different search radii are established. Hence, we investigate the effects of the number of parallel SEFormer blocks in Table 4. In implementation, the parallel SEFormer blocks have gradually doubled search radii. While we set the initial radii as 0.4/0.8/1.2/2.4 for the multi-scale features. According to the results, it can be found that 2 parallel SEFormer blocks achieve the best performance. Increasing or decreasing the block number causes about 0.2% LEVEL_1 mAP reduction.

Effect of the number of heads. Multi-head Transformer often has better performance than single-head Transformer. Hence, we provide an investigation of the effects of head number in Table 5. It can be found that single-head SEFormer achieves $78.87\%/78.37\%$ and $70.14\%/69.69\%$ for LEVEL_1 and LEVEL_2 mAP/mAPH respectively. Adopting double-head SEFormer can reach $79.02\%/78.52\%$ LEVEL_1 and $70.31\%/69.85\%$ LEVEL_2 mAP/mAPH. But the results reduce if we further increase the head number. Hence we choose head number $h = 2$ in this work.

Effect of multi-scale features. Table 6 demonstrates the effects of using multi-scale features. Only using the single-scale feature (conv1) only achieves 78.54% and 69.94%

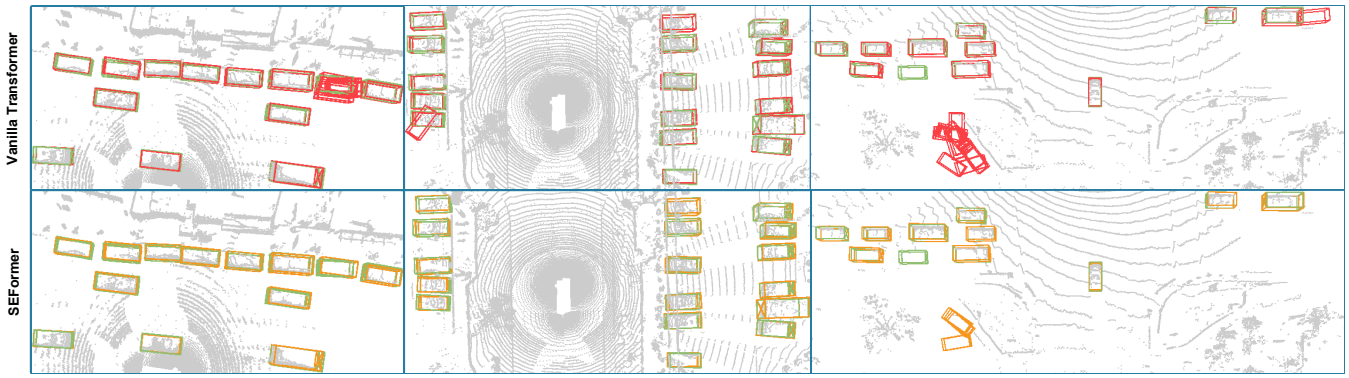


Figure 4: Qualitative visualization on WOD. The green boxes denote the groundtruth.

Head Num (h)	LV_1 (IoU=0.7) 3D mAP/mAPH	LV_2 (IoU=0.7) 3D mAP/mAPH
1	78.87/78.37	70.14/69.69
2	**79.02/78.52**	**70.31/69.85**
4	79.00/78.51	70.30/69.85

Table 5: Effects of the head number in SEFormer.

conv1	conv2	conv3	conv4	LV_1 (IoU=0.7) 3D mAP/mAPH	LV_2 (IoU=0.7) 3D mAP/mAPH
✓				78.54/78.04	69.94/69.49
✓		✓		78.81/78.32	70.16/69.71
✓		✓	✓	**79.02/78.52**	**70.31/69.85**
✓	✓	✓	✓	78.77/78.29	69.96/69.5 2

Table 6: Effects of multi-scale features.

LEVEL 1 and LEVEL 2 mAP. Introducing more features of different scales gradually improves the performance while using features of all 4 scales reduces the precision to some extent. Please see our supplementary material for more results.

Effect of grid interpolation. Table 7 illustrates the effects of grid interpolation. For the control group, grid interpolation is replaced with random sampling within a radius. Please see our supplementary material for more results and discussions.

Structure of the spatial structure module. In the spatial structure module, we aggregate the multi-scale features one by one. While multiple SEFormer blocks with different radii are adopted to extract structure information from one feature. To show the effects of such strategy, we design three different structures for the spatial structure module, fully parallel, full chained, and half parallel half chained. Fig. 5 illustrates the difference among such three structures. Half parallel half chained denotes the structure used in this work. Their effects on model performance are shown in Table 8. It can be found that the half parallel half chained structure has better results than others.

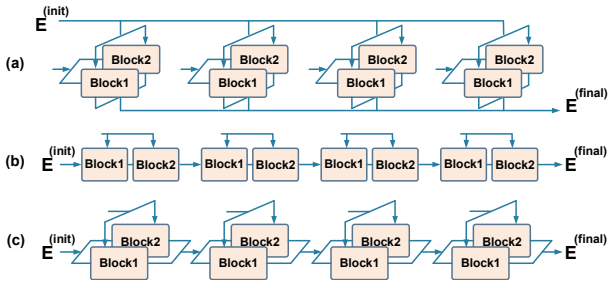


Figure 5: Illustration of (a) fully parallel (b) fully chained and (c) half parallel half chained spatial structure module.

	LV_1 (IoU=0.7) 3D mAP/mAPH	LV_2 (IoU=0.7) 3D mAP/mAPH
w/o GI	78.63/78.14	69.83/69.38
w/ GI	**79.02/78.52**	**70.31/69.85**

Table 7: Effects of grid interpolation.

SSM Structure	LEVEL_1 (IoU=0.7) 3D mAP/mAPH	LEVEL_2 (IoU=0.7) 3D mAP/mAPH
(a)	78.70/78.23	70.04/69.56
(b)	78.86/78.39	70.12/69.68
(c)	**79.02/78.52**	**70.31/69.85**

Table 8: Comparison among different SSM structures.

Conclusion

This work proposes a new Transformer, SEFormer. In vanilla Transformer, all the value points share the same transformation. Hence it lacks the ability to encode the distance-direction-oriented local spatial structure. To solve such problem, SEFormer learns different transforms for value points based on their relative directions and distances to the center query point. Based on the proposed SEFormer, we establish a new 3D detection network including a SEFormer based spatial structure module to extract point-level structure information and a SEFormer based head to capture object-level structure features. Compared with state-of-the-art solutions, our SEFormer achieves higher detection precision on the Waymo Open dataset.

Acknowledgements

This work was partly supported by the National Key Research and Development Program of China under Grant 2021YFB3200903, in part by the National Natural Science Foundation of China under Grant 61934005, in part by the Natural Science Foundation of Beijing Municipality under Grant L211005.

References

Cao, X.; Yuan, P.; Feng, B.; and Niu, K. 2022. CF-DETR: Coarse-to-Fine Transformers for End-to-End Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1): 185–193.

Chen, C.; Chen, Z.; Zhang, J.; and Tao, D. 2022. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, volume 1.

Chen, Q.; Sun, L.; Cheung, E.; and Yuille, A. L. 2020. Every view counts: Cross-view consistency in 3d object detection with hybrid-cylindrical-spherical voxelization. *Advances in Neural Information Processing Systems*.

Chi, C.; Wei, F.; and Hu, H. 2020. Relationnet++: Bridging visual representations for object detection via transformer decoder. *Advances in Neural Information Processing Systems*, 33: 13564–13574.

Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; and Li, H. 2020. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. *arXiv preprint arXiv:2012.15712*.

Dong, J.; Huang, Y.; Zhang, S.; Chen, S.; and Zheng, N. 2022. Construct Effective Geometry Aware Feature Pyramid Network for Multi-Scale Object Detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1).

Fan, H.; Yang, Y.; and Kankanhalli, M. 2022. Point spatio-temporal transformer networks for point cloud video modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 2181–2192.

Fan, H.; Yang, Y.; and Kankanhalli, M. S. 2021. Point 4D Transformer Networks for Spatio-Temporal Modeling in Point Cloud Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14204–14213.

Fan, L.; Pang, Z.; Zhang, T.; Wang, Y.-X.; Zhao, H.; Wang, F.; Wang, N.; and Zhang, Z. 2022. Embracing single stride 3d object detector with sparse transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8458–8468.

Ge, R.; Ding, Z.; Hu, Y.; Wang, Y.; Chen, S.; Huang, L.; and Li, Y. 2020. Afdet: Anchor free one stage 3d object detection. *arXiv preprint arXiv:2006.12671*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11): 1231–1237.

Guo, M.; Cai, J.; Liu, Z.; Mu, T.; Martin, R. R.; and Hu, S. 2021. PCT: Point cloud transformer. *Comput. Vis. Media*, 7(2): 187–199.

He, C.; Li, R.; Li, S.; and Zhang, L. 2022a. 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*, 8417–8427.

He, Q.; Wang, Z.; Zeng, H.; Zeng, Y.; and Liu, Y. 2022b. Svga-net: Sparse voxel-graph attention network for 3d object detection from point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 870–878.

He, Q.; Zeng, H.; Zeng, Y.; and Liu, Y. 2022c. SCIR-Net: Structured Color Image Representation Based 3D Object Detection Network from Point Clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 4486–4494.

Hu, J. S.; Kuai, T.; and Waslander, S. L. 2022. Point density-aware voxels for lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8469–8478.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.

Li, J.; Dai, H.; Shao, L.; and Ding, Y. 2021a. Anchor-free 3d single stage detector with mask-guided attention for point cloud. In *Proceedings of the 29th ACM International Conference on Multimedia*, 553–562.

Li, J.; Dai, H.; Shao, L.; and Ding, Y. 2021b. From voxel to point: Iou-guided 3d object detection for point cloud with voxel-to-point decoder. In *Proceedings of the 29th ACM International Conference on Multimedia*, 4622–4631.

Li, Z.; Wang, F.; and Wang, N. 2021. LiDAR R-CNN: An Efficient and Universal 3D Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7546–7555.

Mao, J.; Niu, M.; Bai, H.; Liang, X.; Xu, H.; and Xu, C. 2021a. Pyramid r-cnn: Towards better performance and adaptability for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2723–2732.

Mao, J.; Xue, Y.; Niu, M.; Bai, H.; Feng, J.; Liang, X.; Xu, H.; and Xu, C. 2021b. Voxel Transformer for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3164–3173.

Pan, X.; Xia, Z.; Song, S.; Li, L. E.; and Huang, G. 2021. 3d object detection with pointformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7463–7472.

Qi, C. R.; Litany, O.; He, K.; and Guibas, L. J. 2019. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, 9277–9286.

Qi, C. R.; Liu, W.; Wu, C.; Su, H.; and Guibas, L. J. 2018. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 918–927.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Sheng, H.; Cai, S.; Liu, Y.; Deng, B.; Huang, J.; Hua, X.-S.; and Zhao, M.-J. 2021. Improving 3D Object Detection with Channel-wise Transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2743–2752.

Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; and Li, H. 2020. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10529–10538.

Shi, S.; Jiang, L.; Deng, J.; Wang, Z.; Guo, C.; Shi, J.; Wang, X.; and Li, H. 2022. PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3D object detection. *arXiv preprint arXiv:2102.00463*.

Shi, S.; Wang, X.; and Li, H. 2019. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 770–779.

Shrivastava, A.; Gupta, A.; and Girshick, R. 2016. Training region-based object detectors with online hard example mining. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 761–769.

Smith, L. N.; and Topin, N. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, 1100612. International Society for Optics and Photonics.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

Sun, P.; Wang, W.; Chai, Y.; Elsayed, G.; Bewley, A.; Zhang, X.; Sminchisescu, C.; and Anguelov, D. 2021a. Rsn: Range sparse net for efficient, accurate lidar 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5725–5734.

Sun, P.; Zhang, R.; Jiang, Y.; Kong, T.; Xu, C.; Zhan, W.; Tomizuka, M.; Li, L.; Yuan, Z.; Wang, C.; et al. 2021b. Sparse r-cnn: End-to-end object detection with learnable proposals. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14454–14463.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Wang, Y.; Fathi, A.; Kundu, A.; Ross, D. A.; Pantofaru, C.; Funkhouser, T.; and Solomon, J. 2020. Pillar-based object detection for autonomous driving. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, 18–34. Springer.

Wang, Y.; Zhang, X.; Yang, T.; and Sun, J. 2021. Anchor DETR: Query Design for Transformer-Based Object Detection. *arXiv preprint arXiv:2109.07107*.

Wu, A.; Han, Y.; Zhu, L.; and Yang, Y. 2021a. Instance-invariant domain adaptive object detection via progressive disentanglement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wu, A.; Liu, R.; Han, Y.; Zhu, L.; and Yang, Y. 2021b. Vector-decomposed disentanglement for domain-invariant object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9342–9351.

Xie, Q.; Lai, Y.-K.; Wu, J.; Wang, Z.; Lu, D.; Wei, M.; and Wang, J. 2021. VENet: Voting Enhancement Network for 3D Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3712–3721.

Xu, Q.; Zhong, Y.; and Neumann, U. 2022. Behind the curtain: Learning occluded shapes for 3D object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 2893–2901.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11040–11048.

Yang, Z.; Sun, Y.; Liu, S.; Shen, X.; and Jia, J. 2019. Std: Sparse-to-dense 3d object detector for point cloud. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1951–1960.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11784–11793.

Zhao, H.; Jiang, L.; Jia, J.; Torr, P. H.; and Koltun, V. 2021. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 16259–16268.

Zheng, W.; Tang, W.; Jiang, L.; and Fu, C.-W. 2021. SE-SSD: Self-Ensembling Single-Stage Object Detector From Point Cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14494–14503.

Zheng, Y.; Duan, Y.; Lu, J.; Zhou, J.; and Tian, Q. 2022. HyperDet3D: Learning a Scene-conditioned 3D Object Detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5585–5594.

Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; and Vasudevan, V. 2020. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, 923–932. PMLR.