ReAGFormer: Reaggregation Transformer with Affine Group Features for 3D Object Detection

Chenguang Lu[1], Kang Yue[2,1], and Yue Liu[1]✉

[1] Beijing Engineering Research Center of Mixed Reality and Advanced Display, School of Optics and Photonics, Beijing Institute of Technology, Beijing, China
chenguang.lu@outlook.com, liuyue@bit.edu.cn
[2] Institute of Software, Chinese Academy of Sciences, Beijing, China
einhep@gmail.com

Abstract. Direct detection of 3D objects from point clouds is a challenging task due to sparsity and irregularity of point clouds. To capture point features from the raw point clouds for 3D object detection, most previous researches utilize PointNet and its variants as the feature learning backbone and have seen encouraging results. However, these methods capture point features independently without modeling the interaction between points, and simple symmetric functions cannot adequately aggregate local contextual features, which are vital for 3D object recognition. To address such limitations, we propose ReAGFormer, a reaggregation Transformer backbone with affine group features for point feature learning in 3D object detection, which can capture the dependencies between points on the aligned group feature space while retaining the flexible receptive fields. The key idea of ReAGFormer is to alleviate the perturbation of the point feature space by affine transformation and extract the dependencies between points using self-attention, while reaggregating the local point set features with the learned attention. Moreover, we also design multi-scale connections in the feature propagation layer to reduce the geometric information loss caused by point sampling and interpolation. Experimental results show that by equipping our method as the backbone for existing 3D object detectors, significant improvements and state-of-the-art performance are achieved over original models on SUN RGB-D and ScanNet V2 benchmarks.

Keywords: 3D object detection · Transformer · Point cloud.

1 Introduction

3D object detection from point clouds is a fundamental task in 3D scene understanding and has wide applications in robotics, augmented reality, etc. However, most of the latest progress in 2D object detection cannot be directly applied to 3D object detection due to the sparsity and irregularity of point clouds.

Prior works first convert the point cloud into the regular data format [1–4] and then use convolutional neural networks for feature extraction and 3D object detection. However, the conversion process always leads to geometric information

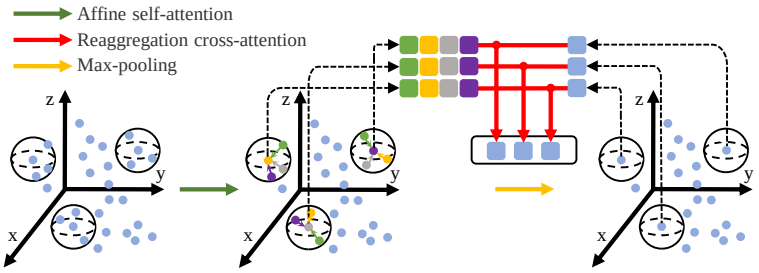

Fig. 1. Illustration of the ReAGF Transformer block. Affine self-attention is introduced to align the group feature space while capturing the dependencies between points within each group. Compared to using only symmetric functions (*e.g.* max), we reaggregate point features via dependencies learned from reaggregation cross-attention, thus improve feature aggregation efficiency.

loss due to quantization errors. PointNet and its variants [5, 6] alleviate this issue by extracting features directly from the raw point clouds, which can preserve the spatial structure and geometric information of the point cloud. As a result, PointNet and its variants are widely used as the feature learning backbone in 3D object detection [7–11]. However, these methods cannot adequately consider the dependencies between points during the capture of point features, and the simple symmetric function (*e.g.* max) cannot effectively utilize the dependencies between points to aggregate local contextual features, which are vital for 3D object detection.

Recently, Transformer [12] has achieved great success in computer vision [13–16]. Thanks to its long-range dependencies modeling capability, the Transformer is an ideal way to address the above limitations. However, how to integrate the advantages of PointNet-like backbone and Transformer to boost 3D object detection is still an open problem. One effort is to combine Transformer with PointNet++ and its variants, such as PCT [17] and PT [18], which focus on the classification and segmentation of point clouds, and the resulting architecture may be suboptimal for other tasks such as 3D object detection. Other solutions introduce the sampling and grouping in PointNet++ into Transformer and design a pure Transformer model for 3D object detection, such as Pointformer [19], but such solutions still use simple symmetric functions (*e.g.* max) to aggregate point features, which limits the representation of the model.

In this paper, we propose a plug-and-play reaggregation Transformer backbone with affine group features for 3D object detection, named as ReAGFormer, which utilizes the ability of the Transformer to model the dependencies between points and reaggregate point features through learned attention, while retaining the flexible receptive fields. Specifically, we propose a reaggregation Transformer block with affine group features (ReAGF Transformer block) to form the downsampling stage of the backbone. As shown in Fig. 1, in the ReAGF Transformer block, we introduce affine self-attention (ASA) to interact on the relationship

between points. ASA first conducts an affine transformation on the features of each intra-group point to eliminate the perturbation of the feature space caused by the sparsity of the point cloud, and align the features of all groups. Then self-attention is employed to capture the relationship between points on the aligned group features. For local aggregation features generated by the symmetric function (*e.g.* max), we model the dependencies between them and the intra-group points by reaggregation cross-attention (RCA), and reaggregate the features by the learned attention. To reduce the geometric information loss caused by point sampling and interpolation, we also use multi-scale connections on the feature propagation layer [6] in the upsampling stage.

To validate the effectiveness and generalization of our method, we replace the backbone of three different state-of-the-art methods of VoteNet [7], BRNet [10] and Group-Free [11] with our proposed ReAGFormer while not changing the other network structures. Experimental results show that when using our proposed ReAGFormer as the feature extraction backbone, all three methods achieve significant improvements, and the modified BRNet and Group-Free achieve state-of-the-art results on ScanNet V2 [20] and SUN RGB-D [21] datasets, respectively.

Our main contributions can be summarized as follows:

- We introduce the reaggregation Transformer block with affine group features (ReAGF Transformer block), which alleviates the perturbation of the local feature space by affine transformation and models the dependencies between points, while reaggregating the point set features with the learned attention.
- Based on ReAGF Transformer block, we build reaggregation Transformer backbone with affine group features, named as ReAGFormer, which can align different groups of feature space and efficiently capture the relationship between points for 3D object detection. Our ReAGFormer can be served as a plug-and-play replacement features learning backbone for 3D object detection.
- Experiments demonstrate the effectiveness and generalization of our backbone network. Our proposed method enables different state-of-the-art methods to achieve significant performance improvements.

2 Related Work

2.1 Point Cloud Representation Learning

Grid-based methods such as projection-based methods [22, 23] and voxel-based methods [2, 24, 25] were frequently used in early point cloud representations. Such methods can effectively solve the problem of difficult point cloud feature extraction caused by irregular point clouds. However, the quantification process in the projection-based methods suffers from information loss, while voxel-based methods require careful consideration of computational effort and memory cost.

Recently, the method of learning features directly from the raw point cloud has received increasing attention. Prior works include MLP-based [5, 6, 26] methods, convolution-based methods [27–33] and graph-based methods [34–37]. As

representative methods, PointNet and its variants [5, 6] are widely used for point feature learning in 3D object detection. However, these methods lack the ability to capture dependencies, in addition, the symmetric function in PointNet and its variants cannot adequately aggregate local point set features. In this work, we address the above limitations with Transformer to boost 3D object detection.

2.2 3D Object Detection in Point Clouds

Due to the sparsity and irregularity of point clouds, early 3D object detection methods usually transformed point clouds into regular data structures. One class of methods[1, 4, 38] projects point cloud to the bird's eye view. Another class of methods[2, 3, 39, 40] converts the point cloud into voxels. There are also methods that use templates [41] or clouds of oriented gradients [42] for 3D object detection.

With the rapid progress of deep learning on point clouds, a series of networks represented by PointNet [5] and PointNet++ [6] that directly processes point clouds are proposed and gradually serve as the backbone of 3D object detectors. PointRCNN [43] introduces a two-stage object detection method that generates 3D proposals directly from the raw point cloud. PV-RCNN [44] combines the advantages of both PointNet++ and voxel-based methods. VoteNet [7] introduces deep hough voting to design an end-to-end 3D object detector, and subsequently derives a series of methods. MLCVNet [8] and HGNet[45] use attention mechanism and hierarchical graph network, respectively, to boost the detection performance. To address the issues of outlier points on detection performance, H3DNet [9] and BRNet [10] introduce hybrid geometric primitives and back-tracing representative points strategy to generate more robust results, respectively. DisARM [46] designs a displacement aware relation module to capture the contextual relationships between carefully selected anchor. In contrast to these methods, we focus on feature learning backbone in 3D object detection. We show that our ReAGFormer can serve as the point feature learning backbone for most of the above methods.

2.3 Transformers in Computer Vision

Transformer [12] has been successfully applied to computer vision and has seen encouraging results in such tasks as image classification [13, 47], detection [14, 48] and segmentation [49, 50]. Transformer is inherently permutation invariant and therefore also well suited for point cloud data. PCT [17] and PT [18] construct transformer on point clouds for classification and segmentation. Stratified Transformer [51] proposes a stratified transformer architecture to capture long-range contexts for point cloud segmentation. DCP [52] is the first method to introduce transformer to the point cloud registration task. In point cloud video understanding, P4Transformer [53] introduces point 4D convolution and transformer to embed local features and capture information about the entire video. Transformer also shows great potential for low-level tasks in point clouds, such as point cloud upsampling [54], denoising [55] and completion [56]. Transformer

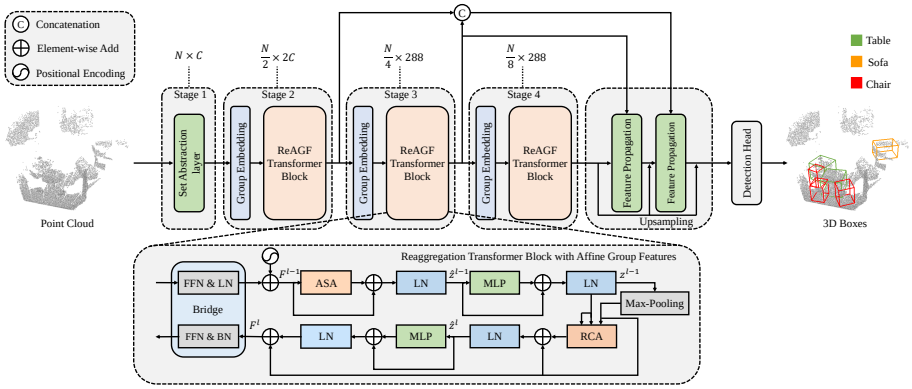

Fig. 2. The architecture of our ReAGFormer backbone for 3D object detection. ReAGFormer has four downsampling stages. The first stage is a set abstraction layer [6], and all other stages consist of group embedding and ReAGF Transformer block. The upsampling stage is a feature propagation layer with multi-scale connection.

is also used for 3D object detection such as Pointformer [19], Group-Free [11] and 3DETR [57]. These methods have seen great progress, however, they neglect the perturbation of the local point set feature space caused by the sparsity and irregularity of point clouds, while still using simple symmetric functions to aggregate point set features. In contrast, we propose a reaggregation Transformer block with affine group features that can alleviate the perturbation of the feature space, while reaggregating the point set features with the learned attention to boost the symmetric function.

3 Proposed Method

In this work, we proposed ReAGFormer, a reaggregation Transformer backbone with affine group features for point feature learning in 3D object detection. As shown in Fig. 2, the proposed ReAGFormer involves four stages of downsampling to generate point sets with different resolutions, and an upsampling stage to recover the number of points. Each downsampling stage involves two main components: group embedding and reaggregation Transformer block with affine group features (ReAGF Transformer block). The group embedding is used to generate the suitable input for the Transformer block. In the ReAGF Transformer block, we introduce affine self-attention (ASA) to apply an affine transformation on the group feature space, while modeling the dependencies between points. For group aggregation point features generated by symmetric functions (*e.g.* max), we introduce reaggregation cross-attention (RCA) to boost the efficiency of symmetric function aggregation by reaggregating group features using the captured dependencies. Moreover, we also utilize feature propagation layers [6] with multi-scale connections in the upsampling stage to reduce the information loss due to point sampling and interpolation. In this section, we describe each part in detail.

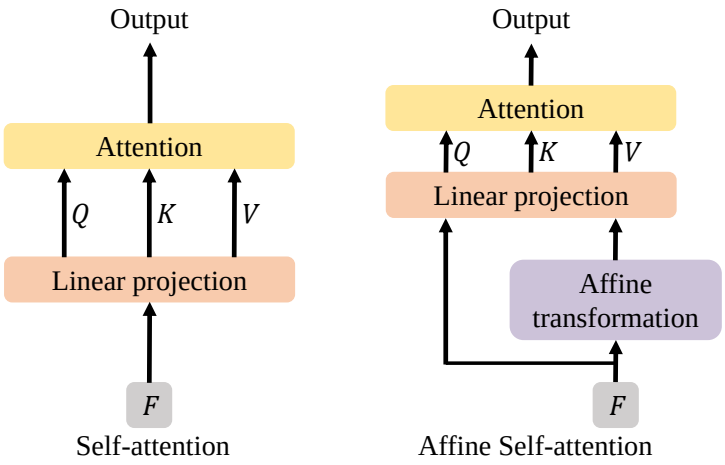

Fig. 3. A comparison between self-attention and our affine self-attention. By a simple affine transformation, we align the feature spaces of different groups and therefore better capture the dependencies between points within a group.

3.1 Group Embedding

The way in which the point cloud data is fed into the ReAGF Transformer block is vital to the overall architecture. To efficiently capture the fine-grained spatial geometric features of the point cloud scene and obtain the flexible receptive fields, we follow the sampling and grouping strategy in PointNet++ [6] to generate local point sets.

Specifically, we use farthest point sampling (FPS) to sample $N^{'}$ points from the input point cloud $P = \{x_i\}_{i=1}^{N}$. Taking each sampling point as the centroid, a ball query is used to generate $N^{'}$ groups according to the specified radius r, in which each group contains k points. Groups are denoted as $\{G_i\}_{i=1}^{N^{'}}$ and feature learning is performed on groups using the shared MLP layer to extract group features $F = \{F_i \in \mathbb{R}^{k \times C}\}_{i=1}^{N^{'}}$, where C is the feature dimension of each point in the group. The group features F is served as the input sequence for the subsequent Transformer block.

3.2 Reaggregation Transformer Block with Affine Group Features

Affine Self-attention (ASA). To extract the dependencies between each intra-group point, we resort to Transformer and self-attention [12]. However, the feature space of each group consisting of intra-group point features may be unaligned due to the sparsity and irregularity of point clouds, as well as the variety of structures within each group, which may lead to perturbations in the group feature space. We argue that the perturbation in the feature space can

limit the modeling of the relationship between points. To alleviate the above problem, we propose the affine self-attention (ASA).

Specifically, we project the group features $F = \{F_i \in \mathbb{R}^{k\times C}\}_{i=1}^{N'}$ generated by the group embedding to query (Q), key (K) and value (V) as the input for the attention calculation. Different from the self-attention, as shown in Fig. 3, to alleviate the perturbation of the group feature space, we apply the affine transformation to the group features that are used to generate key and value, and then perform linear projection and shared self-attention, formulated as:

$$h^s = \text{softmax}(\frac{(W_q^s F)(W_k^s \cdot \text{AT}(F))^T}{\sqrt{d}})(W_v^s \cdot \text{AT}(F)) \ , \tag{1}$$

$$\text{ASA}(F) = [h^0, h^1, ..., h^s]W_o \ , \tag{2}$$

where W_q^s, W_k^s and W_v^s are the projection parameters that generate Q, K and V. s denotes the s-th head. W_o is the projection matrix used to generate the output. $[\cdot]$ is the concatenation and d is the feature dimension of the s-th head. AT is the affine transformation.

For affine transformation (AT) module, inspired by [58], we utilize a simple transformation method. Specifically, for the group feature $F = \{F_i \in \mathbb{R}^{k\times C}\}_{i=1}^{N'}$ generated by the group embedding, we formulate the following operation:

$$\hat{F}_i = \delta(\frac{F_i - f_i}{\sigma + \epsilon}), \quad \sigma = \sqrt{\frac{1}{N' \times k \times C}\sum_{i=1}^{N'}\sum_{j=1}^{k}(F_i - f_i)^2} \ , \tag{3}$$

$$\hat{F} = \{\hat{F}_i\}_{i=1}^{N'} = \text{AT}(F) \ , \tag{4}$$

where $F_i = \{f_j \in \mathbb{R}^C\}_{j=1}^{k}$ is the point features of the i-th group. $f_i \in \mathbb{R}^C$ is centroid feature of the i-th group and $\delta(\cdot)$ is shared MLP. $\hat{F}$ denotes the group feature after affine transformation and ϵ is set to 1e-5 to ensure the correctness of the calculation.

Reaggregation Cross-attention (RCA). Although the simple symmetric function (*e.g.* max) can satisfy the permutation invariance of the point cloud, it cannot utilize dependencies between points to aggregate the local point set features, which limits the representation of the model. To alleviate this issue, we propose the reaggregation cross-attention. Specifically, for the group features F extracted by ASA, we first apply a symmetric function to aggregate the point features of each group, and then perform cross-attention on the aggregated features and their corresponding intra-group point features, which can be formulated as:

$$h^s = \text{softmax}(\frac{(W_q^s \cdot \text{MAX}(F))(W_k^s F)^T}{\sqrt{d}})(W_v^s F) \ , \tag{5}$$

$$\text{RCA}(\text{MAX}(F), F) = [h^0, h^1, ..., h^s]W_o \ , \tag{6}$$

where W_q^s, W_k^s, W_v^s, W_o, d and $[\cdot]$ have the same meaning as Eq. (1) and Eq. (2). MAX denotes the symmetric function and we use the max-pooling.

Based on ASA and RCA, the reaggregation Transformer block with affine group features (ReAGF Transformer block) can be summarized as:

$$\begin{aligned}
\hat{z}^{l-1} &= \mathrm{LN}(\mathrm{ASA}(F^{l-1}) + F^{l-1}) \ , \\
z^{l-1} &= \mathrm{LN}(\mathrm{MLP}(\hat{z}^{l-1}) + \hat{z}^{l-1}) \ , \\
\hat{z}^{l} &= \mathrm{LN}(\mathrm{RCA}(\mathrm{MAX}(z^{l-1}), z^{l-1}) + \mathrm{MAX}(z^{l-1})) \ , \\
F^{l} &= \mathrm{LN}(\mathrm{MLP}(\hat{z}^{l}) + \hat{z}^{l}) + \mathrm{MAX}(z^{l-1}) \ ,
\end{aligned} \tag{7}$$

where ASA and RCA are affine self-attention and reaggregation cross-attention, respectively. F^l and z denote the output group features of stage l and temporary variables, respectively. LN denotes layer normalization.

Normalized Relative Positional Encoding. The positional encoding has a vital role in the Transformer. Since the coordinates of points naturally express positional information, such methods of applying transformer on point clouds as PCT [17] do not use positional encoding. However, we find that adding positional encoding helps to improve detection performance. We argue that the reason is that the detection task requires explicit position information to help with object localization. In this work, we use learnable normalized relative positional encoding. Specifically, we compute the normalized relative position between each intra-group point and the corresponding centroid, then we map it to the group feature dimension by shared MLP layer, which can be formulated as:

$$\mathrm{NRPE} = \mathrm{MLP}(\frac{P_i - p_i}{r}) \ , \tag{8}$$

where $P_i = \{p_j \in \mathbb{R}^3\}_{j=1}^{k}$ denotes the point coordinates of the i-th group. $p_i \in \mathbb{R}^3$ is the centroid coordinates of the i-th group. NRPE is the relative positional encoding and r indicates the radius of the ball query in the group embedding. The positional encoding is added to the input F^{l-1} before performing the attention calculation shown in Eq. (7).

Feature Connection Bridge. The computational process of group embedding and ReAGF Transformer block have a large difference, so their output features may have semantic gap. To connect these two modules more naturally, inspired by [59], we use a simple bridge module, as shown in Fig. 2. Specifically, we use linear projection and normalization between the group embedding and ReAGF Transformer block to eliminate their semantic gap.

3.3 Feature Propagation Layer with Multi-scale Connection

Previous methods [7, 8, 10, 11] often utilize hierarchical feature propagation layers [6] for upsampling of points. Point sampling in the group embedding and interpolation in the feature propagation layer inevitably suffer from information loss. To alleviate the above limitations, inspired by [60, 61], we introduce a multi-scale connection based on the feature propagation layer, as shown in the

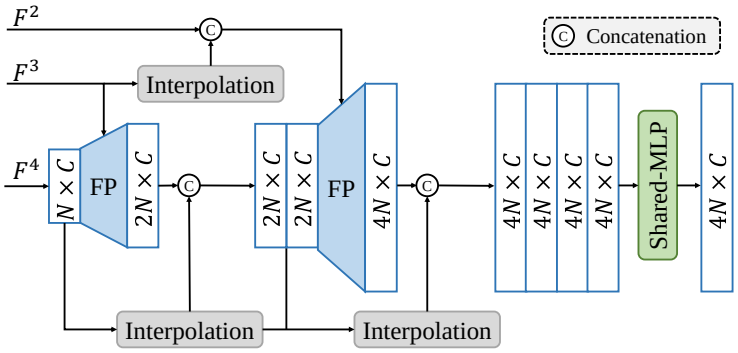

Fig. 4. Upsampling stage consisting of feature propagation (FP) layer with multi-scale connection. F^l denotes output group features of downsampling stage l. N is the number of points and C is the point feature dimension.

Fig. 4. Specifically, in the upsampling stage, we perform point interpolation and feature mapping on the input point features of each layer using the feature propagation layer. The output of all previous layers is concatenated to the output of the current layer, and the concatenated features are served as input to the subsequent layer. Moreover, we fuse the features from the downsampling on the skip connection and concatenate them to the corresponding upsampling layer.

4 Experiments

4.1 Datasets and Evaluation Metrics

We validate our approach using two large-scale indoor scene datasets: SUN RGB-D [21] and ScanNet V2 [20]. SUN RGB-D is a 3D scene understanding dataset with 10,335 monocular RGB-D images and oriented 3D bounding box annotations for 37 categories. We follow VoteNet [7] and divide ∼5K samples for training, while using 10 common categories for evaluation. ScanNet V2 is a large-scale 3D reconstructed indoor dataset consisting of 1513 scenes, containing 18 categories of axis-aligned 3D bounding box annotations, and point clouds obtained from reconstructed mesh. Following the setup of VoteNet, we use about 1.2K training samples. For both datasets, we follow a standard evaluation protocol [7], which is mean Average Precision (mAP) with IoU thresholds of 0.25 and 0.5.

4.2 Implementation Details

We apply our ReAGFormer on three state-of-the-art models (*i.e.* VoteNet [7], BRNet [10], and Group-Free [11]) by replacing the backbone of these models with our ReAGFormer, and the replaced models are named as *ReAGF-VoteNet*, *ReAGF-BRNet*, and *ReAGF-Group-Free*, respectively. The number of input points and the data augmentation follow the corresponding baseline [7, 10,

Table 1. Performance comparison by applying our backbone to state-of-the-art models on SUN RGB-D and ScanNet V2. VoteNet* denotes that the result is implemented in MMDetection3D [63], which has better results than the original paper [7]. For Group-Free, we reports the results for 6-layer decoder and 256 object candidates.

Method	SUN RGB-D		ScanNet V2	
	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
VoteNet* [7]	59.1	35.8	62.9	39.9
+Ours (ReAGF-VoteNet)	**62.3(↑3.2)**	**40.7(↑4.9)**	**66.1(↑3.2)**	**45.4(↑5.5)**
BRNet [10]	61.1	43.7	66.1	50.9
+Ours (ReAGF-BRNet)	**61.5(↑0.4)**	**44.8(↑1.1)**	**67.4(↑1.3)**	**52.2(↑1.3)**
Group-Free [11]	**63.0**	45.2	**67.3**	48.9
+Ours (ReAGF-Group-Free)	62.9(↓0.1)	**45.7(↑0.5)**	67.1(↓0.2)	**50.0(↑1.1)**

11]. Our model is divided into four stages in the downsampling part. Except for stage 1, each stage contains a group embedding and a reaggregation Transformer block with affine group features. Note that we use the standard set abstraction layer [6] in stage 1, because using transformer in the early stages does not help the results. We argue that the point feature extraction is not complete in the shallower layers, and thus the dependencies between points cannot be built effectively. For the upsampling stage, we use 2 feature propagation layers with multi-scale connections. The ball query radius of the group embedding is set to {0.2, 0.4, 0.8, 1.2} and the number of sampling points is {2048, 1024, 512, 256}. The upsampling stage interpolates the points to {512, 1024}. The feature dimension of the points generated by the backbone is set to 288. For ASA and RCA, the number of head is set to 8, and a dropout of 0.1 is used. The initial learning rate of the Transformer block is 1/20 of the other parts, and the model is optimized with the AdamW optimizer [62]. More implementation details are described in the supplementary material.

4.3 Evaluation Results

Evaluation on Different State-of-the-art Models. We apply our proposed ReAGFormer on three existing state-of-the-art models: VoteNet [7], BRNet [10] and Group-Free [11]. We replace the backbone of these three methods with our proposed ReAGFormer and evaluate them on SUN RGB-D and ScanNet V2 datasets, and the results are shown in Table 1. Our proposed ReAGFormer enables all three methods to achieve performance improvements. In particular, ReAGF-VoteNet gets 4.9% and 5.5% improvement on mAP@0.5 on both datasets. Similarly, ReAGF-BRNet outperforms BRNet with gains of 1.1% mAP@0.5 and 1.3% mAP@0.5 on SUN RGB-D and ScanNet V2, respectively. For both datasets, ReAGF-Group-Free also achieves improvement of 0.5% and 1.1% on mAP@0.5, respectively. Note that by applying our approach to the baseline model, the performance improvement on the more challenging mAP@0.5 is better than mAP@0.25, which demonstrates that our ReAGFormer adequately models the interaction between points and improves object localization accuracy.

Table 2. Performance comparison on SUN RGB-D (left) and ScanNet V2 (right). VoteNet* indicates that the resulting implementation is based on the MMDetection3D [63] toolbox, which has better results than the original paper [7]. 4×PointNet++ denotes 4 individual PointNet++. - indicates that the corresponding method does not report results under this condition or dataset. For Group-Free, we report the results for 6-layer decoder and 256 object candidates.

SUN RGB-D	backbone	mAP@0.25	mAP@0.5
VoteNet* [7]	PointNet++	59.1	35.8
MLCVNet [8]	PointNet++	59.8	-
HGNet [45]	GU-Net	61.6	-
SPOT [64]	PointNet++	60.4	36.3
H3DNet 1BB [9]	PointNet++	-	-
H3DNet 4BB [9]	4×PointNet++	60.1	39.0
Pointformer+VoteNet [19]	Pointformer	61.1	36.6
3DETR [57]	PointNet++	59.1	32.7
BRNet [10]	PointNet++	61.1	43.7
Group-Free [11]	PointNet++	**63.0**	45.2
CaVo [65]	U-Net	61.3	44.3
DisARM+VoteNet [46]	PointNet++	61.5	41.3
DisARM+Group-Free [46]	PointNet++	-	-
ReAGF-VoteNet (Ours)	ReAGFormer (Ours)	62.3	40.7
ReAGF-BRNet (Ours)	ReAGFormer (Ours)	61.5	44.8
ReAGF-Group-Free (Ours)	ReAGFormer (Ours)	62.9	**45.7**

ScanNet V2	backbone	mAP@0.25	mAP@0.5
VoteNet* [7]	PointNet++	62.9	39.9
MLCVNet [8]	PointNet++	64.7	42.1
HGNet [45]	GU-Net	61.3	34.4
SPOT [64]	PointNet++	59.8	40.4
H3DNet 1BB [9]	PointNet++	64.4	43.4
H3DNet 4BB [9]	4×PointNet++	67.2	48.1
Pointformer+VoteNet [19]	Pointformer	64.1	42.6
3DETR [57]	PointNet++	65.0	47.0
BRNet [10]	PointNet++	66.1	50.9
Group-Free [11]	PointNet++	67.3	48.9
CaVo[65]	U-Net	-	-
DisARM+VoteNet [46]	PointNet++	66.1	49.7
DisARM+Group-Free [46]	PointNet++	67.0	50.7
ReAGF-VoteNet (Ours)	ReAGFormer (Ours)	66.1	45.4
ReAGF-BRNet (Ours)	ReAGFormer (Ours)	**67.4**	**52.2**
ReAGF-Group-Free (Ours)	ReAGFormer (Ours)	67.1	50.0

Table 3. Ablation study on ASA and RCA of the ReAGF Transformer block. If ASA and RCA are not used, each layer well be a standard set abstraction layer [6].

ASA	RCA	mAP@0.25	mAP@0.5
-	-	64.1	42.6
✓	-	65.0	45.2
-	✓	66.0	44.9
✓	✓	**66.1**	**45.4**

Comparisons with the State-of-the-art Methods. In order to verify the effectiveness of our proposed ReAGFormer, we compare ReAGF-VoteNet, ReAGF-BRNet and ReAGF-Group-Free with previous state-of-the-art methods on SUN RGB-D and ScanNet V2 datasets. Table 2 shows the comparison results. By replacing the original backbone with our ReAGFormer, VoteNet achieves competitive results on both datasets. For ScanNet V2, ReAGF-BRNet achieves 67.4% on mAP@0.25 and 52.2% on mAP@0.5, which outperforms all previous state-of-the-art methods. On the SUN RGB-D dataset, ReAGF-Group-Free achieves 62.9% on mAP@0.25 and 45.7% on mAP@0.5, which is better than previous state-of-the-art methods on the more challenging mAP@0.5.

4.4 Ablation Study

In this section, we conduct ablation experiments to verify the effectiveness of each module. If not specified, the models used in all experiments are trained on ReAGF-VoteNet, and evaluated on ScanNet V2 validation set.

ReAGF Transformer Block. We investigate the effects of the ReAGF Transformer block consisting of ASA and RCA, and the results are summarized in Table 3. If the ReAGF Transformer block consisting of ASA and RCA is not

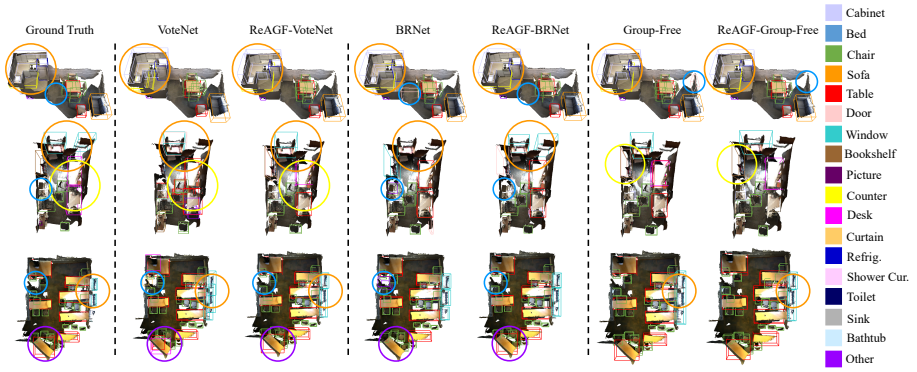

Fig. 5. Qualitative comparison results of 3D object detection on ScanNet V2. ReAGF-VoteNet, ReAGF-BRNet and ReAGF-Group-Free denote the replacement of the baseline original backbone with our ReAGFormer. With the help of ReAGFormer, VoteNet and BRNet achieve more reliable results (orange circles, blue circles and purple circles). Objects with similar shapes (*e.g.* Table and Desk, Bookshelf and Door) can be easily confused, but our method can alleviate such problem (yellow circles). Color is used for better illustration purpose, and it is not used in the experiment. (*Best viewed in color.*)

Table 4. Ablation study on affine transformation of the ASA. If the affine transformation as shown in Eq. (4) is not used, ASA will be the standard self-attention [12].

Downsampling method	Affine transformation	mAP@0.25	mAP@0.5
Set abstraction layer	-	64.1	42.6
	✓	64.6	43.7
ReAGF transformer block	-	65.9	44.5
	✓	**66.1**	**45.4**

used, each layer will be a standard set abstraction layer [6]. We can observe that by applying ASA and RCA separately, performance is improved by 2.6% and 2.3% on mAP@0.5, respectively. If both ASA and RCA are used, we can achieve the best performance improvement. Table 4 ablates the affine transformation (AT) module in the ASA. The best result is achieved using our complete ReAGF transformer block, and our AT also improves the performance of the set abstraction layer, which demonstrates the effectiveness of our AT module.

Positional Encoding. To investigate whether positional encoding is effective and normalized relative positional encoding is better, we conduct comparison without positional encoding and with absolute or normalized relative positional encoding. As shown in Table 5, using normalized relative positional encoding brings 2.1% mAP@0.25 improvement and 0.4% mAP@0.5 improvement compared to not using positional encoding. We argue that the reason is that the detection task requires explicit position information to help object localization. We also find that normalized relative positional encoding outperforms absolute

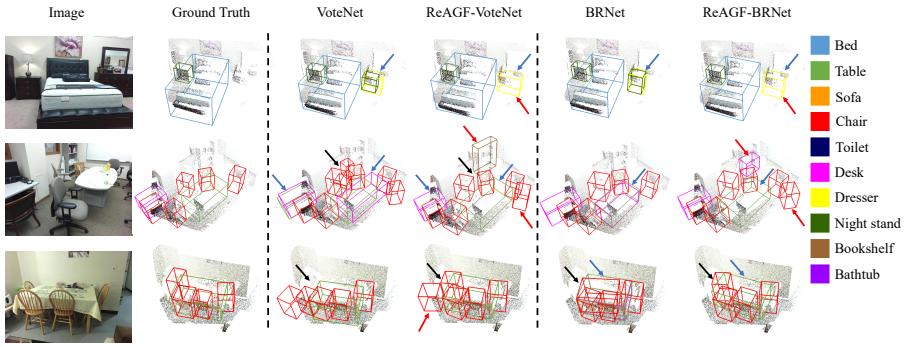

Fig. 6. Qualitative results of 3D object detection on SUN RGB-D. Our method generates more reasonable boxes (see black arrows) and can distinguish between objects with similar shapes (see blue arrows). Moreover, our method can even detect objects that are not annotated in ground truth (see red arrows). Images and colors are only used for better illustration, and they are not used in our network. (*Best viewed in color.*)

Table 5. Ablation study on the effectiveness of positional encoding and performance of different positional encoding.

Positonal encoding	mAP@0.25	mAP@0.5
None	64.0	45.0
Absolute	64.8	43.9
Normalized relative	**66.1**	**45.4**

positional encoding, and even the network without positional encoding is 1.1% better than that using absolute positional encoding on mAP@0.5.

Feature Connection Bridge. In Table 6, we compare the impact of with and without features connection bridge on the 3D object detection performance. With the feature connection bridge, we eliminate the semantic gap between the group embedding and ReAGF Transformer block, thus achieving the improvement of 0.5% on mAP@0.25 and 0.9% on mAP@0.5.

Multi-scale Connection. We investigate the effect of multi-scale connection by replacing it with cascade connection [66, 67] and residual connection [61], and the results are summarized in Table 7. We can see that multi-scale connection achieves the best results compared to the other methods. This demonstrates that multi-scale connection can more fully aggregate multi-scale contextual information and reduce information loss caused by point sampling and interpolation.

4.5 Qualitative Results and Discussion

Fig. 5 illustrates the qualitative comparison of the results on ScanNet V2. These results show that applying our ReAGFormer to the baseline achieves more reli-

Table 6. Ablation study on the feature connection bridge.

Feature connection bridge	mAP@0.25	mAP@0.5
-	65.6	44.5
✓	**66.1**	**45.4**

Table 7. Ablation study on the different connection methods for feature propagation layer.

Connection method	mAP@0.25	mAP@0.5
Cascade	65.2	44.9
Residual	65.0	44.3
Multi-scale	**66.1**	**45.4**

able results. Specifically, ReAGF-VoteNet, ReAGF-BRNet and ReAGF-Group-Free can detect more reasonable and accurate results (orange circles, blue circles and purple circles), despite the challenges of cluttered scenes or fewer points. In addition, our method achieves better results for similarly shaped objects (yellow circles). For example, the desk in the second row of the scene is treated as a table by VoteNet [7] in Fig. 5, but ReAGF-VoteNet successfully detects a desk.

Fig. 6 visualizes the qualitative results on SUN RGB-D scenes. Our model generate more reasonable boxes even in cluttered and occluded scenes (see black arrows). In addition, our method can also better distinguish between similarly shaped objects on SUN RGB-D. For example, in the first row of Fig. 6, we successfully solve the problem of different categories generated by the same object (see blue arrows). In the second row, we can detect the table and the desk correctly (see blue arrows). Besides, our method can even detect objects that are not annotated in the ground truth (see red arrows).

5 Conclusion

In this paper, we present ReAGFormer, a reaggregation Transformer backbone with affine group features for 3D object detection. We introduce affine self-attention to align different groups of feature spaces while modeling the dependencies between points. To improve the efficiency of feature aggregation, we utilize reaggregation cross-attention to reaggregate group features based on learned attention. Moreover, we also introduce a multi-scale connection in the feature propagation layer to reduce the information loss caused by point sampling and interpolation. We apply our ReAGFormer to existing state-of-the-art detectors and achieve significant performance improvements on the main benchmarks. Experiments demonstrate the effectiveness and generalization of our method.

Acknowledgements This work was supported by the Key-Area Research and Development Program of Guangdong Province (No.2019B010139004) and the National Natural Science Foundation Youth Fund No.62007001.

References

1. Chen, X., Ma, H., Wan, J., Li, B., Xia, T.: Multi-view 3d object detection network for autonomous driving. In: CVPR. pp. 1907–1915 (2017)
2. Zhou, Y., Tuzel, O.: Voxelnet: End-to-end learning for point cloud based 3d object detection. In: CVPR. pp. 4490–4499 (2018)
3. Song, S., Xiao, J.: Deep sliding shapes for amodal 3d object detection in rgb-d images. In: CVPR. pp. 808–816 (2016)
4. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: CVPR. pp. 7652–7660 (2018)
5. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR. pp. 652–660 (2017)
6. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS **30**, 5099–5108 (2017)
7. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: ICCV. pp. 9277–9286 (2019)
8. Xie, Q., Lai, Y.K., Wu, J., Wang, Z., Zhang, Y., Xu, K., Wang, J.: Mlcvnet: Multi-level context votenet for 3d object detection. In: CVPR. pp. 10447–10456 (2020)
9. Zhang, Z., Sun, B., Yang, H., Huang, Q.: H3dnet: 3d object detection using hybrid geometric primitives. In: ECCV. pp. 311–329. Springer (2020)
10. Cheng, B., Sheng, L., Shi, S., Yang, M., Xu, D.: Back-tracing representative points for voting-based 3d object detection in point clouds. In: CVPR. pp. 8963–8972 (2021)
11. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3d object detection via transformers. In: ICCV (2021)
12. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV. pp. 213–229. Springer (2020)
15. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. ICCV (2021)
16. Liu, Y., Zhang, Y., Wang, Y., Hou, F., Yuan, J., Tian, J., Zhang, Y., Shi, Z., Fan, J., He, Z.: A survey of visual transformers. arXiv preprint arXiv:2111.06091 (2021)
17. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational Visual Media **7**(2), 187–199 (2021)
18. Zhao, H., Jiang, L., Jia, J., Torr, P., Koltun, V.: Point transformer. arXiv preprint arXiv:2012.09164 (2020)
19. Pan, X., Xia, Z., Song, S., Li, L.E., Huang, G.: 3d object detection with pointformer. In: CVPR. pp. 7463–7472 (2021)
20. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR. pp. 5828–5839 (2017)
21. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: CVPR. pp. 567–576 (2015)
22. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: ICCV. pp. 945–953 (2015)

23. Lang, A.H., Vora, S., Caesar, H., Zhou, L., Yang, J., Beijbom, O.: Pointpillars: Fast encoders for object detection from point clouds. In: CVPR. pp. 12697–12705 (2019)
24. Riegler, G., Osman Ulusoy, A., Geiger, A.: Octnet: Learning deep 3d representations at high resolutions. In: CVPR. pp. 3577–3586 (2017)
25. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: IROS. pp. 922–928 (2015)
26. Jiang, L., Zhao, H., Liu, S., Shen, X., Fu, C.W., Jia, J.: Hierarchical point-edge interaction network for point cloud semantic segmentation. In: ICCV. pp. 10433–10441 (2019)
27. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. NeurIPS **31**, 820–830 (2018)
28. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: ECCV. pp. 87–102 (2018)
29. Wang, S., Suo, S., Ma, W.C., Pokrovsky, A., Urtasun, R.: Deep parametric continuous convolutional neural networks. In: CVPR. pp. 2589–2597 (2018)
30. Wu, W., Qi, Z., Fuxin, L.: Pointconv: Deep convolutional networks on 3d point clouds. In: CVPR. pp. 9621–9630 (2019)
31. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: ICCV. pp. 6411–6420 (2019)
32. Xu, M., Ding, R., Zhao, H., Qi, X.: Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In: CVPR. pp. 3173–3182 (2021)
33. Boulch, A., Puy, G., Marlet, R.: Fkaconv: Feature-kernel alignment for point cloud convolution. In: ACCV (2020)
34. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. Acm Transactions On Graphics (TOG) **38**(5), 1–12 (2019)
35. Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J.: Graph attention convolution for point cloud semantic segmentation. In: CVPR. pp. 10296–10305 (2019)
36. Xu, Q., Sun, X., Wu, C.Y., Wang, P., Neumann, U.: Grid-gcn for fast and scalable point cloud learning. In: CVPR. pp. 5661–5670 (2020)
37. Zhao, H., Jiang, L., Fu, C.W., Jia, J.: Pointweb: Enhancing local neighborhood features for point cloud processing. In: CVPR. pp. 5565–5573 (2019)
38. Liang, M., Yang, B., Wang, S., Urtasun, R.: Deep continuous fusion for multi-sensor 3d object detection. In: ECCV. pp. 641–656 (2018)
39. Hou, J., Dai, A., Nießner, M.: 3d-sis: 3d semantic instance segmentation of rgb-d scans. In: CVPR. pp. 4421–4430 (2019)
40. Yan, Y., Mao, Y., Li, B.: Second: Sparsely embedded convolutional detection. Sensors **18**(10), 3337 (2018)
41. Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J.: Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In: CVPR. pp. 3947–3956 (2019)
42. Ren, Z., Sudderth, E.B.: Three-dimensional object detection and layout prediction using clouds of oriented gradients. In: CVPR. pp. 1525–1533 (2016)
43. Shi, S., Wang, X., Li, H.: Pointrcnn: 3d object proposal generation and detection from point cloud. In: CVPR. pp. 770–779 (2019)
44. Shi, S., Guo, C., Jiang, L., Wang, Z., Shi, J., Wang, X., Li, H.: Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In: CVPR. pp. 10529–10538 (2020)
45. Chen, J., Lei, B., Song, Q., Ying, H., Chen, D.Z., Wu, J.: A hierarchical graph network for 3d object detection on point clouds. In: CVPR. pp. 392–401 (2020)

46. Duan, Y., Zhu, C., Lan, Y., Yi, R., Liu, X., Xu, K.: Disarm: Displacement aware relation module for 3d detection. In: CVPR. pp. 16980–16989 (2022)
47. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357. PMLR (2021)
48. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
49. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H., et al.: Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: CVPR. pp. 6881–6890 (2021)
50. Strudel, R., Garcia, R., Laptev, I., Schmid, C.: Segmenter: Transformer for semantic segmentation. arXiv preprint arXiv:2105.05633 (2021)
51. Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J.: Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8500–8509 (2022)
52. Wang, Y., Solomon, J.M.: Deep closest point: Learning representations for point cloud registration. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3523–3532 (2019)
53. Fan, H., Yang, Y., Kankanhalli, M.: 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. pp. 14204–14213 (2021)
54. Qiu, S., Anwar, S., Barnes, N.: Pu-transformer: Point cloud upsampling transformer. arXiv preprint arXiv:2111.12242 (2021)
55. Xu, X., Geng, G., Cao, X., Li, K., Zhou, M.: Tdnet: transformer-based network for point cloud denoising. Applied Optics **61**(6), C80–C88 (2022)
56. Yu, X., Rao, Y., Wang, Z., Liu, Z., Lu, J., Zhou, J.: Pointr: Diverse point cloud completion with geometry-aware transformers. In: ICCV. pp. 12498–12507 (2021)
57. Misra, I., Girdhar, R., Joulin, A.: An End-to-End Transformer Model for 3D Object Detection. In: ICCV (2021)
58. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. In: ICLR (2021)
59. Peng, Z., Huang, W., Gu, S., Xie, L., Wang, Y., Jiao, J., Ye, Q.: Conformer: Local features coupling global representations for visual recognition. arXiv preprint arXiv:2105.03889 (2021)
60. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017)
61. Chen, B., Liu, Y., Zhang, Z., Lu, G., Zhang, D.: Transattunet: Multi-level attention-guided u-net with transformer for medical image segmentation. arXiv preprint arXiv:2107.05274 (2021)
62. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
63. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d (2020)
64. Du, H., Li, L., Liu, B., Vasconcelos, N.: Spot: Selective point cloud voting for better proposal in point cloud object detection. In: ECCV. pp. 230–247. Springer (2020)
65. You, Y., Ye, Z., Lou, Y., Li, C., Li, Y.L., Ma, L., Wang, W., Lu, C.: Canonical voting: Towards robust oriented bounding box detection in 3d scenes. In: CVPR. pp. 1193–1202 (2022)

66. Qin, X., Zhang, Z., Huang, C., Dehghan, M., Zaiane, O.R., Jagersand, M.: U2-net: Going deeper with nested u-structure for salient object detection. Pattern Recognition **106**, 107404 (2020)
67. Cai, Y., Wang, Y.: Ma-unet: An improved version of unet based on multi-scale and attention mechanism for medical image segmentation. arXiv preprint arXiv:2012.10952 (2020)