AShapeFormer : Semantics-Guided Object-Level Active Shape Encoding for 3D Object Detection via Transformers

Zechuan Li[1], Hongshan Yu[1,2,3*], Zhengeng Yang[1*], Tongjia Chen[1], Naveed Akhtar[4]
[1]Hunan University, [2]National University of Defense Technology
[3]Quanzhou Institute of Industrial Design and Machine Intelligence Innovation, Hunan University
[4]The University of Western Australia
{lizechuan, yuhongshan, yzg050215, tomchen}@hnu.edu.cn; naveed.akhtar@uwa.edu.au

Abstract

3D object detection techniques commonly follow a pipeline that aggregates predicted object central point features to compute candidate points. However, these candidate points contain only positional information, largely ignoring the object-level shape information. This eventually leads to sub-optimal 3D object detection. In this work, we propose AShapeFormer, a semantics-guided object-level shape encoding module for 3D object detection. This is a plug-n-play module that leverages multi-head attention to encode object shape information. We also propose shape tokens and object-scene positional encoding to ensure that the shape information is fully exploited. Moreover, we introduce a semantic guidance sub-module to sample more foreground points and suppress the influence of background points for a better object shape perception. We demonstrate a straightforward enhancement of multiple existing methods with our AShapeFormer. Through extensive experiments on the popular SUN RGB-D and ScanNetV2 dataset, we show that our enhanced models are able to outperform the baselines by a considerable absolute margin of up to 8.1%. Code will be available at https://github.com/ZechuanLi/AShapeFormer

1. Introduction

As an important scene understanding task, 3D object detection [13,20,47] aims to detect 3D bounding boxes and semantic categories in 3D point cloud scenes. It plays an important role in many downstream tasks, such as augmented reality [2, 3], mobile robots [18, 41, 52], and autonomous navigation [1,36,37,39]. Object detection has made significant progress in the 2D domain [15,22,33]. However, owing to the sparse and irregular nature of the point cloud data, 2D detection techniques are generally not readily applicable to the 3D object detection task.

*Corresponding author

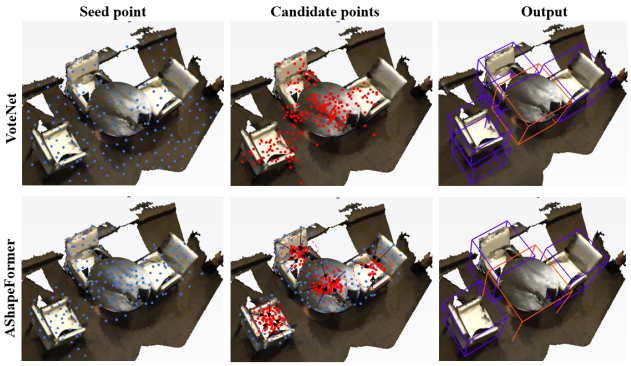

Figure 1. (**Top**) VoteNet [29] seed points contain many background points, leading to sub-optimal candidate points, which are also intrinsically weak as they fail to account for object shape and contour features. (**Bottom**) The proposed AShapeFormer selects more relevant seed points, leading to more appropriate candidates that additionally encode the object shape information actively. This results in high quality 3D object detection.

Inspired by their 2D counterparts, early attempts in 3D object detection, e.g., [16, 39], mapped irregular point clouds to regular 3D voxels, thereafter using 3DCNNs for feature extraction and object detection. However, voxelization inevitably loses fine-grained information of the point clouds, which adversely affects the detection performance. With the advances that allow direct processing of the point clouds with deep neural models, e.g., [30, 31], recent methods aim at directly predicting the 3D bounding boxes from the original unordered point clouds. Among these techniques, VoteNet [29] and its variants [7, 12, 28, 45, 46, 50] have achieved remarkable performance.

These point-wise methods follow a common underlying pipeline which includes first aggregating certain predicted point features into candidate points. The candidate points are later used to estimate the 3D bounding box information, e.g., center, size, and orientation, along with the associated semantic labels. As illustrated in Fig. 1, despite their excellent performance, these methods still face a few major

challenges. (1) The final prediction relies strongly on the quality of the candidate points. However, these points fail to encode important object-level features such as contours and the shape of the 3D objects. (2) The methods must regress over the candidate points, and these points are often influenced by the background points. This propagates the error to cause offsets in the eventual predictions. Current attempts to mitigate these issues rely on generating new features [7, 45] or sampling more points [42]. However, these are resource intensive solutions, which must still rely on the vote point regression quality.

To address the problems, we introduce a novel plug-n-play neural module, named AShapeFormer. It can be easily assembled with many existing 3D object detection methods to provide a considerable performance boost. Our key driving insight is that by utilizing implicit object-level shape features, a detector can be made aware of the object shape distribution. Specifically, our module utilizes multi-head attention to encode the object shape information. We aggregate the object shape features using a self-attention mechanism. Inspired by ViT [10] and BERT [19], we introduce a shape token as the output of the final shape feature to avoid information loss caused by simplistic operations, e.g., pooling. Additionally, we devise a semantic guidance mechanism to sample more foreground points and assign different weights to their features, which improves the shape feature generation. Semantic segmentation scores are also utilized during the aggregation of vote points to reduce the influence of irrelevant vote points and obtain better candidates.

We provide successful demonstration of boosting both point-based [28, 29] and Transformer [23] baselines with our method, achieving strong performance gains. Our experimental results (§ 4.1) show that AShapeFormer boosts the multi-class mean average precision (mAP) up to 3.5% on the challenging SUN RGB-D dataset [38] and 8.1% on the ScanNet V2 dataset [9]. Highlights of our contributions include the following.

- We propose a plug-and-play active shape encoding module named AShapeFormer, which can be combined with many existing 3D object detection networks to achieve a considerable performance boost.
- To the best of our knowledge, our method is the first to combine multi-head attention and semantic guidance to encode strong object shape features for robust classification and accurate bounding box regression.
- We demonstrate a considerable mAP boost on SUN RGB-D (mAP@0.25) and ScanNet V2 datasets by enhancing the state-of-the-art methods with our module.

2. Related Work

Due to its key importance in several downstream tasks, 3D object detection has recently attracted significant interest of the research community. According to the different backbones, indoor 3D object detection methods are mainly divided into the following categories.

1. Voxel-based methods: Projecting the 3D point cloud to a regular 3D voxel representation can resolve the problems caused by the sparse and irregular nature of the point clouds. VoxNet [25] first used 3D convolutional network layers [21] for feature extraction and detection of point clouds exploiting voxelization. 3D-SIS [16] maps 2D image data to voxels, and realizes the fusion of multi-modal data to achieve better object detection and instance segmentation performance at the cost of more complex training process. GSDN [14] employs sparse convolution [48] to improve the efficiency of 3D convolution. Its encoder-decoder structure is built from sparse 3D convolution blocks. FCAF3D [34] adopts the basic architecture of GSDN to improves it as an anchor-free method. This is claimed to improve the efficiency and performance of the original proposal. Although 3D convolution can effectively process point cloud voxels, the process of voxelization inevitably damages the fine-grained information [39] in the point clouds. Moreover, the operation of filling zeros [51] in voxelization also introduces noise, which is detrimental to the detection accuracy.

2. Voting-based methods: With the emergence of methods to directly process 3D point clouds under neural modelling, e.g., PointNet [30] and PointNet++ [31]; it has become viable to directly detect 3D objects in the original point clouds. Numerous point-based detection methods have recently appeared in the literature. Among them, VoteNet [29] has attracted considerable attention for the indoor 3D object detection. It redefines the traditional Hough voting process as an object center point regression problem through MLPs, and generates object proposals by sampling from multiple voting points within the same cluster. MLCVNet [46] introduces a context module based on the VoteNet to learn the contextual information of the scene, which helps in semantic understanding during the detection. H3DNet [50] selects the optimal solution from multiple voting results, and uses multiple geometric primitives to provide more and more accurate constraints to the bounding box. VeNet [45] uses customized modules for the VoteNet before, during and after voting, to gain an accuracy advantage. BRNet [7] proposes a representative point generation module to trace back virtually generated representative points from the voting, so that the network can take some advantage from the object shape.

3. Transformer-based methods: Transformers [40] have achieved excellent performance in the field of NLP [19, 32] and 2D images [4, 10, 17, 27, 43]. They have also gradually started to surface in the techniques for point cloud processing. 3DETR [26] first proposed an end-to-end transformer model for 3D object detection, which almost maintains the vanilla Transformer architecture. Groupfree3D [23] uses a Transformer block to replace the group operation af-

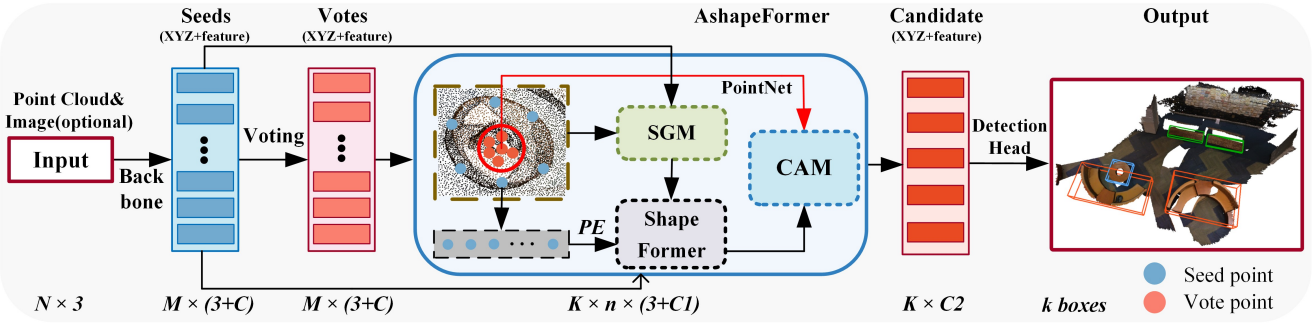

Figure 2. **Schematics of AShapeFormer in action:** Given an input point cloud of N points with XYZ coordinates, and optional corresponding RGB image I, we sample seed points using a backbone. The sampled seed points are processed by a voting module. The seed and vote points are used by our AShapeFormer module. Within AShapeFormer, the Semantics Guided Module (SGM) pays more attention to the foreground points, while Shape Former additionally incorporates object-level shape features with multi-head attention, also employing positional encoding (PE). The Channel Attention Module (CAM) fuses the shape information enriched features with candidate features to provide candidates to a detection head that predicts the bounding boxes.

ter voting. It extracts context information adaptively, and realizes the interaction between candidate-candidate and candidate-original point clouds. TokenFusion [44] proposes a fusion method of multi-modal data using Transformers, which can be used in a variety of visual tasks. Although the Transformer-based methods have achieved good performance in 3D object detection, their convergence is problematic. This is because these methods use complex networks where the attention module requires a large number of input point clouds, which considerably adds to the computing and memory requirements of the training process.

Object shape awareness: Besides the three categories of methods discussed above, the dimension of object shape awareness in the existing literature also relates to our central idea. We can find instances where approaches implicitly or explicitly aim to perceive the object shape to achieve performance gains. For example, RBGNet [42] generates a certain number of rays centered on the voting point and uses the category of the points on the rays to perceive the surface geometry. However, a large number of manual parameters and feature extraction branches increase the complexity of the network and require a very long training time. BRNet [7] generatively backtracks the points to capture local structural features in the original point cloud. It essentially converts VoteNet into a two-stage detection method. HGNet [6] perceives the local shape information by modeling the relative position between the point clouds. However, object-level shape information is not used in the method, which still hinders accurate 3D bounding box estimation.

3. Proposed Approach

This section describes the technical details of our approach Active Shape Encoding via TransFormer (AShapeFormer). An overview of the approach is first provided in § 3.1. In § 3.2 to § 3.5, we elaborate on the components and the learning objective of our technique. To discuss the method, we adopt a VoteNet [29] based backbone to extract features and sampling seed points for the AShapeFormer, and also use its detection head to output the 3D bounding boxes. However, it is emphasized that our central module is easily assembled with other detectors, e.g., [23, 28], due to the plug-n-play nature of AShapeFormer.

3.1. Overview

As illustrated in Fig. 2, the input to our method is a 3D point cloud P. It can optionally also contain the RGB image I of the scene. First, a PointNet++ [31] backbone is used to sample seed points. If the input data includes images, we optionally add a 2D backbone. The backbone extracts input features while sampling the seed points, and then obtains vote points through a voting step. The seed and vote points are fed to the AShapeFormer module. There are three main sub-modules in AShapeFormer, namely Semantics Guided Module (SGM), Channel Attention Module (CAM), and ShapeFormer. To better encode the object-level shape information, SGM gives more attention to the foreground points when sampling the seed points, and adaptively weights different features. ShapeFormer encodes the object-level shape features through a multi-head attention mechanism. CAM adaptively fuses shape features and the candidate features. Finally, the features enriched with object-level shape information are fed to the detection head to provide the 3D bounding boxes.

3.2. Object-Level Shape Encoding

In this section, we introduce a naive shape encoding method and propose Channel Attention Module (CAM) to fuse the object-level shape feature and candidate feature. Indoor 3D object detection generally follows a pipeline that aggregates some predicted object centre features into candidate point features. Let us denote a set of seed points as $\{s_i\}_{i=1}^{M}$, where $s_i = [x_i; f_i]$, with $x_i \in \mathbb{R}^3$ and $f_i \in \mathbb{R}^3$, and a set of vote points as $\{v_i\}_{i=1}^{M}$, where $v_i = [y_i; g_i] \in$

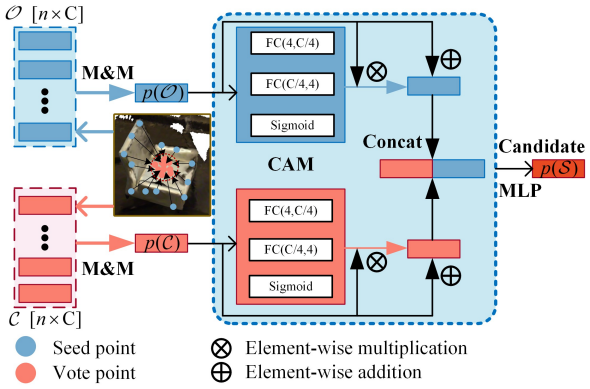

Figure 3. **Object-Level Shape Encoding without SGM and ShapeFormer. Notations:** M&M denotes MLP followed by Max-pooling. FC is fully connected layer.

$\mathbb{R}^{3+C}$. Here, x_i and y_i are the coordinates of the seed and vote points, respectively; and the vote point y_i is the center point of the object where x_i location is predicted by the voting module. The f_i and g_i denote the features of the seed and vote points, respectively.

In the pipeline, K candidate points $\{v_j\}_{j=1}^{K}$ are obtained by sampling from M vote points $\{v_i\}_{i=1}^{M}$. Local grouping based on ball query [31] centered on $\{v_j\}_{j=1}^{K}$ is done to obtain a set of candidate clusters $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_K\}$. As shown in Fig. 3, each cluster $\mathcal{C} = \{v_j^1, v_j^2, \ldots, v_j^n\}$ gets a single vector representation as

$$p(\mathcal{C}) = \text{MaxPool}\left(\text{MLP}\left(\mathcal{C}\right)\right), \tag{1}$$

where $p(\mathcal{C})$ is the candidate feature. A few existing methods [28, 29] directly feed $p(\mathcal{C})$ to the detection head to predict the 3D bounding box. However, these center point features do not particularly help the network to perceive object-level information, which is sub-optimal. To sidestep the issue, BRNet [7] and RBGNet [42] aim at enabling object shape perception by generating pseudo-representative points. This improves the detection performance, ascertaining the importance of shape information in the task. However, these operations not only require a large number of hyper-parameters, but also ignore an important fact that as compared to the pseudo-representative point sets, real seed point clusters can describe the shape information of objects more accurately. This is a key insight used in our method. As illustrated in Fig. 3, we encode the object-level shape feature using the seed points located on the same object.

The vote points belonging to the same object are much more compact than the seed points. Hence, using indices of the candidate local grouping is beneficial for grouping seed points into cluster $\mathcal{O} = \{s_j^1, s_j^2, \ldots, s_j^n\}$, which belong to the same object. To get a single vector representation of the object shape $p(\mathcal{O})$, a naive approach can be similar to the candidate point local grouping, i.e., using MLP and Max-pooling

$$p(\mathcal{O}) = \text{MaxPool}\left(\text{MLP}\left(\mathcal{O}\right)\right). \tag{2}$$

We consider the above naive option at this stage to emphasize on the plug-n-play nature of our module as well as to keep the flow of discussion. We eventually use a sophisticated 'ShapeFormer' sub-module § 3.3 to account for the object shape. In any case, candidate features $p(\mathcal{C})$ and object-level shape features $p(\mathcal{O})$ are distributed in different feature spaces. Hence, in order to fuse these features, we propose the Channel Attention Module (CAM). As shown in Fig. 3, CAM consists of fully connected layers with sigmoid activations. It adaptively learns weights for the features in different spaces such that they seamlessly fuse in the later processing. The candidate and shape features are combined through CAM as follows

$$p(\mathcal{S}) = \text{Concat}\left(\text{CAM}(p(\mathcal{C}), \text{CAM}\left(p(\mathcal{O})\right)\right), \tag{3}$$

where $p(\mathcal{S})$ is the new candidate feature, which is enriched with object-level shape information. The $p(\mathcal{S})$ is eventually fed to a detection head to generate the 3D bounding boxes. We note that our experimental results (§ 4.3) also find the naive object-level shape encoding method proposed in this section reasonably effective. However, the explicit contribution of shape information goes beyond that. In our experience, the naive method still suffers from two main problems. (1) It experiences loss of fine-grained information in the interaction of shape key points. (2) It is marred by interference of the background points. These problems are resolved with the use of the proposed ShapeFormer and the Semantic Guided Module (SGM), discussed below.

3.3. ShapeFormer

Multi-head attention [4, 19, 40] is known for its ability to model contextual information. Its permutation invariant properties are also ideal for point cloud processing. However, its high computational requirement hinders its use for point clouds. We circumvent this issue by dealing with object specific points in our approach.

We introduce a shape encoding module based on a multi-head attention, named ShapeFormer - see Fig. 4. As compared to the naive shape encoding method discussed in the preceding section, ShapeFormer allows stronger shape encoding. In a naive construction, one gets a single vector representation of the object shape with an operation like max pooling, inevitably losing fine-grained information [35]. Addressing that, we prepend a learnable embedding shape token to the sequence of point features, whose state at the output of the ShapeFormer serves as the object-level shape feature. Specifically, given the seed points cluster on the same object (shape key points) $\mathcal{O} = \{s_j^1, s_j^2, \ldots, s_j^n\}$, its corresponding features are $\mathcal{F} = \{f_j^1, f_j^2 \ldots, f_j^n\}$. The input $\mathbf{z^{(0)}}$ of ShapeFormer is

$$\mathbf{z^{(0)}} = \left[f_j^0; \mathcal{F}\right] + \text{PE}(\mathcal{O}), \tag{4}$$

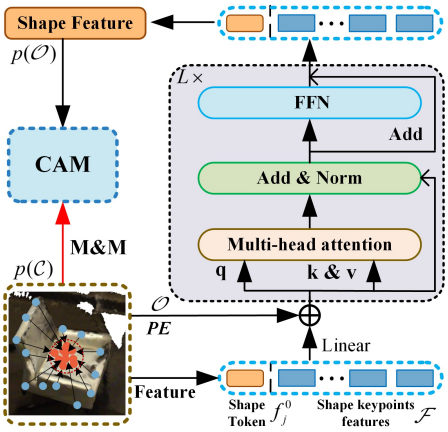

Figure 4. **ShapeFormer:** A shape feature is encoded into shape token through self-attention, avoiding information loss caused by common strategies, e.g., pooling. $\bigoplus$ is element-wise addition.

where f_j^0 is the shape token and $\mathrm{PE}(\cdot)$ is the positional encoding function - explained in detail shortly. The input $\mathbf{z}$ is operated on by learnable weight matrices W_q, W_k and W_v to obtain query q, key k and value v as

$$q = zW_q, k = zW_k, v = zW_v. \tag{5}$$

The attention module performs the following computation.

$$y' = \sum \mathrm{softmax}\left(\frac{qk^\intercal}{\sqrt{d}}\right) v, \tag{6}$$

$$y = \mathrm{Concat}\left(y'^{(0)}, y'^{(1)}, \ldots, y'^{(m-1)}\right), \tag{7}$$

where y' is the output of single attention head and m is the number of the attention heads. Subsequently, it performs

$$\mathbf{o} = \mathcal{A}\left(\mathcal{F}\left(\mathcal{A}\left(y\right)\right)\right), \tag{8}$$

where $\mathcal{A}(\cdot)$ denotes add and normalization operations and $\mathcal{F}(\cdot)$ denotes a FFN with two linear layers with ReLU activation. The calculations expressed as Eq. (6)-(8) comprise one layer of ShapeFormer. The output of the l^{th} layer is

$$\mathbf{o}^l = \left[f_j^{0(l)}, f_j^{1(l)}, f_j^{2(l)}, \ldots, f_j^{n(l)}\right], \tag{9}$$

The shape token $f_j^{0(l)}$ is fed to an MLP to get the shape feature $p(\mathcal{O}) : p(\mathcal{O}) = \mathrm{MLP}\left(f_j^{0(l)}\right)$.

Object-Scene Positional Encoding. Learned positional encoding can benefit transformer based modules [19, 27, 32, 40]. In ShapeFormer, we must pay more attention to the relative positional relationships between the shape key points and the object centers. These relations allow us to encode shape information at object level. Therefore, we propose Object-Scene Positional Encoding, which not only encodes the absolute position of the point cloud, but also

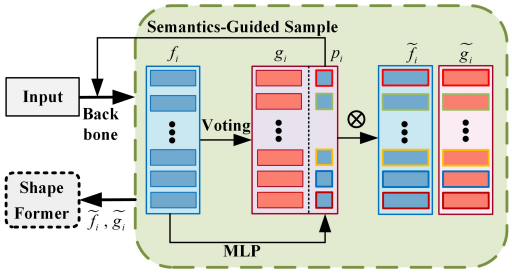

Figure 5. **Semantics Guided Module:** Semantic scores assign different weights to different points and guide sampling in the backbone. $\bigotimes$ denotes element-wise multiplication.

enables object-level positional encoding. Specifically, our positional encoding consists of two components

$$\mathrm{PE} = \mathrm{PE}_s + \mathrm{PE}_o, \tag{10}$$

where PE_s is Scene-level positional encoding and PE_o is Object-level encoding. We compute these as

$$\mathrm{PE}_s = \mathrm{MLP}\left(\left[s_j^c; s_j^1, s_j^2 \ldots, s_j^n\right]\right), \tag{11}$$

$$\mathrm{PE}_o = \mathrm{MLP}\left(\left[\mathbf{0}; s_j^0 - s_j^c, s_j^1 - s_j^c \ldots, s_j^n - s_j^c\right]\right), \tag{12}$$

where s_j^c denotes the candidate coordinates, s_j^n represents the shape key point coordinates. $\mathbf{0}$ is a zero vector with the same dimension as s.

3.4. Semantics Guided Module

The background points in a scene affect local feature extraction [5, 11, 49], which inevitably affects the shape encoding adversely through the shape key points. Although foreground points are selected for voting during training, we cannot supervise them during testing. Therefore, we further improve our AShapeFormer with a Semantics Guided Module (SGM) to alleviate the influence of background points in shape features. As illustrated in Fig. 5, we feed the seed point features f_i to MLP layers to predict the point cloud semantic segmentation score, which is proportional to the probability that the point cloud belongs to the foreground points [5]. If the point cloud lies within the ground truth range of the object bounding box, we consider the point cloud as the foreground point, otherwise it belongs to the background. We compute the foreground confidence $p_i \in [0, 1]$ as

$$p_i = \sigma\left(\mathrm{MLP}_s\left(f_i\right)\right), \tag{13}$$

where MLP_s represents multiple MLP layers and $\sigma(\cdot)$ is the sigmoid activation. Binary cross-entropy loss function is used for the semantic segmentation. The segmentation score is used as a weight to adjust the contribution of different seed points on the shape feature as

$$\widetilde{f}_i = p_i \otimes f_i, \tag{14}$$

where $\otimes$ denotes element-wise multiplication. The re-weighted features $\widetilde{f}_i$ are fed to the ShapeFormer module for

shape encoding. We use the same strategy to re-weight the vote features for better vote aggregation. The SGM module helps AShapeFormer reduce or even eliminate the influence of background points on object shape perception, and better use the features of foreground points to encode more accurate shape information.

Through the backbone, due to the guidance of semantic information, we can sample more seed points without worrying about the background points. Our experiments (§ 4.3) ascertain that more foreground seed points and vote points are obtained through SGM.

3.5. Network Loss

We train the entire network end-to-end using the loss function of the newly proposed AShapeFormer, defined as

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{sgm}} + \lambda_2 \mathcal{L}_{\text{vote}} + \lambda_3 \mathcal{L}_{\text{box}} + \lambda_4 \mathcal{L}_{\text{cls}} + \lambda_5 \mathcal{L}_{\text{obj}}, \quad (15)$$

where λ's are the balancing factors. $\mathcal{L}_{\text{sgm}}$ is used to supervise the foreground/background seed points prediction in SGM, which we define as follows

$$\mathcal{L}_{\text{sgm}} = -\frac{1}{M} \sum_{i=1}^{M} \left[\widehat{p_i} \ln (p_i) + (1 - \widehat{p_i}) \ln (1 - p_i) \right], \quad (16)$$

where p_i and $\widehat{p_i}$ denote the predicted segmentation score and the ground-truth score (1 for foreground and 0 for background). M is the total number of input points.

The loss terms $\mathcal{L}_{\text{vote}}$, $\mathcal{L}_{\text{obj}}$, $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{box}}$ in Eq. (15) indicate the per-point vote regression loss, the objectness loss, the object classification loss, and the bounding box loss, respectively. These loss terms are inspired by the label assignment strategy of VoteNet [29]. We provide details of these losses along with balancing factor settings in the supplementary material.

4. Experiments

4.1. Setup and Implementation Details

Due to its plug-n-play nature, the proposed AShapeFormer can be assembled to several backbones. In our experiments, we show its assembly with VoteNet [29], imVoteNet [28] , RBGNet [42] and GroupFree3D [23]. To generate foreground/background labels for the sample points, we regard all the points within the labeled 3D bounding boxes as foreground points, and the points outside the boxes as the background points. We optimize the networks using the Adam algorithm, which is trained on an RTX 3090 GPU with batch size of 8. We set the initial learning rate to 0.004 when training on the SUNRGBD dataset and 0.008 when training on the Scannet dataset, and decay it by 0.1 in the steps of [120, 140, 180]. We train the network from scratch for a total of 200 epochs. We also implement our method on the 3D object detection toolbox MMdetection3D [8]. Our implementation is consistent with the MMdetection3D framework, and uses the Adamw [24] algorithm. More training details are provided in the supplementary material.

We evaluate the performance of the proposed AShapeFormer on two popular datasets of indoor scenes, namely; ScanNet dataset [9] and SUN RGB-D dataset [38].

4.2. Comparisons with State-of-the-art

We compare our method with the existing state-of-the-art on ScanNet V2 and SUN RGB-D dataset, considering methods such as VoteNet [29] imVoteNet [28], BRNet [7], GroupFree3D [23], TokenFusion [44] etc.

Quantitative comparison. The results on SUN RGB-D are summarized in Table 1. As a plug-n-play module, our enhancements outperform the baselines by remarkable performance gains. For instance, we achieve an absolute gain of more than 3.5% and 1.7% for VoteNet [29] and imVoteNet [28], respectively. Note that, considering the plug-n-play nature of our contribution, mAP@0.25 is a very challenging metric because 3D object detectors have already achieved remarkable performance on this metric. It is hard to achieve a high gain under this metric. In particular, our AShapeFormer applied to imVoteNet* [28] achieves 65.8% on map@0.25, which outperforms all the existing methods. This ascertains that our method is effective even for highly optimized techniques. Table 2 summarizes the results on ScanNetV2. Taking VoteNet [29] as the baseline, our method achieves remarkable 4.5% and 8.1% improvements at mAP@0.25 and mAP@0.5, respectively. Applying AShapeFormer to the more recent Transformer method GroupFree3D [23] and RBGNet [42] also has a significant improvement.

Qualitative comparison. In Fig. 6 and Fig. 7, we visualize the representative 3D object detection results from our method and the baseline methods. These results demonstrate that applying our method to the baseline detector achieves more reliable detection results with more accurate bounding boxes and orientations. As compared to the baselines, our method can discover more missing objects. For example, in Fig. 6 upper left corner, VoteNet misses the challenging object, which is discovered by enhancing it with AShapeFormer. Our method also eliminates false positives, e.g., the results in the second row of Fig. 6 show that there are three chairs around each table. VoteNet detects 5 chairs, whereas our enhancement results are consistent with the ground truth. Figure. 7 shows the visualization results of imVoteNet+AShapeformer on the SUN RGBD dataset. The second and third column both show the ability of our method to eliminate false positives as result of using more foreground point information in the process. In the first column, our method also detects two desks that are not labeled in the scene. This implies that the indicator AP score, might actually underestimate the performance of AShapeFormer. We only provide representative examples in Fig. 6 and 7 to

Model	mAP@0.25	bed	table	sofa	chair	toilet	desk	dresser	nightstand	shelf	bathtub
VoteNet [29]	57.7	83.0	47.3	64.0	75.3	90.1	22.0	29.8	62.2	28.8	74.4
VoteNet∗ [29]	59.7	84.8	49.6	67.8	77.6	87.4	24.3	29.3	61.9	32.1	82.1
BRNet [7]	61.1	86.9	51.8	66.4	77.4	91.3	29.6	35.9	65.9	29.7	76.2
Groupfree3D [23]	63.0	87.8	53.8	70.0	79.4	91.1	32.6	36.0	66.7	32.5	80.0
imVoteNet [28]	63.4	87.6	51.1	70.7	76.7	90.5	28.7	41.4	69.9	41.3	75.9
imVoteNet∗ [28]	64.5	88.5	51.6	73.2	79.2	90.2	30.9	38.0	67.3	46.4	79.7
RBGNet [42]	64.1	88.4	54.5	71.0	82.7	91.3	32.1	38.7	66.7	34.5	80.6
FCAF3D [34]	64.2	88.3	53.0	69.7	81.1	91.3	34.0	40.1	71.9	33.0	79.0
TokenFusion [44]	64.9	-	-	-	-	-	-	-	-	-	-
DisARM [11]	65.3	87.5	52.7	74.1	80.7	91.6	33.3	39.8	69.5	43.7	79.9
Ours (VoteNet)	**61.2(+3.5)**	86.9	51.5	67.8	78.8	91.2	29.0	33.6	65.0	31.3	76.6
Ours (VoteNet∗)	**62.2(+2.5)**	86.9	51.3	69.3	78.9	90.2	28.2	34.6	65.9	35.6	80.7
Ours (imVoteNet)	**65.1(+1.7)**	89.2	53.7	72.9	78.3	90.8	30.2	43.2	70.0	46.5	76.1
Ours (imVoteNet∗)	**65.8(+1.3)**	87.6	55.2	72.8	80.9	92.5	31.2	45.8	67.7	43.7	80.9

Table 1. 3D object detection results on SUN RGB-D validation set with mAP@0.25. ∗ denotes that the model is implemented on MMDetection3D. Ours ($\mathcal{M}$) denotes that $\mathcal{M}$ is enhanced with our AShapeFormer. Our enhancement enables considerable performance gain despite the highly competitive performance of existing methods on mAP@0.25. Best results in each column are green highlighted.

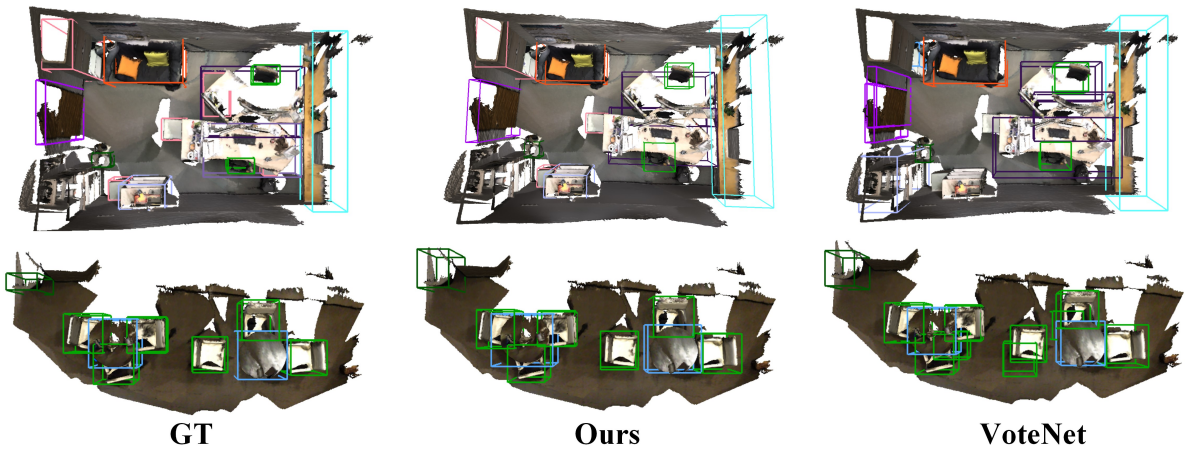

Figure 6. Representative qualitative results on ScanNet V2 dataset [9]. As compared to the baseline, i.e., VoteNet [29], AShapeFormer enhancement not only enables detection of more challenging objects, but also reduces false positive detections. Best viewed on screen.

ScanNet V2	mAP@0.25	mAP@0.5
HGNet [6]	61.3	34.4
VoteNet [29]	58.6	33.5
VoteNet∗ [29]	63.8	44.2
MLCVNet [46]	64.5	41.4
3DETR [26]	65.0	47.0
GroupFree3D [23]	69.1	52.8
RBGNet [42]	70.6	55.2
Ours (VoteNet)	**63.1(+4.5)**	**41.6(+8.1)**
Ours (VoteNet∗)	**66.6(+2.8)**	**47.8(+3.6)**
Ours (GroupFree3D)	**70.4(+1.3)**	**53.4(+0.6)**
Ours (RBGNet)	**71.1(+0.5)**	**56.6(+1.4)**

Table 2. 3D object detection results on ScanNet V2 validation set. A consistent improvement is achieved by enhancing existing methods with the AShapeFormer module.

illustrate important points. More representative results are also provided in the supplementary material.

4.3. Ablation Study and Discussion

We conduct an extensive ablation study to analyze the efficacy of different sub-modules of our method. Table 3 compares the detection results of the naive method (§ 3.2), ShapeFormer (§ 3.3) and SGM (§ 3.4) combined with the vanilla VoteNet on the SUN RGB-D dataset when the IOU is 0.25. It can be seen that when the naive shape encoding is used (without Shapeformer and SGM) there is only incremental performance improvement because the background points are mixed in the encoding, and the pooling layer loses shape information. The SGM utilizes guidance of semantic information to not only sample more foreground points, but also further suppress the contribution of the background points in shape encoding. With the help of SGM, we can better encode the shape feature of the object. Hence, as compared to the original VoteNet, absolute performance gain is 1.8%. The ShapeFormer is a standalone enhancement that does not require Naive baseline. It gives better

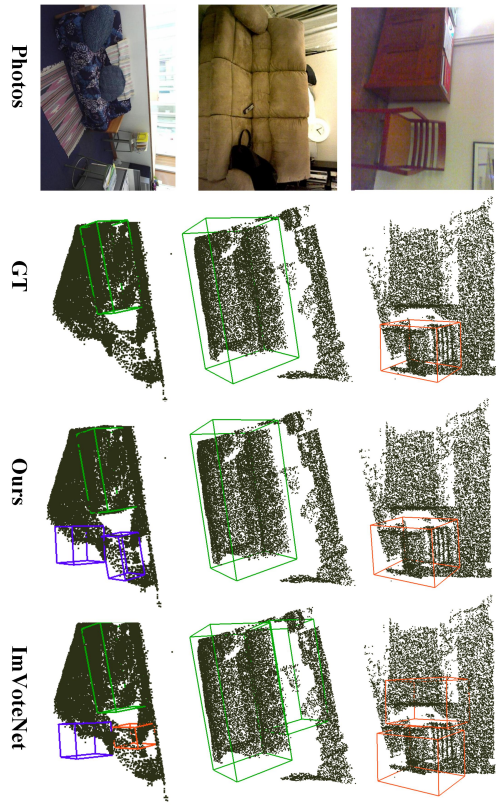

Figure 7. Qualitative results on SUN RGB-D dataset [38]. Best viewed on screen. Our method often correctly detects those objects for which ground truth annotation is not provided. This implies that mAP values of our method are under-estimated.

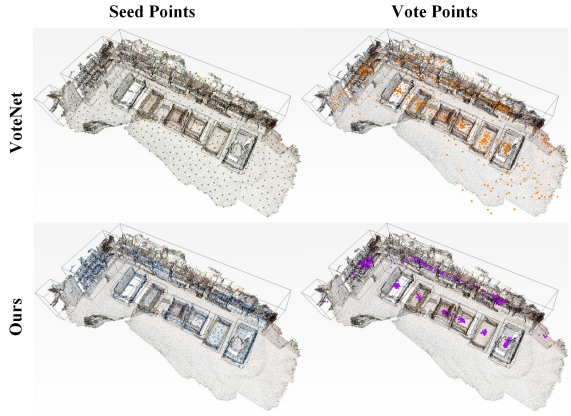

Figure 8. Semantic Guided Sampling and Voting. Best viewed on screen. See supplementary for more results.

results than Naive + SGM option. Finally, when we use ShapeFormer + SGM, we also avoid the loss of fine-grained information in shape encoding, achieving a significant 3.5% absolute performance improvement over VoteNet.

Figure 8 visualizes the positive impact of the SGM module on seed point sampling and voting during testing. The first row shows the seed points and vote points without the SGM module [29]. It can be seen that the seed points and vote points contain a large number of outliers. More background points affect the quality of shape encoding adversely. Scattered vote points containing a large number of outliers cannot provide high-quality candidate points for the detection head. The second row shows that with the help of SGM, we find more foreground points, and the vote points are also more compact and closer to the center, which is beneficial to our shape encoding and vote aggregation.

The candidate points of vanilla VoteNet [29] are aggregated from voting points. The vote points only contain the positioning information of the object, which is sub-optimal. Our method makes full use of the shape key points distributed on the surface of the object, which can encode the shape of the object, so it can more accurately predict the size and direction of the boxs. Table 4 summarizes a clearly favorable comparison of the prediction size errors of AShapeFormer and VoteNet [29] on the ScanNet dataset.

VoteNet [29]	✓	✓	✓	✓	✓
Naive		✓	✓		
SF				✓	✓
SGM			✓		✓
mAP@0.25	57.7	58.9	59.5	59.9	61.2

Table 3. Contribution of sub-modules of AShapeFormer (SUN RGB-D dataset). Naive is the naive approach from § 3.2. SF is Shape Former (§ 3.3), SGM is Semantic Guided Module (§ 3.3).

Method	mean size_cls_loss	mean vote_loss
VoteNet [29]	0.52	0.04
AShapeFormer	0.38	0.03

Table 4. Detection size error comparison on ScanNet V2.

5. Conclusion

We proposes a plug-n-play module to improve the performance of indoor 3D object detection by actively encoding shape information of the object. We first sample the shape key points of the object and re-weight their features by guiding them with semantic information. Then, to avoid the loss of fine-grained information, we utilize multi-head attention to encode object shape features. Finally, object-level shape features are fused with the candidate features and fed to the detection head. Results show that our model achieves state-of-the-art performance when assembled with existing methods. In the future, we will explore to incorporate RGB images and point cloud completion methods to encode more complete shape information in our technique.

Acknowledgement. This work was supported by the NSFC (U2013203, 61973106, U1913202); the Natural Science Fund of Hunan Province (2021JJ10024, 2022JJ40100); the Project of Talent Innovation and Sharing Alliance of Quanzhou City under Grant 2021C062L; the Key Research and Development Project of Science and the Technology Plan of Hunan Province under Grant 2022GK2014.

References

[1] Eduardo Arnold, Omar Y Al-Jarrah, Mehrdad Dianati, Saber Fallah, David Oxtoby, and Alex Mouzakitis. A survey on 3d object detection methods for autonomous driving applications. *IEEE Transactions on Intelligent Transportation Systems*, 20(10):3782–3795, 2019. 1

[2] Ronald T Azuma. A survey of augmented reality. *Presence: teleoperators & virtual environments*, 6(4):355–385, 1997. 1

[3] Mark Billinghurst, Adrian Clark, Gun Lee, et al. A survey of augmented reality. *Foundations and Trends® in Human–Computer Interaction*, 8(2-3):73–272, 2015. 1

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2, 4

[5] Chen Chen, Zhe Chen, Jing Zhang, and Dacheng Tao. Sasa: Semantics-augmented set abstraction for point-based 3d object detection. In *AAAI Conference on Artificial Intelligence*, volume 1, 2022. 5

[6] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3d object detection on point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 392–401, 2020. 3, 7

[7] Bowen Cheng, Lu Sheng, Shaoshuai Shi, Ming Yang, and Dong Xu. Back-tracing representative points for voting-based 3d object detection in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8963–8972, 2021. 1, 2, 3, 4, 6, 7

[8] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020. 6

[9] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 2, 6, 7

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 2

[11] Yao Duan, Chenyang Zhu, Yuqing Lan, Renjiao Yi, Xinwang Liu, and Kai Xu. Disarm: Displacement aware relation module for 3d detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16980–16989, 2022. 5, 7

[12] Francis Engelmann, Martin Bokeloh, Alireza Fathi, Bastian Leibe, and Matthias Nießner. 3d-mpa: Multi-proposal aggregation for 3d semantic instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9031–9040, 2020. 1

[13] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12):4338–4364, 2020. 1

[14] JunYoung Gwak, Christopher Choy, and Silvio Savarese. Generative sparse detection networks for 3d single-shot object detection. In *European conference on computer vision*, pages 297–313. Springer, 2020. 2

[15] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[16] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 1, 2

[17] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3588–3597, 2018. 2

[18] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017. 1

[19] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 2, 4, 5

[20] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12697–12705, 2019. 1

[21] Bo Li. 3d fully convolutional network for vehicle detection in point cloud. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1513–1518. IEEE, 2017. 2

[22] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[23] Ze Liu, Zheng Zhang, Yue Cao, Han Hu, and Xin Tong. Group-free 3d object detection via transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2949–2958, 2021. 2, 3, 6, 7

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 6

[25] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015. 2

[26] Ishan Misra, Rohit Girdhar, and Armand Joulin. An end-to-end transformer model for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2906–2917, 2021. 2, 7

[27] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International conference on machine learning*, pages 4055–4064. PMLR, 2018. 2, 5

[28] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4404–4413, 2020. 1, 2, 3, 4, 6, 7

[29] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 2, 3, 4, 6, 7, 8

[30] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 2

[31] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1, 2, 3, 4

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 2, 5

[33] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[34] Danila Rukhovich, Anna Vorontsova, and Anton Konushin. Fcaf3d: Fully convolutional anchor-free 3d object detection. In *European Conference on Computer Vision*. Springer, 2022. 2, 7

[35] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 4

[36] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. 1

[37] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1

[38] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015. 2, 6, 8

[39] Shuran Song and Jianxiong Xiao. Deep sliding shapes for amodal 3d object detection in rgb-d images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 808–816, 2016. 1, 2

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2, 4, 5

[41] Chen Wang, Danfei Xu, Yuke Zhu, Roberto Martín-Martín, Cewu Lu, Li Fei-Fei, and Silvio Savarese. Densefusion: 6d object pose estimation by iterative dense fusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3343–3352, 2019. 1

[42] Haiyang Wang, Shaoshuai Shi, Ze Yang, Rongyao Fang, Qi Qian, Hongsheng Li, Bernt Schiele, and Liwei Wang. Rbgnet: Ray-based grouping for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1110–1119, 2022. 2, 3, 4, 6, 7

[43] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[44] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12186–12195, 2022. 3, 6, 7

[45] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Dening Lu, Mingqiang Wei, and Jun Wang. Venet: Voting enhancement network for 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3712–3721, 2021. 1, 2

[46] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10447–10456, 2020. 1, 2, 7

[47] Xiuwei Xu, Yifan Wang, Yu Zheng, Yongming Rao, Jie Zhou, and Jiwen Lu. Back to reality: Weakly-supervised 3d object detection with shape-guided label enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8438–8447, 2022. 1

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

[49] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3d single stage object detector. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11040–11048, 2020. 5

[50] Zaiwei Zhang, Bo Sun, Haitao Yang, and Qixing Huang. H3dnet: 3d object detection using hybrid geometric primitives. In *European Conference on Computer Vision*, pages 311–329. Springer, 2020. 1, 2

[51] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018. 2

[52] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017. 1