ISPRS Journal of Photogrammetry and Remote Sensing 165 (2020) 43–53

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

GRNet: Geometric relation network for 3D object detection from point clouds

Ying Li[a,b], Lingfei Ma[a], Weikai Tan[a], Chen Sun[b], Dongpu Cao[b,*], Jonathan Li[a,c,*]

[a] Department of Geography & Environmental Management, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada
[b] Waterloo Cognitive Autonomous Driving Lab, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada
[c] Department of Systems Design Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada

ARTICLE INFO

Keywords:
Deep learning
3D object detection
Point cloud
Geometric relation
Indoor mapping
RGB-D

ABSTRACT

Rapid detection of 3D objects in indoor environments is essential for indoor mapping and modeling, robotic perception and localization, and building reconstruction. 3D point clouds acquired by a low-cost RGB-D camera have become one of the most commonly used data sources for 3D indoor mapping. However, due to the sparse surface, empty object center, and various scales of point cloud objects, 3D bounding boxes are challenging to be estimated and located accurately. To address this, geometric shape, topological structure, and object relation are commonly employed to extract box reasoning information. In this paper, we describe the geometric feature among object points as an intra-object feature and the relation feature between different objects as an inter-object feature. Based on these two features, we propose an end-to-end point cloud geometric relation network focusing on 3D object detection, which is termed as geometric relation network (GRNet). GRNet first extracts intra-object and inter-object features for each representative point using our proposed backbone network. Then, a centralization module with a scalable loss function is proposed to centralize each representative object point to its center. Next, proposal points are sampled from these shifted points, following a proposal feature pooling operation. Finally, an object-relation learning module is applied to predict bounding box parameters. Such parameters are the additive sum of prediction results from the relation-based inter-object feature and the aggregated intra-object feature. Our model achieves state-of-the-art 3D detection results with 59.1% mAP@0.25 and 39.1% mAP@0.5 on ScanNetV2 dataset, 58.4% mAP@0.25 and 34.9% mAP@0.5 on SUN RGB-D dataset.

1. Introduction

With the rapid development of urbanization and the prevalence of commercial and residential buildings, 3D object detection plays a vital role in many applications such as indoor mapping and modeling (Chen et al., 2014), scene understanding (Lin et al., 2013), location-based services (Li et al., 2019a; Chen et al., 2019), and building maintenance (Wang et al., 2018). It seeks to localize and recognize objects in 3D scenes (Li et al., 2019a), including their bounding box size, orientation, and center position (Haala and Kada, 2010). Compared with images, 3D point clouds have advantages in providing precise geometry and robustness to illumination variations. However, the primary problems of 3D object detection are: (1) points distribute sparsely and irregularly (Qi et al., 2017a), (2) geometric patterns vary enormously (Ren and Sudderth, 2018), and (3) points locate on the surface of objects, far from their center (Qi et al., 2019). These challenges lead to the complexity of localization and detection of 3D objects in indoor environments.

In existing 3D detection methods, there are three main 3D data representations: voxel grids, multiview images, and point clouds (Griffiths and Boehm, 2019). Voxel grids (Zhou and Tuzel, 2018) and multi-view images (Kanezaki et al., 2018) are Euclidean-structured data, which are first converted from point clouds and then input to 2D convolutional networks for detection. For example, Zhou and Tuzel (2018) transformed 3D point clouds to voxel grids, and Chen et al. (2017) projected LiDAR data to bird's eye view images, then 2D CNNs were employed for object detection. However, these regular data formats may obscure natural 3D geometric patterns and disturb the invariances of data. To leverage the spatial relation between 2D and 3D data, Chen et al. (2016) first sampled candidate 3D bounding boxes in

* Corresponding authors at: Department of Geography & Environmental Management, University of Waterloo, 200 University Avenue West, Waterloo, ON N2L 3G1, Canada (J. Li).
E-mail addresses: y2424li@uwaterloo.ca (Y. Li), l53ma@uwaterloo.ca (L. Ma), weikai.tan@uwaterloo.ca (W. Tan), chen.sun@uwaterloo.ca (C. Sun), dongpu.cao@uwaterloo.ca (D. Cao), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.isprsjprs.2020.05.008
Received 18 February 2020; Received in revised form 9 May 2020; Accepted 11 May 2020

point clouds and then projected these boxes to 2D images for further detection. Qi et al. (2018) proposed 3D box prediction that relies on 2D detection results. However, these 2D-3D detection pipelines highly rely on the 2D detection results from the first step. Recently, a set of papers have proposed to process point clouds directly. PointRCNN (Shi et al., 2019), STD (Yang et al., 2019b), VoteNet (Qi et al., 2019), and 3D-BoNet (Yang et al., 2019a) detected 3D objects using end-to-end deep point set networks. Local and global features are extracted for object feature learning and bounding box reasoning. However, geometric patterns and object relationships are not exploited in these networks.

When constructing our detection framework, we face two selections: one-stage detection and two-stage detection. One-stage detection (Yang et al., 2019a) generates bounding boxes directly from the extracted point set features without any post-processing steps for refinement. Two-stage detection methods (e.g., Qi et al., 2019; Yang et al., 2019b; Shi et al., 2019; Hou et al., 2019) mainly consist of two steps: proposal generation and bounding box refinement. One-stage detection is efficient and straightforward but highly relies on the performance of the proposed algorithm. If some difficult objects or geometric-salient objects that could be clearly distinguished are missed, they have no chance to be retrieved (Qi et al., 2019). Two-stage detection considers sufficient possible candidates in the first step and refines the coarse results in the second step. This can sometimes avoid misdetection and thus commonly has higher detection performance and computation cost than the former (Yang et al., 2019a). In order to achieve a discriminate performance, we select the two-stage pipeline to construct our model and try to reduce the computation burden.

Different from previous work that inputs RGB-D data as images to 2D CNNs for detection (Gupta et al., 2014), we detect 3D objects from point clouds lifted from depth maps. Geometric attributes and topological structures of 3D objects can be exploited using such data representation (Li et al., 2019b; Xu et al., 2018b). For example, plane, curve, line, and corner are more easily parameterized and described by 3D learners. In this paper, we introduce an efficient and novel bottom-up two-stage 3D object detection framework from point clouds in indoor scenes, termed as geometric relation network (GRNet). We mainly focus on three challenges to improve the 3D detection performance:

(1) Bottom-up feature learning of representative points. Only certain points are selected as candidate points for proposal selection. Their intra-object and inter-object features are exploited.
(2) Centralization of object surface points. 3D object centers are likely to be empty without any point (Qi et al., 2019). We centralize surface points for more accurate bounding box prediction.
(3) Object relation learning. Relation features among 3D proposals can attribute to the bounding box parameter refinement.

To encode the local geodesic information (e.g., coarse local shape) for representative points, we mimic TGNet (Li et al., 2019b) to explore geodesic correlations and attributes among local neighbors. We observe that, in indoor scenes, the topological structure of points in the local region has limited geometric variations. For example, most object surfaces (e.g., beds, desks, and tables) are flat or in regular shape. Thus, we replace the Taylor-Gaussian geometric function with the exponentially trilinear interpolation function to approximate local surface features. We term this new convolution operation as GeoConv. GeoConv is similar to TGConv, but simpler and has fewer parameters.

Our bottom-up backbone framework is constructed based on an encoder-decoder structure, with four-layer down-sampling and two-layer up-sampling layers. To extract both intra-object and inter-object features, GeoConv is applied to the first two down-sampling layers to exploit the intra-object geometric features. We leverage PointNet (Qi et al., 2017a) in the last two down-sampling layers to extract inter-object features. These features are then propagated and concatenated to two up-sampling layers. The output of the backbone network is the selected representative points and their propagated bottom-up features.

Due to the empty object center, VoteNet (Qi et al., 2019) proposes a Hough voting module to regress the surface points to their center. Such operation has been proved effective in 3D object detection. However, the scaling problem is not considered, which results in the sub-optimal regression for small or vertical objects. We follow VoteNet (Qi et al., 2019) to propose a centralization module with a scalable loss function. By adding a scaling control parameter in defining the centralization loss function, object points within a different pattern are centralized in a compact way, which further increases the bounding box prediction results.

Proposals are sampled from these shifted representative points. Their features are learned and aggregated from their nearest neighboring points which are mostly from the same object. Many methods (e.g., Qi et al., 2019; Shi et al. 2019; Yang et al., 2019a) predict bounding boxes using such aggregated intra-object features. However, the relation feature between proposals is not exploited. Thus, we propose a simple relation learning module to learn both intra- object and inter-object features to increase the prediction results. Only features from a certain number of nearest neighbors for each proposal are considered for relation feature learning. These neighbors are searched based on the predicted bounding box center and then the intra-object features are aggregated. Bounding box parameters are generated as the additive sum of prediction results from relation-based inter-object features and aggregated intra-object features.

Our method achieves leading positions on indoor RGB-D datasets. We have achieved 58.4% better 3D mAP@0.25 than VoteNet (Qi et al., 2019) on SUN RGB-D dataset. Besides, our method outperforms VoteNet (Qi et al., 2019) and 3D-SIS (Hou et al., 2019) on ScanNetV2 dataset. Compared with the previous best performance from VoteNet (Qi et al., 2019), our method achieves at least 0.5% mAP@0.25 and 5.5% mAP@0.5 increases with high efficiency. The key contributions of our work are as follows:

(1) A novel geometric convolution is proposed and applied in a bottom-up backbone network. Intra-object geometric features and inter-object relation features for each representative point are extracted in a hierarchical way.
(2) A centralization module is presented to centralize object surface points to its center. This contributes to an improved bounding box prediction.
(3) An object relation learning module is introduced to exploit the relation feature between proposals for better bounding box reasoning.

The rest of this paper is structured as follows. Section 2 reviews the related work. Section 3 details the proposed method. Section 4 describes the environmental setup, presents and discusses the results. Section 6 concludes this paper.

2. Related work

View-based Methods. In order to exploit existing 2D CNNs, some approaches first project point clouds into 2D views and then apply 2D CNNs to detect and localize objects from images. In early work by Xiang et al. (2015), Chen et al. (2016) and Mousavian et al. (2017), point clouds were projected initially to the camera image plane, then RGB images and shape attributes or occlusion patterns were exploited to predict 3D bounding boxes. Li et al. (2016) and Deng and Jan Latecki (2017) treated depth data as 2D maps and applied 2D CNN learners to detect objects in 2D images. Luo et al. (2019) proposed a detection framework via fusing multi-view representations of point clouds to extract high-level features. Wen et al. (2019) projected point clouds into a horizontal plane and used a modified U-net to extract road markings. MV3D (Chen et al., 2017) projected LiDAR point clouds to bird's eye view images first and then constructed a region proposal network (RPN (Ren et al., 2015)) for 3D bounding box prediction. However, these methods have sub-optimal performance in small object

detection (e.g., pedestrians and cyclists) and multiple clutter object detection in the vertical direction. View-based methods are effective and lightweight. Due to the sparsity of point clouds, the projection of point clouds to 2D image planes produces sparse 2D point maps and losses 3D geometric information.

2D-3D Object Detection from RGB-D data. RGB-D data contain RGB features and depth information for each object (Chen et al., 2014). There are multiple methods to exploit these two features. Song and Xiao (2014) proposed sliding shapes based on a 3D volumetric scene extracted from the RGB-D input image to predict 3D bounding boxes. 3D RPN was proposed to learn the objectiveness based on geometric shapes. 3D geometric features and 2D color attributes were learned via a joint object recognition network (ORN). However, the spatial relationship between the 2D imagery and the 3D geometry for joint feature learning is not considered. Lahoud and Ghanem (2017) presented a 2D-driven 3D object detection network that uses 2D detection results to reduce 3D searching space. Then, a histogram of point coordinate was input to simple fully connected networks to regress the bounding box location and pose direction. Hou et al. (2019) integrated 2D images with voxelized point cloud grids based on their geo-spatial relationship to segment semantic instances in commodity RGB-D scans. Qi et al. (2018) and Gong et al. (2020) proposed similar detection frameworks as Lahoud and Ghanem (2017). But Frustum PointNet (Qi et al., 2018) utilized a more flexible and effective PointNet (Qi et al., 2017a) network to perform 3D object instance segmentation and amodal bounding box regression. In comparison, we lift RGB-D data to point clouds and propose a new 3D deep network that can exploit 3D geometry more effectively using point clouds only.

3D-based methods. Compared with view-based detection methods and 3D object detection using 2D-3D features, 3D-based approaches focus more on utilizing geometric features from point clouds. In work by Song and Xiao (2014) and Wang and Posner (2015), SVMs were adopted to classify 3D objects using hand-designed geodesic features extracted from point clouds. Then the object was localized via a sliding window search. Engelcke et al. (2017) extended the work by Wang and Posner (2015) by using 3D CNN instead of SVM on 3D voxelized grids. Ren et al. (2016) designed new geometric features for 3D object detection. Song and Xiao (2016) converted the entire scene represented by point clouds into volumetric grids and applied 3D volumetric CNNs on object proposal for classification. The computation costs for these methods are usually high because 3D convolutions and 3D space searching in large areas cost expensively. More recently, deep networks on point clouds were adopted by GSPN (Yi et al., 2019) and PointRCNN (Shi et al., 2019) to exploit the sparsity of the data. Considering the scanned points lying on the surface of the objects and the empty object center, Qi et al. (Qi et al., 2019) proposed a deep Hough voting network to shift the surface point to the object center. This method achieved discriminate performance in bounding box center prediction and box size estimation.

Object Relation Methods. Recently, the attention mechanism has received increasing noticing in 3D deep learning (Velic̆ković et al., 2017; Wang et al., 2019). The key advantage of attention mechanisms is that it can consume different sized inputs, which is suitable for irregular and unstructured point cloud data. Besides, its output only focuses on the most relevant parts of the input by considering the relationship between neighbors (Velic̆ković et al., 2017). Wang et al. (2019) proposed a novel graph attention convolution (GAC) to learn the most relevant part of different neighboring points based on their dynamically learned features. Such an operation can adjust kernels into specific shapes to adapt to the structure of an object. Zhang and Xiao (2019) introduced a Point Contextual Attention Network (PCAN) to predict the importance of each local point feature considering the 3D context information. The attention mechanism was utilized when aggregating local features to task-relevant features. Xie et al. (2018) also introduced a contextual modeling mechanism inspired by the self-attention mechanism in constructing ShapeContextNet for point cloud recognition. The above approaches show that the attention mechanism has the strength in extracting the most relevant features by learning their relationships. Therefore, motivated by these methods, in this paper, we apply the attention mechanism to deal with the task of 3D object detection. There exist geometric relationships between objects, such as chairs usually locate near desks. We introduce an attention-based object relation feature learning module to aggregate the most relevant neighbors' feature as an inter-object feature, which is combined with the intra-object feature to predict accurate bounding box parameters.

3. Method

In this section, we introduce the detailed framework of our proposed GRNet for 3D detection from point clouds. We first extract geometric features *with* R^C for M representative points using a bottom-up backbone network. Such a network can help each representative point to aggregate its intra-object and inter-object features. Then, a centralization module is proposed to centralize object surface points to its center. After that, K proposal points are selected, and their features are learned and aggregated from their neighboring points. Such features with $R^{(C+3)}$ are then passed to an object relation learning module for bounding box refinement. The overall structure is illustrated in Fig. 1. The input to our network is N points with their XYZ coordinates, and the output is bounding box parameters *with* $R^{C_{out}}$. These parameters are finally post-processed by non-maximum-suppression (NMS) for accurate object bounding box prediction.

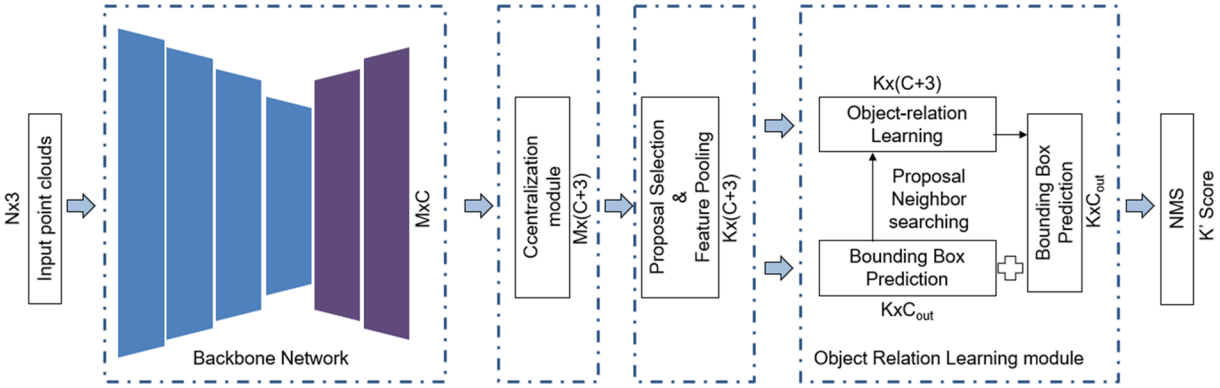

Fig. 1. The framework of GRNet: We first extract geometric features for M representative points using a bottom-up backbone network from N input points. Then a centralization module is proposed to centralize object surface points to its center. After that, K proposal points are selected, and their features are learned and aggregated from their neighboring points. Such features are then passed to an object relation learning module for bounding box refinement.

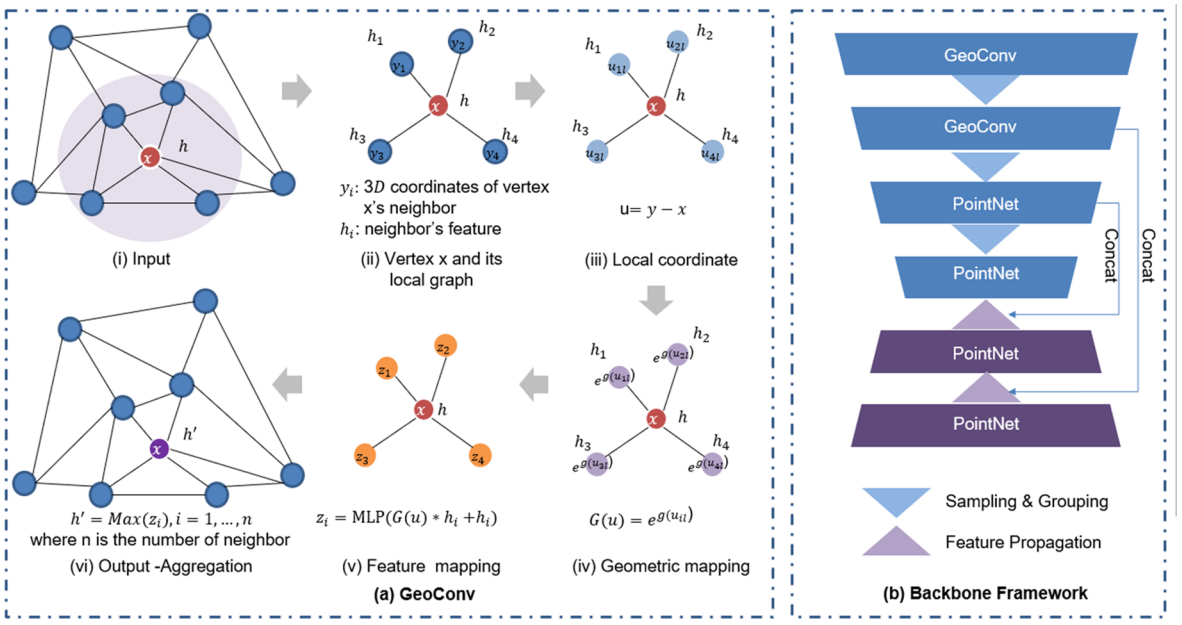

Fig. 2. Details of our proposed backbone network. (a) GeoConv on a graph representation of point clouds: given n nearest neighbors' coordinates y_i and features h_i for each vertex x with $i = 1, \cdots, n$, a local operation is conducted. The product of local neighbor point feature with a geometric feature extracted from relative-coordinates expressed by the parameterized exponential trilinear interpolation function and the input feature are added. Then a max-pooling operation is applied to aggregate such newly generated feature to a new feature $h^{'}$ for vertex x. (b) The architecture of our backbone network, which has four downsampling layers and two upsampling layers. GeoConv is applied to the first two downsampling layers to extract intra-object features, while PointNet (Qi et al., 2017a) is employed in the last two downsampling layers to learn inter-object feature. Then these features are concatenated and interpolated in the following two upsampling layer using PointNet.

3.1. Backbone network

The backbone network is proposed based on the following considerations: (1) intra-object attributes extraction, such as geometric shape, surface variation, and correlation between closed points; (2) inter-object attributes exploitation, e.g., relation features between objects; (3) feature learning and aggregation in a hierarchical way, which can extract point features in different scales; (4) representative points selection, these points are selected to represent the input scene to reduce the computation cost. To meet the above requirements, we construct a bottom-up hierarchical deep framework using a newly defined geometric CNN, GeoConv. The following parts introduce the details of the proposed backbone network.

3.1.1. GeoConv

Although TGNet (Li et al., 2019b) proposed the TGConv to explore geodesic correlations and attributes among local neighbors for each point. However, it introduces a high amount of parameters. Besides, we observe that, in indoor scenes, most points and their local neighbors lie on planes or regular shape surfaces, which can be described by a simplified parameterized geometric function. To reduce the number of parameters and exploit the geodesic intra-object feature of indoor objects, we propose a new geometric CNN, termed GeoConv. GeoConv is similar to TGConv, but simpler and focuses on regular and simplified geometric characteristics.

Given a 3D point cloud $P = \{p_1, \cdots, p_n\} \subseteq R^3$ according to their Euclidean nearest neighbors, a graph $G = (V, E)$ is constructed. $V = \{1, 2, \ldots, n\}$ and $E \subseteq V \times V$ denote vertices and edges respectively. The neighbor set for each vertex x is denoted as $y \in \mathrm{N}(x)$. Let $h = \{h_1, h_2, \ldots, h_N\}$ be a set of input vertex features, each feature $h_N \in R^F$ corresponds to a graph vertex $i \subseteq V$. F represents each vertex's feature dimension. The output $h_y^{'}$ of GeoConv for each vertex is derived as follows:

$$h_y^{'} = max(g_\theta(G(\boldsymbol{u}(x, y)) \bullet h_y + h_y)), y \in \mathrm{N}(x) \quad (1)$$

where $G(\bullet)$ is a geometric mapping function: $R^3 \rightarrow R$, which maps the local Euclidean coordinates $\boldsymbol{u}(x, y) = \boldsymbol{u}(y) - \boldsymbol{u}(x)$ between each vertex and its neighbors' Euclidean coordinates to a geometric parameter. Then the product of $G(\boldsymbol{u}(x, y))$ and feature h_y is added with h_y. $g_\theta(\hat{A}\cdot)$ is the learnable feature mapping function: $R^F \rightarrow R^K$, $max(\hat{A}\cdot)$ denotes the max aggregation function.

As mentioned in TGConv (Li et al., 2019b), a family of parametrized Taylor-Gaussian filters were proposed to interpolate arbitrary values at the vertexes of a graph and capture geometric spatial information in a local region. These filters are defined as products of local neighbor point features with geometric features extracted from local coordinates expressed by a family of Gaussian weighted Taylor kernel functions. TGConv is suitable for both indoor and outdoor objects with variable geometric shapes. However, in indoor scenes, common objects have regular geometric shapes. As mentioned in SpiderCNN (Xu et al., 2018a,b), a family of parameterized trilinear interpolation based kernels have been demonstrated to be effective in extracting geometric features. To reduce the number of parameters but also maintain the kernel's expression ability, an exponential-based trilinear interpolation function is used in this paper as the geometric mapping function $G(u(x, y))$ with learnable parameters as:

$$\begin{aligned} G(u(x, y)) &= G(\Delta x, \Delta y, \Delta z) \\ &= e^{(\sigma_0^T + \sigma_1^T \Delta x + \sigma_2^T \Delta y + \sigma_3^T \Delta z + \sigma_4^T \Delta x \Delta y + \sigma_5^T \Delta x \Delta z + \sigma_6^T \Delta y \Delta z + \sigma_7^T \Delta x \Delta y \Delta z)} \end{aligned} \quad (2)$$

where $\sigma_i^T (i = 0, \cdots, 7)$ is a 1×1 learnable parameter. By varying these parameters, $G(u(x, y))$ can approximate different geodesic values for each vertex x using its neighbor set $y \in \mathrm{N}(x)$.

Because a multi-layer perception (MLP) can approximate an arbitrary continuous function and retains weight sharing as standard convolution (Xu et al., 2018b). We use an shared MLP as our feature mapping function $g_\theta(\hat{A}\cdot)$ to map the addition of the original input feature h_y and the products of h_y with a geometric feature $G(u(x, y))$ to a different feature dimension: $R^F \rightarrow R^K$. Max aggregation, which can exploit the most effective features and adaptively explore related neighbor features (Qi et al., 2017a), is then applied to aggregate the learned new feature $h_y^{'}$.

3.1.2. Backbone framework

A good backbone framework should meet the above four requirements. In VoteNet (Qi et al., 2019), PointNet++ (Qi et al., 2017b) is chosen as the backbone network, which is a hierarchical deep framework with representative point selection. However, the intra-object and inter-object features are not fully exploited. We construct our backbone framework based on PointNet++, but also explore these two features.

Due to the high density of point clouds in the first two downsampling layers, the extracted local neighbors for each point still construct part of the object surface. As shown in Fig. 2, we apply GeoConv in these two upsampling layers to extract intra-object features. When points are sampled sparsely, especially in the last two encoder layers, geometric attributes (e.g., shape) among extracted neighbors are weakened but the inter-object features (e.g., position or layout) are enhanced. Because using GeoConv in all four encoder layers cannot extract the inter-object features, it sharpens the detection performance. Thus, in the last two downsampling layers, we adopt PointNet (Qi et al., 2017a) to extract inter-object features. Then these features are concatenated and interpolated in the following two upsampling layers using PointNet. The output of this backbone is a set of representative points $\{r_i\}_{i=1}^{M}$ where $r_i = [x_i; f_i]$ with $x_i \in R^3$ and $f_i \in R^C$.

3.2. Centralization module

Due to depth sensors mainly capturing surface points of objects, there are limited points or no points around object centers. Thus, existing point-based networks have a problem in extracting scene context around the object center. To solve this, VoteNet (Qi et al., 2019) proposed a Hough voting module to generate new points (votes) that lie close to the object center. Votes are generated from features of representative points. Then these votes can be grouped and aggregated with a learnable module to generate proposals with enough context information. A vote loss function is introduced to regress the displacements of votes based on the Euclidean distance. This network has been demonstrated to be effective in 3D object detection. However, the scaling problem is not considered in defining their vote loss function. Large objects (i.e., bed) can regress better than small objects (i.e., chair) (Qi et al., 2019). To improve this, we follow VoteNet to construct our centralization module but introduce a scaling control parameter in defining the loss function.

Given a set of representative points $\{r_i\}_{i=1}^{M}$, the centralization module generates offset from each representative point position to its center independently. This module is composed of a shared MLP module with three fully connected layers, ReLU and batch normalization. The input is the feature $f_i \in R^C$ of representative points, and the output is the 3D position offset $\Delta x_i \in R^3$ in the Euclidean domain and a feature offset $\Delta f_i \in R^C$. Thus, this module generates $c_i = [y_i; g_i]$ from the representative point r_i and has $y_i = x_i + \Delta x_i$ and $g_i = f_i + \Delta f_i$.

The predicted 3D offset Δx_i is supervised by the following loss function:

$$L_{offset-reg} = \frac{1}{N_{pos}} \sum_i \frac{\|\Delta x_i - \Delta x_i^*\|}{\gamma} 1[r_i \ \ on \ \ object] \tag{3}$$

where $1[r_i \ \ on \ \ object]$ represents whether a representative point r_i is on an object surface, N_{pos} is the total number of representative points on object surface. Δx_i^* is the ground truth displacement from the representative point position x_i to the bounding box center of the object it belongs to. γ is a scale control parameter, which is set to 0.1 in our experiments. Because the offset of different object points varies. Thus, we add a scaling control parameter to enlarge the distance-based regression loss for small objects. Experimental results demonstrate the effective of such scale control parameter.

3.3. Proposal selection and feature pooling

The centralization module moves the object surface points to the object center compactly, while background points still distribute sparsely. Thus, proposal selection should consider such density variation. To ensure the proposal can represent enough possible objects, the sampling and clustering methods are selected according to spatial proximity. A subset of K points are sampled using farthest point sampling (FPS) (Qi et al., 2017b) based on the representative point position $\{x_i\}_{i=1}^{M}$ in 3D Euclidean space. The index of these points is then used to find proposals in shifted representative points $\{y_i\}_{i=1}^{M}$, to get $\{p_k\}_{k=1}^{K}$.

After that, we cluster N group points for each proposal by searching neighboring points $p_k^{(n)}$ in $\{y_i\}_{i=1}^{M}$, if $\|p_k^{(n)} - p_k\| \le r$ for $n = 1, \cdots N$. The corresponding feature for each grouped point is denoted as $g_k^{(n)}$. Ball query searching (Qi et al., 2017b) is adopted as the nearest neighbor finding method, which only considers neighboring points in a fixed radius r. N is set to 16 and the r is set to 0.2 according to experimental results. Although smaller radius can include more clean neighbors (from the same object), it loses context information from background points. Increasing $r$can contaminate neighbors because more nearby object and clutter points are included.

For each proposal, we use a shared MLP for neighboring points' feature mapping. The max operation is applied for feature aggregation:

$$F_k = \underset{n = 1, \cdots N}{max} \{MLP([r_k^{(n)}; g_k^{(n)}])\} \tag{4}$$

where $r_k^{(n)} = p_k^{(n)} - p_k$ is the relative coordinate between neighboring points to its proposal, and $F_k \in R^{(3+C)}$. This aggregated output feature represents the intra-object attribute, because neighboring points mainly come from the same object.

3.4. Object relation learning module

As for discriminate 3D object detection, intra-object feature and inter-object feature are of the same importance. The above aggregated proposal feature represents the intra-object feature generated from points that lies on the same object surface. However, in the real scene, there exists relationships between objects. To leverage the inter-object feature between co-occurrence and locations of objects for better reasoning, we propose an object relation learning module.

We only consider S nearest neighboring proposals for each proposal to leverage their relation features. These neighboring proposals are searched based on the predicted bounding box center position. In this paper, a 3D bounding box is represented as $(x, y, z, h, w, l, \theta)$, where (x, y, z) is the object center coordinates, (h, w, l) is the object size (height, width, length), and θ is the object orientation. Three fully connected layers are applied to predict bounding box parameters $B_{k,1}(x_{k,1}, y_{k,1}, z_{k,1}, h_{k,1}, w_{k,1}, l_{k,1}, \theta_{k,1})$ using the intra-object feature F_k.

Each proposal neighbors are searched using the predicted bounding box center position $(x_{k,1}, y_{k,1}, z_{k,1})$. We formulate the relation between a proposal to its neighboring proposals as a region-to-region undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, ...,S\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denote vertices and edges respectively. The ith neighboring proposal feature is denoted as$F_{i,k}$. We then seek to learn the relation parameter $\alpha_{i,k} \in R^{1\times(3+C)} (i = 1, ..S)$, and the object relation feature F_{rk} as follows:

$$\alpha_{i,k} = \frac{\exp(\widetilde{\alpha_{i,k}} * F_{i,k})}{\sum_{i=1}^{S} \exp(\widetilde{\alpha_{i,k}} * F_{i,k})} \tag{5}$$

$$F_{rk} = \sum_{i=1}^{S} \alpha_{i,k} * F_{i,k} + F_k \tag{6}$$

where $\widetilde{\alpha_{i,k}} (i = 1, ..S)$ is a $1 \times (3 + C)$ learnable parameter. This newly generated relation feature F_{rk} is then sent to three fully connected layers for bounding box parameter prediction, which is denoted as $B_{k,2}(x_{k,2}, y_{k,2}, z_{k,2}, h_{k,2}, w_{k,2}, l_{k,2}, \theta_{k,2})$. The final output of this network is the additive sum of $B_{k,1}$ and $B_{k,2}$. Fig. 3 illustrates the framework of this module.

Following VoteNet (Qi et al., 2019), we use a hybrid of classification and regression formulation. For angle prediction, we pre-define N_a as equally split angle bins and classify the proposal angle into different

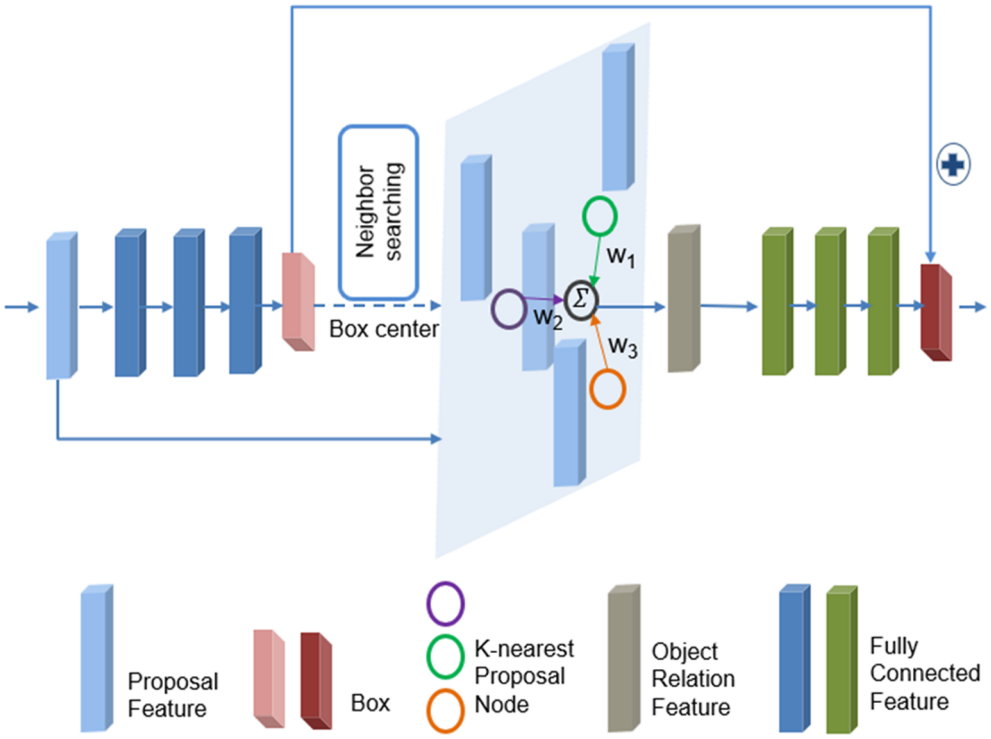

Fig. 3. The framework of the object relation learning module.

bins. Residual is regressed with respect to the bin value. N_a is set to 12 in our experiments. Finally, the NMS based on the objectness score and semantic classification score is applied to eliminate redundant proposals. Specifically, we keep up to 256 proposals during training and testing.

3.5. Loss function

To optimize the proposed end-to-end framework, a multi-task loss is applied. It includes a centralization loss, a 3D bounding box estimation loss, a semantic classification loss, and an objectness loss:

$$L_{GRNet} = L_{offset-reg} + \lambda_1 L_{box} + \lambda_2 L_{sem-cls} + \lambda_3 L_{obj-cls} \quad (7)$$

where $\lambda_1 = 1$, $\lambda_2 = 0.1$ and $\lambda_3 = 0.5$. These parameters are used to weight the losses to maintain that they have similar scales.

$L_{offset-reg}$ is as defined in Section 3.2. As for the last three losses, we follow VoteNet (Qi et al., 2019) to construct them. Both the objectness loss and the semantic classification loss are cross-entropy loss, but for two classes and C semantic classes, respectively. Only positive proposals are considered in calculating the box and semantic losses, which are normalized by the total number of positive proposals. Those proposals, whose distances to their nearest ground truth center are less than 0.2 m, are defined as positive proposals. For those proposals with distance larger than 0.5 m are denoted as negative proposals. Those proposals whose distances are between these two thresholds are neglected. These distance thresholds are determined by experimental results.

The box loss is composed of the center regression, heading estimation and size estimation sub-losses using L1-smooth loss (Qi et al., 2018):

$$L_{box} = L_{center-reg} + 0.1L_{ang-cls} + L_{angle-reg} + 0.1L_{size-cls} + L_{size-reg} \quad (8)$$

where center regression loss $L_{center-reg}$ is defined by Chamfer loss (Fan et al., 2017).

4. Results and discussion

4.1. Experimental setup and Implementation

Dataset. The performance of our method is evaluated on two indoor datasets: SUN RGB-D (Song et al., 2015) and ScanNetV2 (Dai et al., 2017). SUN RGB-D is collected using multiple different RGB-D cameras with varying resolutions from different indoor scenes. It contains 5,285 training images and 5,050 testing images, respectively. There are 37 object categories labeled with amodal oriented 3D bounding boxes. We report model performance on the testing set. Point cloud data are acquired following the method provided by VoteNet (Qi et al., 2019). Detection results on the 10 most common categories are reported.

ScanNetV2 contains 1,201/312 training/testing RGB-D images collected from various indoor rooms. These scenes are labeled with 18 object classes for semantic segmentation and instance segmentation. Compared with SUN RGB-D dataset, scenes in this dataset are annotated with more categories and cover larger areas. Point clouds are sampled from the reconstructed meshes. Because the orientation of the bounding box is not annotated, the axis-aligned bounding boxes are predicted, as in VoteNet (Qi et al., 2019).

Evaluation Criteria. Following VoteNet (Qi et al., 2019) and 3D-BoNet (Yang et al., 2019a), the average precision metric AP_{3D} of 3D detection results is adopted as our evaluation criteria. The predicted bounding box B_p is treated as a valid detection result only its 3D overlap area (IoU) between the predicted bounding box B_p and the ground truth bounding box B_{gt} exceeds a certain ratio. IoU is calculated using the following evaluation metric:

$$IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (9)$$

Predicted bounding boxes with 3D IoU results exceeding 0.25 and 0.5 are used to evaluate the detection performance for all classes in both two datasets.

Implementation Details. In our experiments, we implement our model based on VoteNet (Qi et al., 2019), an open-source framework

Table 1
General setting of the backbone network on ScanNetV2 and SUN RGB-D datasets.

Layer	Backbone (Dataset)	#Point	Sampling radius (m)	#neighbors	mlp
SA1(GeoConv)	ScanNetV2	2048	0.2	64	[4,64,64,128]
	SUN RGB-D	2048	0.1	32	[4,64,64,128]
SA2(GeoConv)	ScanNetV2	1024	0.4	32	[128,128,128,256]
	SUN RGB-D	1024	0.2	32	[128,128,128,256]
SA3(PointNet)	ScanNetV2	512	0.8	16	[256,128,128,256]
	SUN RGB-D	512	0.8	16	[256,128,128,256]
SA4(PointNet)	ScanNetV2	256	1.2	16	[256,128,128,256]
	SUN RGB-D	256	1.2	16	[256,128,128,256]
FP1(PointNet)	ScanNetV2	512	–	3	[512,256,256]
	SUN RGB-D	512	–	3	[512,256,256]
FP2(PointNet)	ScanNetV2	1024	–	3	[512,256,256]
	SUN RGB-D	1024	–	3	[512,256,256]

Table 2
Contribution of GeoConv in backbone network on ScanNetV2 and SUN RGB-D datasets.

	SA1	SA2	SA3	SA4	FP1	FP2	mAP@0.25 (%)		mAP@0.5 (%)	
							ScanNetV2	SUN RGB-D	ScanNetV2	SUN RGB-D
PointNet++ (Qi et al., 2019)	PT	PT	PT	PT	PT	PT	58.29	57.22	37.84	33.91
#1GeoConv-PointNet++	GC	PT	PT	PT	PT	PT	57.59	57.52	36.65	33.70
#2GeoConv-PointNet++	GC	GC	PT	PT	PT	PT	**59.14**	**58.40**	**39.13**	**34.91**
#3GeoConv-PointNet++	GC	GC	GC	PT	PT	PT	58.55	57.35	37.67	33.48
#4GeoConv-PointNet++	GC	GC	GC	GC	PT	PT	57.67	56.60	36.92	32.86

Note: #GeoConv-PointNet++: represents the number of PointNet in PointNet++ replaced by our proposed GeoConv in SA modules. PT represents PointNet (Qi et al., 2017a), GC means GeoConv.
The bold values represent the highest performances for both two datasets were achieved when the PointNet in the first two SA modules was replaced by GeoConv while keeping others unchanged.

Table 3
Effectiveness of different scaling parameters on ScanNetV2 and SUN RGB-D datasets.

Scaling parameter	mAP@0.25 (%)		mAP@0.5 (%)	
	ScanNetV2	SUN RGB-D	ScanNetV2	SUN RGB-D
0.05	56.99	56.74	37.99	34.31
0.1	**59.14**	57.32	**39.13**	33.74
0.15	57.91	56.94	38.15	34.32
0.2	58.68	**58.40**	38.56	**34.92**
0.25	58.09	56.61	37.48	33.27

The bold values represent the best scaling parameters' results for ScanNetV2 and SUN RGB-D, respectively.

Table 4
Effectiveness of Object relation learning module on ScanNetV2 and SUN RGB-D datasets.

Relation module	mAP@0.25 (%)		mAP@0.5 (%)	
	ScanNetV2	SUN RGB-D	ScanNetV2	SUN RGB-D
w/2nn	57.12	57.43	36.54	34.08
w/3nn	**59.14**	**58.40**	**39.13**	**34.92**
w/4nn	56.82	56.98	36.63	34.39
w/5nn	57.94	56.93	38.32	32.83
w/6nn	57.54	56.68	37.95	31.91
w/o	57.78	57.71	37.32	33.45

The bold values show the best results of the relation learning module with 3 nearest neighbors on ScanNetV2 and SUN RGB-D datasets.

for 3D object detection built on the PyTorch platform. This framework is composed of three-part: backbone network, Hough voting module, and object proposal and classification module. The backbone network is based on PointNet++ (Qi et al., 2017b), which has several set-abstraction (SA) layers and feature propagation (FP) layers with skip connections. In the first two SA modules, we replace the PointNet (Qi et al., 2017a) with our proposed GeoConv. Our centralization module is similar to the Hough voting module, but we replace the vote loss function with our proposed scalable loss function. The last module is also replaced by our object relation learning module for discriminate object bounding box reasoning and refining. The training epoch is set to 200.

The general setting of our backbone network for these two datasets are listed in Table.1. The input number of points, sampling radius, the number of nearest neighbors, and the mlp output sizes to each layer are introduced. Most hyper-parameters in the same layer of two datasets are similar, only limited parameters are different. Because SUN RGB-D has more sparse point density than ScanNetV2, the sampling radius in SUN RGB-D is reduced to 0.1 and 0.2 in the first two SA layers with the same number of nearest neighbors as 32. Such changes can ensure that the GeoConv in the first two layers extract enough intra-object geometric information in both two datasets.

4.2. Ablation studies

To demonstrate the effectiveness and importance of each proposed individual module, some ablation studies were conducted on both SUN-RGBD and ScanNetV2 datasets. When testing each module, the remaining modules kept unchanged. The followings are the detailed evaluation of these modules.

4.2.1. Contribution of GeoConv in backbone network

As mentioned in the Section 4.1.2, the backbone network is based on PointNet++ (Qi et al., 2017b), which has several SA modules and FP modules with skip connections and PointNet (Qi et al., 2017a) for feature mapping. We replace PointNet in some SA modules with our proposed GeoConv to extract geometric intra-object features for representative points. When testing the effectiveness of GeoConv, the scaling parameter of centralization loss for ScanNetV2 was set to 0.1 and SUN RGB-D was set to 0.2, and the neighboring number in the

Table 5
3D object detection scores per category on the ScanNetV2 (validation) dataset, evaluated with mAP@0.25 and mAP@0.5.

	3DSIS Geo (Hou et al., 2019)	3DSIS 5views(Hou et al., 2019)	VoteNet(Qi et al., 2019)	GRNet (Ours)	3DSIS Geo (Hou et al., 2019)	3DSIS 5views(Hou et al., 2019)	VoteNet(Qi et al., 2019)	GRNet (Ours)
	mAP@0.25 (%)				mAP@0.5 (%)			
cab	19.76	12.75	36.27	**39.45**	5.06	5.73	8.07	**9.76**
Bed	69.71	63.14	87.92	**88.78**	42.19	50.28	76.06	**80.34**
Chair	66.15	65.98	88.71	**89.18**	50.11	52.59	67.23	**71.01**
Sofa	71.81	46.33	**89.62**	88.34	31.75	55.43	68.82	**75.95**
Tabl	36.06	26.91	**58.77**	58.16	15.12	21.96	42.36	**44.55**
Door	30.64	7.95	47.32	**48.46**	1.38	10.88	15.34	**20.58**
Wind	10.88	2.79	**38.10**	32.70	0.00	0.00	6.43	**8.89**
Bkshf	27.34	2.30	44.62	**46.97**	1.44	13.18	28.00	**38.21**
Pic	0.00	0.00	**7.83**	4.94	0.00	0.00	**1.25**	1.22
Cntr	10.00	6.92	56.13	**63.48**	0.00	0.00	9.52	**29.71**
Desk	46.93	33.34	**71.69**	69.81	13.66	23.62	37.52	**49.00**
Curt	14.06	2.47	47.23	**48.46**	0.00	2.61	11.55	**18.42**
Fridg	**53.76**	10.42	45.37	49.06	2.63	24.54	27.80	**34.19**
Showr	35.96	12.17	57.13	**66.37**	3.00	0.82	9.96	**13.44**
Toil	87.60	74.51	**94.94**	94.07	56.75	71.79	86.53	**90.12**
Sink	42.98	22.87	**54.70**	49.70	8.68	8.94	16.76	**20.9**
Bath	84.30	58.66	**92.11**	90.90	28.52	56.40	78.87	**82.57**
Ofurn	16.20	7.05	**37.20**	35.60	2.55	6.87	11.49	**15.49**
mAP	40.23	25.36	58.65	**59.14**	14.60	22.53	33.54	**39.13**

The bold values show the best detection performances for each object class from different algorithms on ScanNetV2 and SUN RGB-D, respectively.

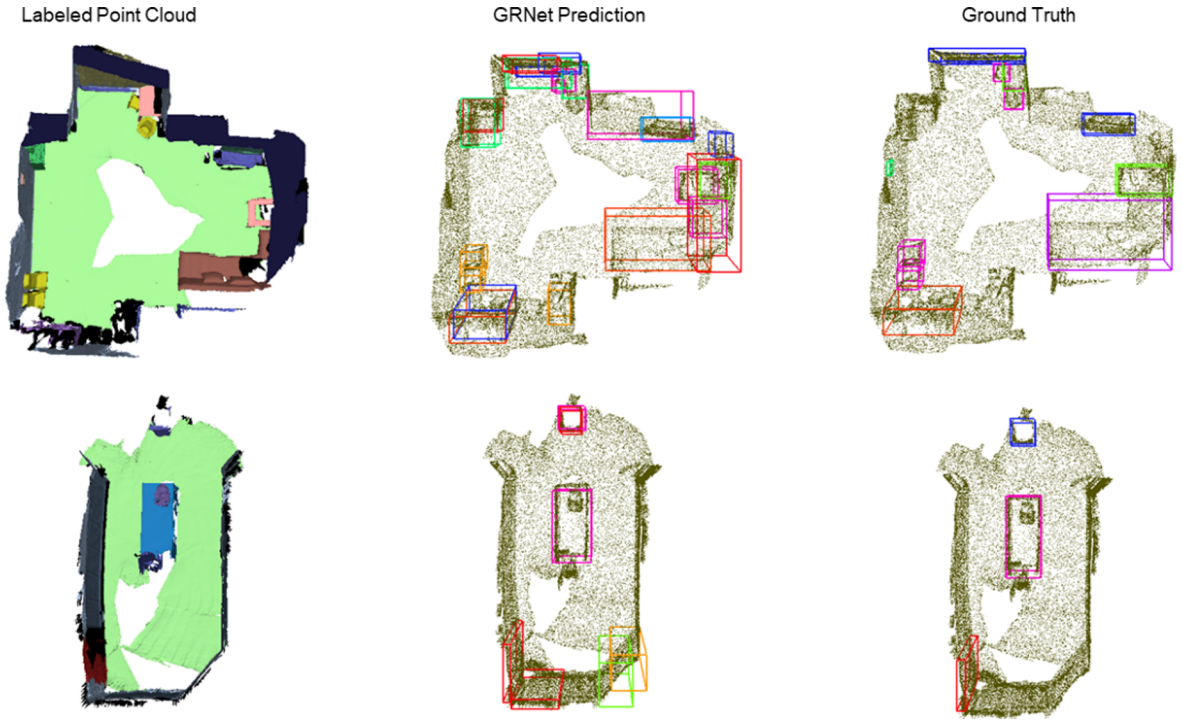

Fig. 4. Qualitative results of 3D object detection in ScanNetV2. From left to right rows: labeled point clouds of the scene, 3D object detection by GRNet, and ground-truth annotations.

object relation learning module for both two datasets was set to 3. We found that, as shown in Table 2, the highest performances for both two datasets were achieved when the PointNet in the first two SA modules was replaced by GeoConv while keeping others unchanged. Because the GeoConv is mainly focused on the intra-object geometric features learning, with an increased sampling ratio, the relation features between those remaining points are increasing. The geometric attributes among these points are weakened. Thus, the performance dropped when replacing more PointNet layers among SA modules with the GeoConv layer.

4.2.2. Comparison of different scaling parameters

In this part, we tested different scaling parameters to see their effectiveness. We selected 0.05, 0.1, 0.15, 0.2 and 0.25 in our experiments, as shown in Table 3. The highest results for ScanNetV2 with 59.14% mAP@0.25 and 39.13% mAP@0.5 were achieved using 0.1, while the best results for SUN RGB-D were accomplished with 58.40% mAP@0.25 and 34.92% mAP@0.5 using 0.2. Because SUN RGB-D has more sparse point density than ScanNetV2, the best scaling parameter for SUN RGB-D was 0.2. The performances for these two datasets with larger or smaller scaling parameters than the parameters with the best results were decreased. The reduced scaling parameter leads to a compact grouping, which causes the contamination of non-object points

Table 6
3D object detection scores per category on the SUN RGB-D (test) dataset, evaluated with mAP@0.25 and mAP@0.5.

Model	Input	Bathtub	Bed	Bookshelf	Chair	Desk	Dresser	Night-stand	Sofa	Table	Toilet	mAP
mAP@0.25 (%)												
DSS (Song and Xiao, 2016)	Geo& RGB	44.2	78.8	11.9	61.2	20.5	6.4	15.4	53.5	50.3	78.9	42.1
COG (Ren and Sudderth, 2016)	Geo& RGB	58.3	63.7	31.8	62.2	45.2	15.5	27.4	51	**51.3**	70.1	47.6
2D-driven (Lahoud and Ghanem, 2017)	Geo& RGB	43.5	64.5	31.4	48.3	**27.9**	25.9	41.9	50.4	37	80.4	45.1
F-PointNet (Qi et al., 2018)	Geo& RGB	43.3	81.1	33.3	64.2	24.7	**32.0**	58.1	61.1	51.1	**90.9**	54
PointFusion (Xu et al., 2018a)	Geo& RGB	37.3	68.6	**37.7**	55.1	17.2	24.0	32.3	53.8	31.0	83.8	45.4
VoteNet (Qi et al., 2019)	Geo only	74.4	83	28.8	75.3	22	29.8	62.2	64	47.3	90.1	57.7
GRNet (Ours)	Geo only	**76.8**	**84.3**	29.3	**76.2**	26.0	26.1	59.2	**64.8**	51.1	90.4	**58.4**
mAP@0.5 (%)												
VoteNet (Baseline)	Geo only	41.4	49.5	5.4	52.3	4.9	12.1	33.9	42.9	18.5	60.5	32.1
GRNet (Ours)	Geo only	41.3	**54.9**	5.0	**55.9**	**5.8**	**14.9**	**36.1**	**46.1**	**24.6**	**63.9**	**34.9**

The bold values show the best detection performances for each object class from different algorithms on ScanNetV2 and SUN RGB-D, respectively.

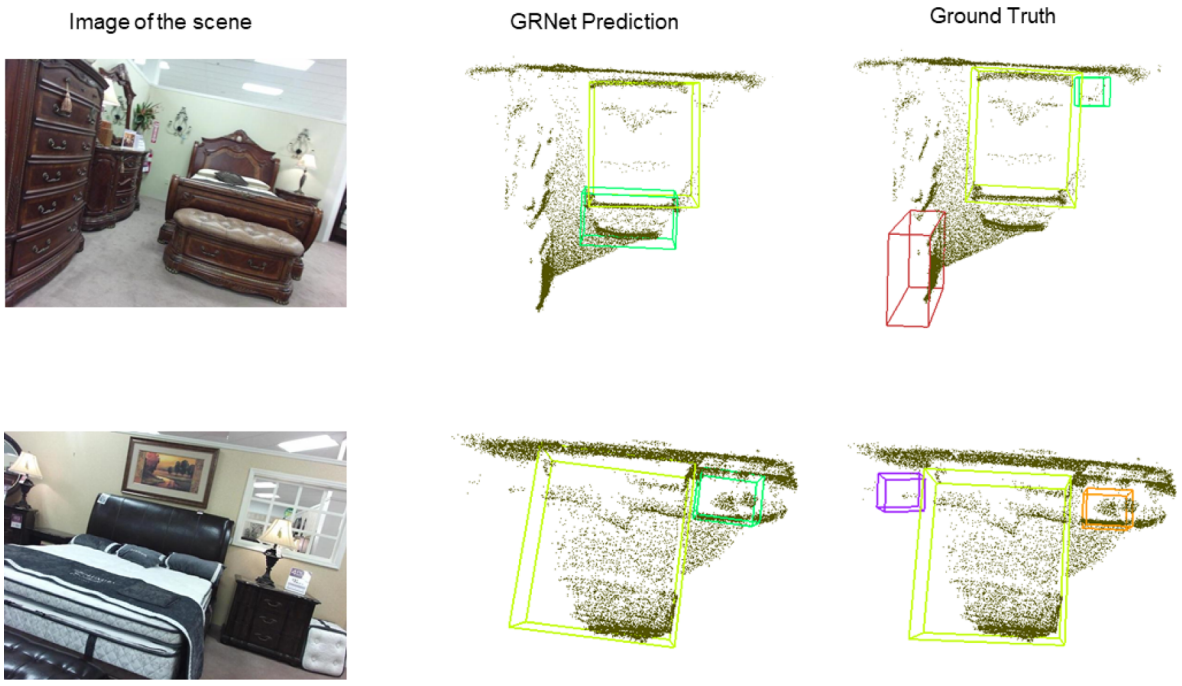

Fig. 5. . Qualitative results on SUN RGB-D. From left to right rows: an image of the scene (not used in GRNet), 3D object detection by GRNet, and ground-truth annotations.

Table 7
Model size and processing time (per frame or scan).

Method	Model size	SUN RGB-D	ScanNetv2
F-PointNet (Qi et al., 2018)	47.0 MB	0.09 s	–
3D-SIS (Hou et al., 2019)	19.7 MB	–	2.85 s
VoteNet (Qi et al., 2019)	11.2 MB	0.10 s	0.14 s
GRNet (ScanNetV2)	17.8 MB	–	0.22 s
GRNet (SUN RGB-D)	13.5 MB	0.10 s	–

in proposal feature pooling. With a larger scaling parameter, the aggregated intra-object feature cannot consume enough effective neighboring features. Thus, detection results decreased.

4.2.3. Effectiveness of object relation learning module

We also tested the contribution of our proposed object relation learning module on ScanNetV2 and SUN RGB-D datasets. As shown in Table 4, without (w/o) the relation learning module, the detection results dropped 1.4% at mAP@0.25 and 1.7% mAP@0.5 on ScanNetV2 and decreased 0.7% mAP@0.25 and 1.8% mAP@0.5 on SUN RGB-D, compared to their best results. Relation learning from 3 nearest neighbor proposals achieved the best results with 59.14% mAP@0.25 and 39.13% mAP@0.5. An increasing number of neighboring proposals may induce more irrelevant features for bounding box reasoning. Thus, the detection performance was weakened. With a reduced number of neighboring proposals, e.g., 2 neighbors, some important relation features are missing. This results in a decreased performance.

4.3. Object detection results

4.3.1. ScanNetV2 detection results

Quantitative detection results of ScanNetV2 are listed in Table.5. GRNet outperforms all previous methods by at least 0.5% mAP@0.25 and 5.5% mAP@0.5 increases. The important improvement mainly comes from mAP@0.5 results. Compared with VoteNet (Qi et al., 2019), our method improves the previous state of the art by more than 20% AP in the category “counter”, 11% AP in “desk”, 10% AP in “bookshelf”, 7% AP in 3 categories such as sink, and 4% AP in the other 8 categories. As illustrated in the ablation studies, the centralization module centralized the surface points in a compact way, which contributes to a more effective proposal feature aggregation. The object relation learning module extracted the useful nearest neighbors feature for better bounding box reasoning. These two modules improve the detection results for mAP@0.5. As for the results at mAP@0.25, GeoConv

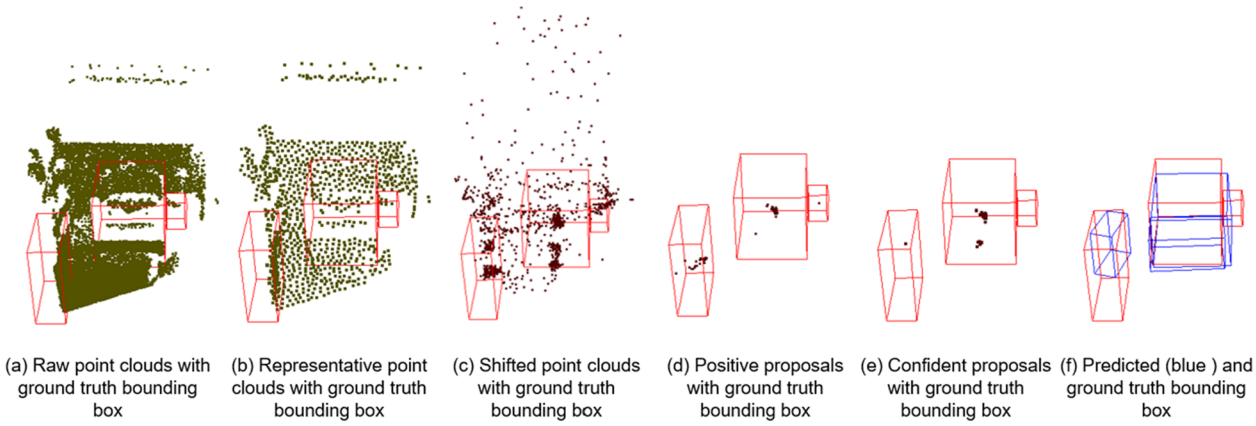

Fig. 6. Staged outputs of GRNet. (a) Raw input of point clouds with ground truth bounding boxes, (b) Representative points with ground truth bounding boxes, (c) Shifted points from the centralization module with ground truth bounding boxes, (d) Positive proposal points with ground truth bounding boxes, (e) Confident proposal points with objectness score (larger than 0.5) and ground truth bounding boxes, and (f) Predicted bounding boxes (blue color) and ground truth bounding boxes.

improves the performance of the representative points' feature by considering both intra-object and inter-object features. Fig. 4 shows some examples of the detection result. Small and shape-similar objects are easy to be mis-detected. There also exists wrong detection in density-compact areas, e.g., corners.

4.3.2. SUN RGB-D detection results

Quantitative results in Table 6 illustrates the detection performance for all classes on SUN RGB-D dataset. GRNet outperforms all previous methods by at least 0.7% mAP@0.25 increase and 2.8% mAP@0.5 increase in SUN RGB-D with point clouds input only. Compared with other detection performances (Song and Xiao, 2016; Ren and Sudderth, 2016; Lahoud and Ghanem, 2017; Qi et al., 2018; Qi et al., 2019), our algorithm can achieve the state-of-art or on-par-with mAP@0.25 detection results on large and geometric-salient objects, such as bed, sofa, bathtub, table and chair. For geometric-weak objects, such as picture and dresser, the improvements are limited. As for detection results on mAP@0.5, our algorithm outperforms the VoteNet (Qi et al., 2019) on 8 categories and on-par-with it on 2 categories. As shown in Fig. 5, the large object with enough scanned point clouds, such as beds, can be detected accurately. However, for thin and density-sparse objects (e.g., bookshelves, desks, and dressers), misdetection occurs commonly. Besides, for shape similar objects, such as tables and nightstands, they are easy to be mis-predicted.

4.4. Optimizer, model size, memory usage and timing

We implemented our model with Python 3.5 and PyTorch 1.0 on one GTX 1080ti GPU. ADAM optimizer (Kingma and Ba, 2014), with an initial learning rate of 0.001, was adopted. The learning rate was decayed at 80, 120, 160 epochs, respectively, with a 0.1 decay rate. The batch size was set to 8 for both training and testing our GRNet-SUNRGB-D and GRNet-ScanNetV2 models. As shown in Table7, the model for GRNet-SUNRGB-D with 20,000 input points has 13.5 M parameters and 17.8 M parameters for GRNet-ScanNetV2 with 40,000 input points. GRNet-SUNRGB-D runs 0.12 s per frame or scan for training, while GRNet-ScanNetV2 runs 0.10 s per frame or scan for training. Because the GRNet (ScanNetV2) has larger model size than VoteNet, its computation cost increases. However, as for GRNet (SUN RGB-D), although it increases around 2 MB model size compared to VoteNet, their computation costs are the same. The main reason is that the GRNet (SUN RGB-D) reduces the search radius and sampling neighbors in the first two SA modules in the backbone network. Such reduction largely relieves the computation burden.

4.5. Detailed analysis of the proposed method

We have tested our GRNet in two indoor environments, which show some differences in point density, room layout, and area. SUN RGB-D has a larger room area, sparser point density, and less labeled objects compared with ScanNetV2. Thus, the application of GeoConv should consider such differences. The sampling radius of GeoConv in the first two SA modules is 0.1 and 0.2 in SUN RGB-D, 0.2 and 0.4 in ScanNetV2, respectively.

In addition, the scaling parameter is also different. Labeled objects in ScanNetV2 are smaller and more compact than SUN RGB-D. As mentioned in VoteNet, voting is only useful for points that are far away from the object center (Qi et al., 2019). Thus, in order to improve the centralization results for small objects, 0.1 scaling parameter was applied as the scaling parameter. However, in SUN RGB-D dataset, labeled objects are larger than ScanNetV2, the best detection result was achieved using 0.2 scaling parameter. We also found that the scaling parameter and relation learning module are more effective in predicting mAP@0.5 bounding box parameters. A compact centralization attributes to neighbors' inter-object and intra-object features learning, which results in a more accurate bounding box prediction. The subgraph (a) in Fig. 6 shows the centralization effects. The further improvement for both mAP@0.25 and mAP@0.5 should consider the RGB information, especially for geometric-weak objects, such as the picture.

Finally, as shown in Fig. 6, those proposals can cover all the labeled objects. However, the post-processing by using NMS based on objectness score and semantic classification score removed low confident proposals (which were actually true positive proposals). Thus, the final detection results missed the true bounding box for objects. From the subgraph (f) in Fig. 6, we can see that the right nightstand is not detected. Additionally, the position of confident proposals affects the predicted bounding box position, which can be seen in the subgraph (e) and (f) in Fig. 6. Thus, how to associate objectness score with the accuracy of the predicted bounding box should be studied in the future to improve our final detection results.

5. Conclusions

In this paper, we have proposed an end-to-end point cloud geometric relation network (GRNet) focused on 3D object detection in indoor scenes. We mainly estimated the oriented 3D bounding boxes (i.e., center, heading angle, and size) and semantic classes of objects. Our network can exploit both intra-object and inter-object features in a bottom-up hierarchical way using our proposed backbone network for representative points. Then, a centralization module with a scalable loss

function was introduced to centralize object points to its center. Proposal points were sampled from these shifted representative points, following a proposal feature pooling operation. Finally, an object-relation learning module was applied to predict bounding box parameters. Such parameters are the additive sum of prediction results from relation-based inter-object features and aggregated intra-object features.

Our model has achieved state-of-the-art 3D detection results with 59.1% mAP@0.25 and 39.1% mAP@0.5 on ScanNetV2 dataset, 58.5% mAP@0.25 and 34.1% mAP@0.5 on SUN RGB-D dataset. Quantitative comparison performance and qualitative results demonstrated the effectiveness of our proposed framework in 3D object detection. However, RGB features are not exploited in this paper, which may contribute to a further improvement for geometric-weak objects. Besides, how to associate the objectness score with the accuracy of the predicted bounding box should be studied in the future to improve the performance of our method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under a discovery grant No. 50474-10033 and the National Natural Science Foundation of China (NSFC) under a grant of No. 41871380. The authors would like to thank Dr. Dunn Emma at the University of Waterloo Writing and Communication Centre for carefully proofreading. Besides, we also would like to thank anonymous reviewers for their insightful comments and suggestions.

References

Chen, K., Lai, Y., Wu, Y., Martin, R.R., Hu, S., 2014. Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information. ACM Trans. Graph. 33 (6), 208. https://doi.org/10.1145/2661229.2661239.
Chen, L., Wang, Q., Lu, X., Cao, D., Wang, F.Y., 2019. Learning driving models from parallel end-to-end driving data set. Proc. IEEE 108, 262–273.
Chen, X., Kundu, K., Zhang, Z., Ma, H., Fidler, S., Urtasun, R., 2016. Monocular 3D object detection for autonomous driving. Proc. IEEE CVPR 2147–2156.
Chen, X., Ma, H., Wan, J., Li, B., Xia, T., 2017. Multi-view 3D object detection network for autonomous driving. Proc. IEEE CVPR 1907–1915.
Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. Proc. IEEE CVPR 5828–5839.
Deng, Z., Jan Latecki, L., 2017. Amodal detection of 3D objects: Inferring 3D bounding boxes from 2D ones in RGB-depth images. Proc. IEEE CVPR 5762–5770.
Engelcke, M., Rao, D., Wang, D.Z., Tong, C.H., Posner, I., 2017. Vote3deep: Fast object detection in 3D point clouds using efficient convolutional neural networks. Proc. ICRA 1355–1361.
Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3D object reconstruction from a single image. Proc. IEEE CVPR 605–613.
Gong, Z., Lin, H., Zhang, D., Luo, Z., Zelek, J., Chen, Y., Nurunnabi, A., Wang, C., Li, J., 2020. A frustum-based probabilistic framework for 3D object detection by fusion of LiDAR and camera data. ISPRS J. Photogramm. Remote Sens. 159, 90–100.
Griffiths, D., Boehm, J., 2019. A review on deep learning techniques for 3D sensed data classification. Remote Sens. 11 (12), 1499. https://doi.org/10.3390/rs11121499.
Gupta, S., Girshick, R., Arbeláez, P., Malik, J., 2014. Learning rich features from RGB-D images for object detection and segmentation. In: Proc. ECCV, pp. 345–360.
Haala, N., Kada, M., 2010. An update on automatic 3D building reconstruction. ISPRS J. Photogramm. Remote Sens. 65 (6), 570–580.
Hou, J., Dai, A., Nießner, M., 2019. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. Proc. IEEE CVPR 4421–4430.
Kanezaki, A., Matsushita, Y., Nishida, Y., 2018. RotationNet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. Proc. IEEE CVPR 5010–5019.
Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv:1412.6980.
Lahoud, J., Ghanem, B., 2017. 2D-driven 3D object detection in RGB-D images. Proc. IEEE CVPR 4622–4630.
Li, B., Zhang, T., Xia, T., 2016. Vehicle detection from 3D LiDAR using fully convolutional network. arXiv:1608.07916.
Li, G., Yang, Y., Qu, X., 2019a. Deep learning approaches on pedestrian detection in hazy weather. IEEE Trans. Ind. Electron., Doi: 10.1109/TIE.2019.2945295.
Li, Y., Ma, L., Zhong, Z., Cao, D., Li, J., 2019b. TGNet: Geometric graph CNN on 3-D point cloud segmentation. IEEE Trans. Geosci. Remote Sens., doi:10.1109/TGRS.2019.2958517.
Lahoud, J., Ghanem, B., 2017. 2d-driven 3d object detection in rgb-d images. Proc. IEEE CVPR 4622–4630.
Lin, D., Fidler, S., Urtasun, R., 2013. Holistic scene understanding for 3D object detection with RGB-D cameras. Proc. IEEE CVPR 1417–1424.
Luo, Z., Li, J., Xiao, Z., Mou, Z.G., Cai, X., Wang, C., 2019. Learning high-level features by fusing multi-view representation of MLS point clouds for 3D object recognition in road environments. ISPRS J. Photogramm. Remote Sens. 150, 44–58.
Mousavian, A., Anguelov, D., Flynn, J., Kosecka, J., 2017. 3D bounding box estimation using deep learning and geometry. Proc. IEEE CVPR 7074–7082.
Qi, C.R., Litany, O., He, K., Guibas, L.J., 2019. Deep Hough voting for 3D object detection in point clouds. arXiv:1904.09664.
Qi, C.R., Liu, W., Wu, C., Su, H., Guibas, L.J., 2018. Frustum PointNets for 3D object detection from RGB-D data. Proc. IEEE CVPR 918–927.
Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep learning on point sets for 3D classification and segmentation. Proc. IEEE CVPR 652–660.
Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep hierarchical feature learning on point sets in a metric space. Proc. NeurIPS 5099–5108.
Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. Proc. NeurIPS 91–99.
Ren, Z., Sudderth, E.B., 2016. Three-dimensional object detection and layout prediction using clouds of oriented gradients. Proc. IEEE CVPR 1525–1533.
Ren, Z., Sudderth, E.B., 2018. 3d object detection with latent support surfaces. Proc. IEEE CVPR 937–946.
Shi, S., Wang, X., Li, H., 2019. PointRCNN: 3D object proposal generation and detection from point cloud. Proc. IEEE CVPR 770–779.
Song, S., Xiao, J., 2014. Sliding shapes for 3D object detection in depth images. Proc. ECCV 634–651.
Song, S., Xiao, J., 2014. Sliding shapes for 3D object detection in RGB-D images. Proc. ECCV 7–7.
Song, S., Xiao, J., 2016. Deep sliding shapes for Amodal 3D object detection in RGB-D images. Proc. IEEE CVPR 808–816.
Song, S., Lichtenberg, S.P., Xiao, J., 2015. Sun RGB-D: a RGB-D scene understanding benchmark suite. Proc. IEEE CVPR 567–576.
Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv:1710.10903.
Wang, C., Hou, S., Wen, C., Gong, Z., Li, Q., Sun, X., Li, J., 2018. Semantic line framework-based indoor building modeling using backpacked laser scanning point cloud. ISPRS J. Photogramm. Remote Sens. 143, 150–166.
Wang, D.Z., Posner, I., 2015. Voting for voting in online point cloud object detection. Proc. Robotics: Sci. Sys. https://doi.org/10.15607/RSS.2015.XI.035.
Wang, L., Huang, Y., Hou, Y., Zhang, S., Shan, J., 2019. Graph attention convolution for point cloud semantic segmentation. Proc. IEEE CVPR 10296–10305.
Wen, C., Sun, X., Li, J., Wang, C., Guo, Y., Habib, A., 2019. A deep learning framework for road marking extraction, classification and completion from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 147, 178–192.
Xiang, Y., Choi, W., Lin, Y., Savarese, S., 2015. Data-driven 3D voxel patterns for object category recognition. Proc. IEEE CVPR 1903–1911.
Xie, S., Liu, S., Chen, Z., Tu, Z., 2018. Attentional ShapeContextNet for point cloud recognition. Proc. IEEE CVPR 4606–4615.
Xu, D., Anguelov, D., Jain, A., 2018a. PointFusion: Deep sensor fusion for 3D bounding box estimation. Proc. IEEE CVPR 244–253.
Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y., 2018b. SpiderCNN: Deep learning on point sets with parameterized convolutional filters. Proc. ECCV 87–102.
Yang, B., Wang, J., Clark, R., Hu, Q., Wang, S., Markham, A., Trigoni, N., 2019a. Learning object bounding boxes for 3D instance segmentation on point clouds. arXiv:1906.01140.
Yang, Z., Sun, Y., Liu, S., Shen, X., Jia, J., 2019. STD: Sparse-to-dense 3D object detector for point cloud. Proc. IEEE CVPR 1951–1960.
Yi, L., Zhao, W., Wang, H., Sung, M., Guibas, L.J., 2019. GSPN: Generative shape proposal network for 3D instance segmentation in point cloud. Proc. IEEE CVPR 3947–3956.
Zhang, W., Xiao, C., 2019. PCAN: 3D attention map learning using contextual information for point cloud based retrieval. Proc. IEEE CVPR 12436–12445.
Zhou, Y., Tuzel, O., 2018. VoxelNet: End-to-end learning for point cloud based 3D object detection. Proc. IEEE CVPR 4490–4499.