SimLOG: Simultaneous Local-Global Feature Learning for 3D Object Detection in Indoor Point Clouds

Mingqiang Wei, *Senior Member, IEEE*, Baian Chen, Liangliang Nan, Haoran Xie, *Senior Member, IEEE*, Lipeng Gu, Dening Lu, Fu Lee Wang, *Senior Member, IEEE*, and Qing Li, *Fellow, IEEE*

***Abstract*— The acquisition of both local and global features from irregular point clouds is crucial for 3D object detection (3DOD). Current mainstream 3D detectors neglect significant local features during pooling operations or disregard many global features of the overall scene context. This paper proposes new techniques for simultaneously learning local-global features of scene point clouds to enhance 3DOD. Specifically, we propose an efficient 3DOD network in indoor point clouds, named SimLOG, which utilizes simultaneous local-global feature learning. SimLOG has two main contributions: a Dynamic Points Interaction (DPI) module to recover local features lost during pooling, and a Global Context Aggregation(GCA) module to aggregate multi-scale features from various layers of the encoder to improve scene context awareness. Unlike traditional local-global feature learning methods, our DPI and GCA modules are integrated into a single feature learning module, making it easily detachable and able to be incorporated into existing 3DOD networks to enhance their performance. SimLOG demonstrates superior performance over twenty competitors in terms of detection accuracy and robustness on both the SUN RGB-D and ScanNet V2 datasets. Specifically, SimLOG boosts the baseline VoteNet by 8.1% of mAP@0.25 on ScanNet V2 and by 3.9% of mAP@0.25 on SUN RGB-D. Code is publicly available at https://github.com/chenbaian-cs/SimLOG.**

***Index Terms*— SimLOG, 3D object detection, dynamic points interaction, global context aggregation.**

Manuscript received 17 March 2023; revised 2 January 2024 and 4 March 2024; accepted 19 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant T2322012, Grant 62172218, and Grant 62032011; in part by Shenzhen Science and Technology Program under Grant JCYJ20220818103401003 and Grant JCYJ20220530172403007; and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022A1515010170. The Associate Editor for this article was Q. Wang. *(Corresponding author: Lipeng Gu.)*

Mingqiang Wei, Baian Chen, and Lipeng Gu are with the School of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China, and also with Shenzhen Institute of Research, Nanjing University of Aeronautics and Astronautics, Shenzhen 518000, China (e-mail: mingqiang.wei@gmail.com; 2116068@nuaa.edu.cn; gulp1224@nuaa.edu.cn).

Liangliang Nan is with the Urban Data Science Section, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: liangliang.nan@tudelft.nl).

Haoran Xie is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, SAR (e-mail: hrxie2@gmail.com).

Dening Lu is with the Department of Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: d62lu@uwaterloo.ca).

Fu Lee Wang is with the School of Science and Technology, The Hong Kong Metropolitan University, Hong Kong, SAR (e-mail: pwang@hkmu.edu.hk).

Qing Li is with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, SAR (e-mail: qing-prof.li@polyu.edu.hk).

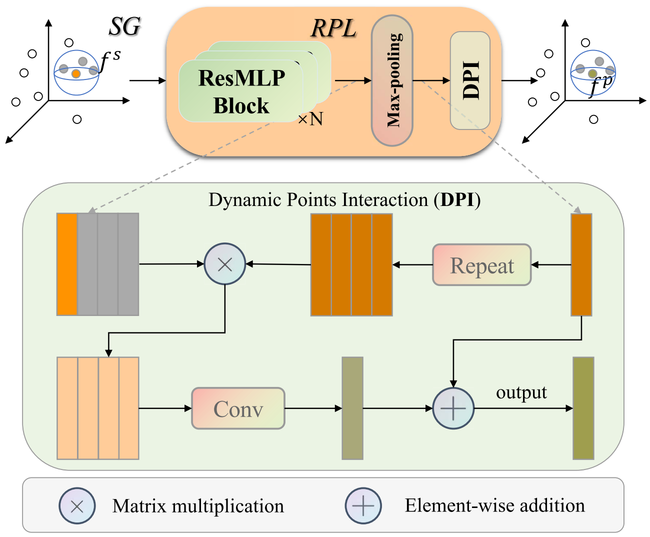

Fig. 1. **The module of Dynamic Points Interaction (DPI)** is designed to preserve local features. DPI enables a seed feature f^p to interact with each point feature in the corresponding point set features f^s to preserve local features. Located within the Residual Points Learning (RPL) module, DPI takes features before and after pooling as input, achieving feature-level interaction to recover potential features lost during the pooling process. Specifically, SG initially downsamples the input point cloud and groups nearby points to form point set features f^s, which are then processed by N ResMLP blocks in RPL. Subsequently, the max-pooling operation is applied to aggregate these features into the seed as point-wise features f^p. The innovative DPI module compensates for any lost features of f^p caused by max-pooling in RPL, given that f^s contains abundant local features.

I. Introduction

Real-world indoor scenes can be digitally captured and efficiently represented by point clouds [1], [2]. However, the captured point clouds are spatially irregular as compared to 2D images. They also exhibit characteristics of sparsity, incompleteness, noise, and outliers, particularly at edges, corners, and occluded regions. Extracting features from these irregular and degraded point clouds tends to weaken the ability of cutting-edging 3D object detection (3DOD) models to accurately locate and identify objects.

Representation learning on point clouds is the first step for indoor 3DOD networks. Currently, four types of representation techniques are widely used, i.e., point-, voxel-, pillar-, and BEV-based methods. Point-based methods [3], [4], [5], [6] take the raw point cloud as input without any transformation. By iteratively downsampling, they can effectively learn local features. Voxel-based methods [7], [8] divide the point cloud into regular 3D voxels and apply 3D CNNs to learn local features. However, the memory and computational costs associated with these methods increase exponentially as the resolution of voxels increases, making it challenging to strike an optimal balance between efficiency and accuracy for these methods. Compared to voxel-based methods, both pillar- [9], [10], [11], [12] and BEV-based [13], [14], [15] methods offer higher efficiency in processing point clouds. Specifically, pillar-based methods simplify voxels into pillars without splitting along the Z-axis, but through discretization in the X-Y plane. They employ max pooling to aggregate features of all points within each pillar into a single “pixel” in a sparse pseudo-image that can be processed by a conventional 2D backbone for 3DOD. BEV-based methods compress voxel features along the Z-axis through a pooling operation or by stacking 3D convolutions with a stride of 2, thereby generating Bird’s Eye View (BEV) features for 3DOD from a top-down perspective. Both types of methods were initially designed for outdoor autonomous driving scenarios, with the assumption that objects are solely distributed across the X-Y plane and with no objects stacked along the Z-axis.

The key to 3DOD is the simultaneous learning of different scales and types of features from scene point clouds, to effectively capture both local geometric details and global scene features (context). The local features aid in the regression of the size and orientation of object bounding boxes, while the global features enhance the classification of objects. Existing point-based 3D detectors primarily rely on the point-based backbone, such as PointNet [16], PointNet++ [17], as well as more recent architectures like PointMLP [18] and PointNeXt [19], to effectively learn point features. As a result, they naturally inherit several demerits of these point-based backbones. First, utilizing PointNet/PointNet++ as backbones leads to the loss of some local features: PointNet/PointNet++ utilizes a simple symmetric function, i.e., max-pooling, to deal with the permutation invariance of point clouds; max-pooling inevitably selects the maximal value in each dimension as the representative feature. This means that some equally important non-maximum features will be lost in each dimension. *In this regard, we propose to design a Dynamic Points Interaction module to preserve such local features.* Second, the global context can effectively describe the semantic information of the entire scene and the correlations between different objects in the scene. However, PointNet/PointNet++ only extracts high-level feature representations by continuously expanding the receptive field while ignoring the global context. The lack of global contextual information negatively impacts the performance of these point-based detectors. The recent PointFormer [20] resorts to Transformer [21] to learn context-aware representations, where multi-head attention depends on a large number of parameters to simulate the long-range dependency. This approach heavily increases the computation and memory demands. *In this regard, we propose to design a Global Context Aggregation module to mine global features.*

We propose a **sim**ultaneous **lo**cal-**g**lobal feature learning paradigm for 3DOD, called SimLOG for short. Inspired by dynamic learning [22], [23], [24], we design a Dynamic Points Interaction (DPI) module to preserve local features during pooling (see Figure 1). In DPI, the input point cloud is first sampled and grouped to form a series of point sets, which are then fed to a Residual Points Learning module. This module comprises several residual MLP blocks and is used to learn the deep feature representation and aggregate these point sets to seeds by the max-pooling operation. The pooled seeds have simplified local context-aware features, while the grouped point sets possess detailed and redundant local geometric features. DPI allows a seed to interact with each point in the corresponding point set to preserve local features. Meanwhile, we observe that with the decreasing number of sampling points, the receptive field of each point in different encoder stages constantly increases. To address this, we design a Global Context Aggregate (GCA) module to concatenate the multi-level features together to represent the contextual guidance. The final extracted features by GCA are therefore aware of the global information. *Unlike the traditional wisdom of local-global feature learning, our DPI and GCA are integrated into a single feature learning module, making it detachable and able to be incorporated into existing 3DOD networks to boost their performance.*

We conduct experiments on the ScanNet and SUN-RGBD datasets [25], [26], and extensive experiments demonstrate the effectiveness of improvement under several evaluation metrics.

In summary, our contributions are as follows:

- We propose a 3DOD network, SimLOG, to simultaneously learn local and global context features. Extensive experiments show clear improvements in our SimLOG over twenty competitors in terms of both numerical and visual evaluations.
- We design three modules, among which DPI and RPL extract rich local geometric information, and GCA captures the global scene context. Ablation experiments show the effectiveness of these modules in promoting the detection performance of SimLOG.
- Both DPI and the feature learning module are detachable and can be incorporated into existing point-based networks to boost their performance.

II. Related Work

A. Feature Learning for 3D Object Detection

1) Local Feature Learning: Local feature extraction can be divided into four categories: point-, voxel-, pillar-, and BEV-based methods. Among these, point-based methods, due to their ability to directly process point clouds without the need for voxelization, pillarization, or BEV transformation, can learn intricate local point cloud features and easily handle stacked objects commonly found in indoor scenes. Leveraging these advantages, point-based methods have become the mainstream backbone for feature extraction in indoor 3D object detectors. Specifically, PointNet [16] and PointNet++ [17] are pioneers in point-based methods, directly utilizing unstructured 3D points and progressively learning point features through symmetric functions and Set Abstraction (SA) layers. PointMLP [18] preserves the simplicity of PointNet++ by avoiding complex local feature extractors. It adopts a feedforward residual MLP structure, efficiently learning local features from point clouds. PointNeXt [19] identifies that the considerable performance improvement in subsequent methods after PointNet++ is primarily attributed to improved training strategies (e.g., data augmentation) and deeper networks, rather than innovations in model architecture. Consequently, introducing inverted residual bottlenecks and separable MLPs to PointNet++, along with proposing more effective training strategies, leads to significant performance gains for PointNet++. PCCN [27] exploits parameterized kernel functions to generalize convolution for learning the non-grid structured data. DGCNN [28] constructs a graph in the local region of sampled points, and dynamically computes message propagation in each layer of the network. These strategies mostly use a pooling operation for feature aggregation to progressively expand the receptive field of the sampled points, resulting in the loss of local features.

2) Multi-scale Feature Learning: The continuous sampling point clouds expand the perception field of each sampled point. By concatenating multi-scale features together as the overall scene information, local features are given awareness of the global context. To encode multi-scale voxel-wise features from feature volumes to the key points, PV-RCNN [29] introduces the VoxelSet Abstraction (VSA) module. MLCVNet [30] incorporates multi-level context information from local point patches to global scenes into VoteNet. HVPR [31] proposes the Attentive Multi-scale Feature Module (AMFM), which refines the hybrid pseudo image to obtain scale-aware features.

3) Joint Learning of Local-and-Global Features: Local features focus on region-detail representations while global features tend to describe the scene context. PointFormer [20] simultaneously extracts local and global features in an encoder by Transformer. MLCVNet [30] captures local and global features in the feature encoding stage and the proposal generating stage, respectively. HyperDet3D [32] utilizes a hypernetwork to learn the scene-conditioned global prior knowledge, which is integrated with local aggregated candidate features to improve representation learning. Similarly, our SimLOG extracts point-wise features from a local and global perspective. Different from existing methods, our simultaneous local-global feature learning is implemented within a feature learning module, and it thus can be seamlessly incorporated into existing models. To be specific, we integrate the side output from multi-level encoding blocks to form the global context, which is more parameter-efficient than PointFormer [20].

B. Indoor 3D Object Detection on Point Clouds

Point clouds have long posed a challenge for feature extraction due to their sparse and irregular characteristics. This has led to the development of various techniques to transform them into regular grid representations. 3D-SIS [33] associates 3D voxel grids and 2D images to extract features. F-PointNet [34] and 2D-driven [35] capture foreground areas from RGB-driven 2D proposals. GSPN [36] generates shape proposals to segment the target area. DSS [37] uses deep sliding shapes to predict 3D bounding boxes on the 3D volumetric scene. COG [38] presents a novel solution with the clouds of oriented gradient descriptors.

The emergence of VoteNet [4] has greatly propelled the development of indoor 3D object detection (3DOD) technology. Built upon PointNet++, it identifies instance centroids by voting from points in a local region, thereby achieving 3DOD. Subsequent advancements [30], [39], [40], [41], [42], [43], [44] in indoor 3DOD have largely evolved from the foundations laid by VoteNet. For instance, MLCVNet [30] proposes three different context learning modules: respectively Patch-to-Patch Context, Object-to-Object Context, and Global Scene Context to capture long-range dependencies at various levels. MFFVoteNet [39] integrates the image feature module into VoteNet to provide robust object class signals, facilitating deterministic detection even in occlusion. EPNet++ [41] introduces a novel cascaded bidirectional fusion mechanism, leveraging rich semantic information absorbed from image features to enhance point features in a cascaded bidirectional interactive fusion manner. This results in a more powerful and discriminative feature representation. DAVNet [42] achieves accurate distribution learning by refining discriminative features through the utilization of an adaptive receptive field. Subsequently, it delivers dependable location scores by further leveraging the distribution of statistical information associated with the localization quality of target objects. SCNet [43] enhances 3DOD performance by promoting semantic consistency between the semantic category of 3D bounding boxes and the categories of all points within the boxes. Following the significant success of transformers in the image domain, 3D object detection methods based on Transformers [20], [45], [46], [47], [48], [49], [50] have made significant advancements. For instance, Pointformer [20] designs a transformer backbone to learn both context-dependent local features and context-aware global representations for 3D object detection. Group-free [45], meanwhile, directly extracts object features from all points by using the self-attention mechanism in transformers. PQ-Transformer [46] achieves both object detection and room layout estimation through transformers for mutual gains. Subsequently, fully convolutional methods [51], [52], [53] are introduced, which improve the efficiency of point cloud processing by using sparse convolutions to directly handle voxelized point clouds. Moreover, some methods [54], [55], [56] ingeniously explore the inherent properties of point clouds, such as geometry and color, to achieve scene perception. For instance, RepSurf [54], inspired by triangle meshes and umbrella curvature in computer graphics, proposes a novel representation of point clouds to explicitly depict the local structure. Point-GCC [55] fully exploits the geometric and color information of point clouds, introducing a 3D scene pre-training framework via geometry-color contrast.

The aforementioned point-based object detection methods mostly use PointNet++ as the backbone to extract features. However, their limited feature learning capacity often restricts them from performing optimally. We propose a novel feature learning paradigm for 3D object detection, which excavates and retains the complete local geometric cues by a dynamic points interaction module and captures the global scene context from different-level feature encoders.

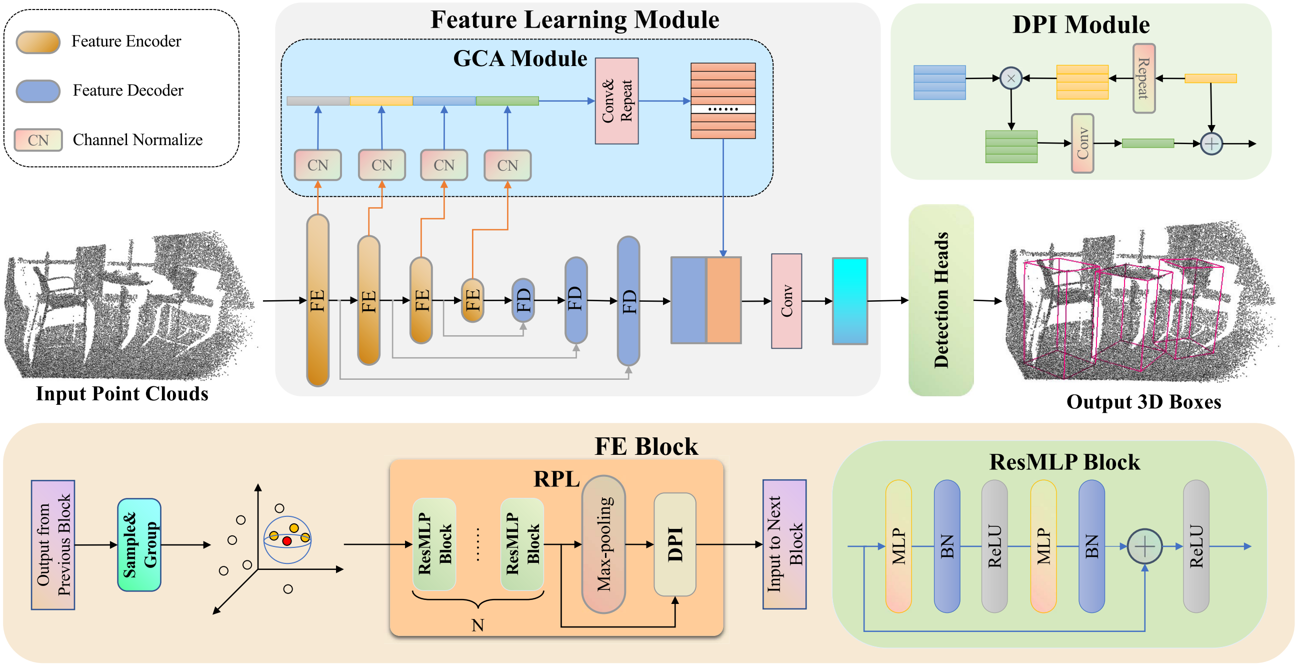

Fig. 2. **The pipeline of our SimLOG.** SimLOG comprises a feature learning module and a detection head. The feature learning module captures local and global point cloud features, serving as input for high-quality 3D object detection in the detection head. Following an encoder-decoder architecture, this module contains a feature encoding (FE) block to learn the high-level feature representations, a Global Context Aggregation (GCA) module to be aware of the scene context, and a feature decoding (FD) block to recover the discarded foreground points for accurate prediction. In each FE block, a Sample-and-Group (SG) module samples the seed points and groups the local region features near the seeds to expand the receptive field of the sampled points. Then, a Residual Points Learning (RPL) module comprises multiple ResMLP blocks, a max-pooling layer, and a Dynamic Point Interaction (DPI) module to further learn and aggregate deep features. The DPI module, situated after the max-pooling layer, takes features both before and after pooling as input, to recover the aggregated local features. The outputs of FE at different levels have various receptive fields, which are concatenated together by a GCA module as the global context to incorporate the global information into point features.

III. Methodology

A. Overview

We propose SimLOG, a 3D object detector that learns both local and global features. It comprises of a Dynamic Points Interaction (DPI) module for local feature preservation, and a Global Context Aggregate (GCA) module for concatenating multi-level features to represent the contextual guidance. As shown in Figure 2, SimLOG mainly consists of a feature learning module and a detection head. Following an encoder-decoder architecture, the feature learning module contains a feature encoding (FE) block to learn the high-level feature representations, a Global Context Aggregation (GCA) module to be aware of the scene context, and a feature decoding (FD) block to recover the discarded foreground points for accurate prediction. Each FE contains a Sample-and-Group (SG) module, a Residual Points Learning (RPL) module, and a Dynamic Points Interaction (DPI) module. SG samples the seed points and groups the local region features near the seeds to expand the receptive field of the sampled points. RPL further learns and aggregates the deep features. DPI bridges SG and RPL to recover the pooled local features. The outputs of FE at different levels have varying receptive fields, which are concatenated together by a GCA module as the global context to incorporate the global information into point features. FD follows the feature propagation module of PointNet++ to recover the discarded foreground points caused by downsampling.

B. Preliminary

MLCVNet [30] demonstrates that the contextual information between different objects makes a significant contribution to object recognition. Hence, it designs three levels of context modules to learn the contextual information in the voting and proposal stages of VoteNet, namely Patch-to-Patch Context (PPC), Object-to-Object Context (OOC), and Global Scene Context (GSC) modules. Besides, PointFormer [20] adopts Transformer to effectively learn context-aware feature representations. Specifically, a pointformer block, consisting of the Local Transformer (LT) module and the Global Transformer (GT) module, substitutes for the SA layer of PointNet++ for feature extraction. However, these methods inevitably lose local features during the pooling stage and fail to recover them for downstream uses.

1) Backbone: VoteNet [4] serves as the baseline of our SimLOG, which consists of a point feature extraction module and a detection head. PointNet++ is the backbone network to extract high-level point features from the input point clouds. The detection head contains a voting module and a proposal module. The voting module takes previous features as input and regresses the offset from each seed point to the corresponding object center by MLPs, emulating the Hough voting process. The proposal module groups the predicted centers to form object candidates and generates the 3D bounding boxes and classified labels.

2) Residual Feature Learning: The residual feed-forward MLPs are effective for feature learning in PointMLP [18]. The Residual Points Learning (RPL) module stacks the residual MLP blocks to learn deeper point representations. As shown in Figure 2, our RPL is formulated as

$$g_i = \mathcal{A}(\phi(f_{i,j}) | j = 1, \dots, K), \tag{1}$$

where $f_{i,j}$ is the feature of the j-th point near the i-th sampled point, $\phi(\cdot)$ denotes the residual MLP block used to capture the deep features. Specifically, a residual MLP block contains the mapping function $MLP(x) + x$, in which $MLP(\cdot)$ is composed of full connection, normalization, and activation layers. The aggregation function $\mathcal{A}$ is the max-pooling operation conducted on the features from the last residual MLP block to aggregate the local region features into the sampled point. Similar to ResNet [57], the residual connections enable MLPs to be easily extended to numerous layers for deeper feature representations.

3) Positional Encoding: MLPs and Fourier Positional Encoding are two common methods for positional encoding, both of which elevate the low-dimensional coordinates of point clouds to higher-dimensional spaces. The key difference lies in that MLPs use learnable one-dimensional convolutions to expand the point cloud dimensionality, while Fourier Positional Encoding employs a heuristic sinusoidal function to map the low-dimension coordinates to the higher-frequency representations. In contrast to MLPs, Fourier Positional Encoding efficiently transforms inputs into a high-dimensional feature space using a genius technique that involves a binary-like encoding through alternating sine and cosine functions. It offers advantages such as high efficiency and low memory consumption. Specifically, the function γ converts the coordinates ($xyz \in [0, 1]$) of the input points to a higher dimensional hypersphere with a set of sine-cosine functions as

$$\delta_i(v) = (a_i cos(2\pi b_i v), a_i sin(2\pi b_i v)), \tag{2}$$

$$\gamma(v) = [\delta_1(v), \dots, \delta_m(v)], v \in \{x, y, z\}, \tag{3}$$

where b_i is the Fourier basis frequency and a_i is the corresponding Fourier series coefficient. For simplicity, we set $a_i = 1$ and generate b_i via a power function $b_i = T^{i/m}$, $i = 0, \dots, m-1$. The results from the Fourier embedding are concatenated together as positional encoding with a dimension of $3 * m$, and are subsequently transformed to match the dimension of the corresponding point features.

C. Dynamic Points Interaction Module

The max-pooling operation leads to the loss of some local geometric features while it is critical to achieve the permutation invariance of point clouds. Prior works, e.g., PointNet [16], PointNet++ [17], and VoteNet [4], primarily extract high-level feature representations by progressively expanding their receptive fields and aggregating local neighborhood features through the max-pooling operation. This method may lead to the loss of local details, especially in the boundary regions of scenes, thereby affecting the detection of small objects (see Figure 5). To counter this, we introduce a Dynamic Points Interaction (DPI) module, which compensates for the feature loss caused by the max-pooling operation without bypassing the max-pooling operation.

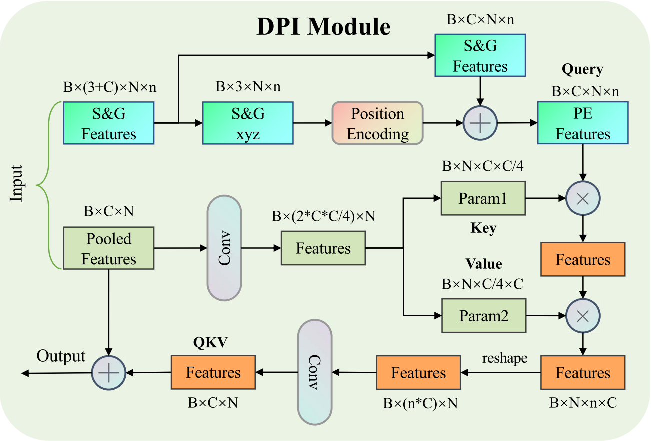

Fig. 3. **Details of Dynamic Points Interaction (DPI).** The input includes both grouped features and pooled features. The grouped features, augmented with positional encoding information, serve as queries, while the pooled features are divided into key-value pairs for the dot-product operation with the queries.

1) Architecture of DPI: The architecture of DPI is illustrated in Figure 3. It takes grouped features and pooled features as the query term and the key-value pair (QKV), respectively. QKV simulates the interaction processing between pooled seeds and grouped sets, and the aggregated features are added to the pooled features to recover the lost features.

2) Implementation Detail of DPI: The input of DPI includes the previous grouped features $F^g \in R^{B\times(3+C)\times N\times n}$ and pooled features $F^p \in R^{B\times C\times N}$. $F^g = \{c_1, c_2, \dots, c_N\}$, where $c_i = \{p_i, p_j, j = 1, \dots, n-1\}$ is a grouped point set in the local region. p_i is a sampled point as the centroid of the set and p_j is a neighboring point of p_i within a given radius. Let $\{x_i, f_i\}_t$ denote the point p_i in the t-th point set, where $x_i \in R^3$ represents the coordinates and $f_i \in R^C$ denotes the features of the points. Subsequently, the Positional Encoding (PE) module takes x_i as input and transforms its dimension to the same of f_i and adds x_i to f_i in an element-wise manner for generating the queries f_q. This process is formulated as

$$f_q = p_f \oplus PE(p_{xyz}), \tag{4}$$

where p_{xyz} and p_f represent the coordinates and features of points respectively.

The pooled features F^p first expand the dimension from C to $2 * C * C/m$ by a convolution layer. Then, these features are equally split into key-value pairs in the feature channel dimension, with the keys being $f_k(: \ C * C/m)$ (Param1 in Figure 3) and the values $f_v(C * C/m \ :)$ (Param2 in Figure 3) respectively. A reshaping operation is applied on the QKV features to rearrange the feature dimension ($f_q \in R^{B\times N\times n\times C}, f_k \in R^{B\times N\times C\times(C/m)}, f_v \in R^{B\times N\times(C/m)\times C}$) for the proceeding of Dot-Product between queries and key-value pairs. To improve efficiency, we employ a bottleneck structure between the key and value, with the number of feature channels being reduced by a factor of m (we set m=4). This entire process is formulated as

$$y = RB(RB(f_q \odot f_k) \odot f_v), \tag{5}$$

$$o = R(y + F_p), \tag{6}$$

where R and B denote the activation function and the normalization function, respectively.

Remark 1: Similar to the attention mechanism [21] that excels at modeling dependencies between input sequences, DPI can be viewed as a mapping from the query term (i.e., grouped features) and key-value pairs (i.e., pooled features) to the output. Specifically, each element in the pooled features represents the corresponding feature group before pooling. To ensure interaction between the pooling seed and each point in the group, we expand the pooled features to the same size as the grouped features. We then evenly split them into key-value pairs. Compared to separate mapping functions in the attention mechanism [21], this not only reduces the number of parameters but also increases the correlation between keys and values since they directly come from the same pooled features. The non-pooled grouped features serve as queries to explore crucial detailed features from the pooled features. These explored features then act as additional enhanced features, added back to the original pooled features. This process strengthens the preservation of detailed features within the pooled features, effectively mitigating the feature loss caused by pooling.

Remark 2: The feature extraction modules in [20] and [30] rely on the sophisticated feature extractor to excavate the local geometric information by using an attention mechanism. However, this method is limited in its ability to capture all of the crucial information contained within the local features. The succeeding aggregation function (e.g., max-pooling) still inevitably discards some crucial features. We give full consideration to this issue. The grouped set has redundant and comprehensive local features, particularly including the part of features neglected by the pooled seeds. Hence, we take the pooled seed to continually interact with each point in the corresponding grouped set to attain the completed local geometric information.

D. Global Context Aggregation Module

The global context describes the semantic information of a whole scene, which is of great importance in inferring the classes of objects due to the close connection between the scene and objects. Prior works, e.g., PointNet [16], PointNet++ [17], VoteNet [4] and GCPANet [63], learn rich high-level feature representations by progressively expanding their receptive fields and aggregating local neighborhood features through pooling. However, these methods either overlook global context [4], [16], [17] or solely rely on high-level features containing plentiful semantic information to generate global features, disregarding the low-level features from the shallow layers that may contain local geometric cues [63].

Hence, we propose the global context aggregation (GCA) module, which fuses multi-layer features to serve as global context guidance, to enhance the ability of feature representations for 3D bounding box regression and object classification. Specifically, we first conduct the channel normalization (CN) to the outputs of each feature encoding block. This operation is to compress the number of feature channels to k for subsequent concatenation. The formulation of CN is summarized as

$$CN(f) = MaxP(MLP(f)), \tag{7}$$

where $MaxP(\cdot)$ stands for the max-pooling operation.

To address the issue of the inconsistent number of sampled points from different encoders, the max-pooling function is applied to compress the features to a 1D vector. Subsequently, these vectors representing respective encoders are concatenated together as the global context by

$$g = MLP(Cat[CN(f_i)]), i = 1, 2, 3, 4. \tag{8}$$

The global context representations not only facilitate message propagation among different objects in the scene but also benefit the inference in object classification.

IV. Experiments

This section validates the proposed SimLOG in two indoor datasets and compares it with the state-of-the-art 3DOD methods. In Section IV-A, we introduce the two datasets and the training details of SimLOG. In Section IV-B, we show the comparison results of SimLOG and its competitors. In Section IV-C, we analyze the effectiveness of each component in SimLOG through comprehensive ablation studies. In Section IV-D, we introduce the application of DPI in existing models, followed by the discussion of the limitation of SimLOG in Section IV-E.

A. Datasets and Training Details

We evaluate SimLOG and its competitors in two indoor datasets, i.e., SUN RGB-D [26] and ScanNet V2 [25].

SUN RGB-D [26] is a single-view RGB-D dataset for 3D scene understanding. It contains 5050 indoor RGB and depth images annotated with amodal-oriented bounding boxes of 37 object categories for training, and the rest 5285 RGB-D images for testing. Before being fed into the network, depth images are first converted to point clouds by the provided camera parameters. The evaluation metric is the standard mean Average Precision (mAP), and the evaluation is conducted on the 10 most common categories.

ScanNet V2 [25] is a densely annotated dataset consisting of 3D reconstructed meshes, which have rich textures, and semantic and geometric information. It contains 1513 indoor scenes captured from hundreds of different rooms, with semantic and instance labels for all the points, as well as 3D object bounding boxes. Compared to the fragmentary scan in SUN RGB-D, the scenes of ScanNet are larger and more comprehensive, and local geometric details of objects are well captured. The vertices of meshes in the dataset are sampled as point clouds.

1) Data Augmentation: To reduce computational complexity, we down-sample each point cloud using the farthest point sampling (FPS) with 20,000 points for SUN RGB-D and 40,000 points for ScanNet. The height attribute of each point is included as an extra feature to feed into the network. Following VoteNet [4], we apply randomly flipping, rotating, and scaling operations to the point clouds to augment the training data.The flipping probability in both horizontal and vertical directions is set to 0.5. The range of rotation angle is set to $\pi/6$. The range of scale ratio is set between 0.85 and 1.15.

2) Training Details: Our model is implemented with PyTorch on an NVIDIA GeForce RTX 3060 GPU and optimized by the Adam optimizer in an end-to-end manner. For ScanNet V2, we set the initial learning rate to 1e-3 and weight decay to 1e-1. The total training epochs are 48, and the learning rate gradually decreases in the 12, 24, and 36 epochs by 5×. For SUN RGB-D, we set the base learning rate to 1e-3 and weight decay to 5e-2. The total epochs are 36, and the learning rate steadily decreases in the 12 and 24 epochs by 5×.

B. Comparisons

We compare SimLOG with its competitors of 3D-2D query-based 3DOD methods [33], [34], [35], [36], [38], voting-based methods that excavate informative local representations, i.e., VoteNet [4] and its variants [56], [60], [61], [64], and attention-based methods [20], [30], [59], [65] that explore the relationships between the local objects and point clusters. The results are reported in Table II, Table II and Table III.

1) Quantitative Results: The overall quantitative results on SUN RGB-D and ScanNet V2 datasets are reported in Table I and II, and the results for single categories on ScanNet V2 are reported in Table III.

Observing the experimental results on SUN RGB-D as shown in Table I, our SimLOG achieves state-of-the-art performance in mAP@0.25 (61.6%). Specifically, when compared to methods such as CorrelaBosst [40], MFFVoteNet [39], EPNet++ [39], DAVNet [42], and SCNet [43], all of which use VoteNet [4] as the baseline, our method consistently outperforms them in both mAP@0.25 and mAP@0.50 metrics. Notably, our method demonstrates significant improvement compared to SCNet [43], achieving a noticeable increase of +1.7% in mAP@0.25.

The experimental results on ScanNet V2 reported in Table II further demonstrate the superior performance of our method. When compared to other methods, including ImVoxelNet [8], CorrelaBoost [40], MFFVoteNet [39], AShapeFormer [44], and SCNet [43], all of which use VoteNet [4] as a baseline, our method maintains state-of-the-art performance in both mAP@0.25 and mAP@0.50 metrics. Notably, SimLOG demonstrates a significant advantage over SCNet [43], achieving a substantial improvement of +3.4% in mAP@0.25 and +8% in mAP@0.50.

TABLE I

3D Object Detection Results on the SUN RGB-D Validation Set. Mean Average Precision With 3D IoU Thresholds of 0.25 and 0.5 Is Used for Evaluation. The Bold Text Means the Best Result. ∗ Denotes That These Methods Use VoteNet as the Baseline

SUN RGB-D	mAP@0.25	mAP@0.5
DSS [37]	42.1	-
COG [38]	47.6	-
F-PointNet [34]	54.0	-
VoteNet [4]	57.7	32.0
H3DNet [56]	60.1	39.0
3DETR [58]	59.1	32.7
RGNet [59]	59.2	-
MLCVNet [30]	59.8	-
PointFormer [20]	61.1	36.9
ImVoxelNet [8]	40.7	-
CorrelaBoost* [40]	61.0	37.7
MFFVoteNet* [39]	59.3	-
EPNet++* [41]	61.5	-
DAVNet* [42]	60.3	**39.4**
SCNet* [43]	59.9	-
Ours*	**61.6**	38.9

TABLE II

3D Object Detection Results on the ScanNet V2 Validation Set. Mean Average Precision With 3D IoU Thresholds of 0.25 and 0.5 Is Used for Evaluation. The Bold Text Means the Best Result. ∗ Denotes That These Methods Use VoteNet as the Baseline

ScanNet V2	mAP@0.25	mAP@0.5
DSS [37]	15.2	6.8
F-PointNet [34]	19.8	10.8
GSPN [36]	30.6	17.7
3D-SIS [33]	40.2	22.5
VoteNet [4]	58.6	33.5
HGNet [60]	61.3	34.4
DOPS [61]	63.7	38.2
RGNet [59]	48.5	26.0
MLCVNet [30]	64.7	42.1
3DETR [58]	65.0	47.0
PointFormer [20]	64.1	42.6
ImVoxelNet [8]	48.1	-
ImGeoNet [62]	54.8	28.4
CorrelaBoost* [40]	61.0	41.2
MFFVoteNet* [39]	63.9	41.1
AShapeFormer* [44]	66.6	47.8
SCNet* [43]	63.3	40.5
Ours*	**66.7**	**48.5**

In Table I and Table II, it is also observed that our SimLOG operates smoothly on the two datasets, while its competitors may not perform consistently between them. For example, MLCVNet [30] performs well on ScanNet V2 but yields poor results on SUN RGB-D; RGNet [59] performs better on SUN RGB-D than on ScanNet V2. This is because ScanNet V2 consists of reconstructed meshes that cover complete objects

TABLE III

COMPARISON OF SIMLOG WITH ITS COMPETITORS IN THE **SCANNETV2** VALIDATION SET. MAP@0.5 IS USED FOR EVALUATION

Methods	cab	bed	chair	sofa	table	door	win	bkshf	pic	cntr	desk	curt	fridg	showr	toil	sink	bath	ofurn	mAP
VoteNet [4]	8.1	76.1	67.2	68.8	42.4	15.3	6.4	28.0	1.3	9.5	37.5	11.6	27.8	10.0	86.5	16.8	78.9	11.7	33.5
MLCVNet [30]	16.6	**83.3**	78.1	74.7	55.1	28.1	17.0	**51.7**	3.7	13.9	**47.7**	28.6	36.3	13.4	70.9	25.6	85.7	27.5	42.1
PointFormer [20]	19.0	80.0	75.3	69.0	50.5	24.3	15.0	41.9	1.5	26.9	45.1	30.3	41.9	25.3	75.9	35.5	82.9	26.0	42.6
3DETR [58]	**23.4**	79.4	76.5	77.8	53.0	27.7	19.7	41.8	6.1	28.8	46.8	30.7	37.8	**30.1**	**96.0**	30.2	84.4	28.3	47.0
Ours	21.8	80.9	**79.4**	**86.4**	**57.8**	**32.7**	**24.3**	41.2	**10.8**	**33.4**	42.0	**38.8**	**50.3**	21.6	90.4	**36.1**	**88.3**	**35.7**	**48.5**

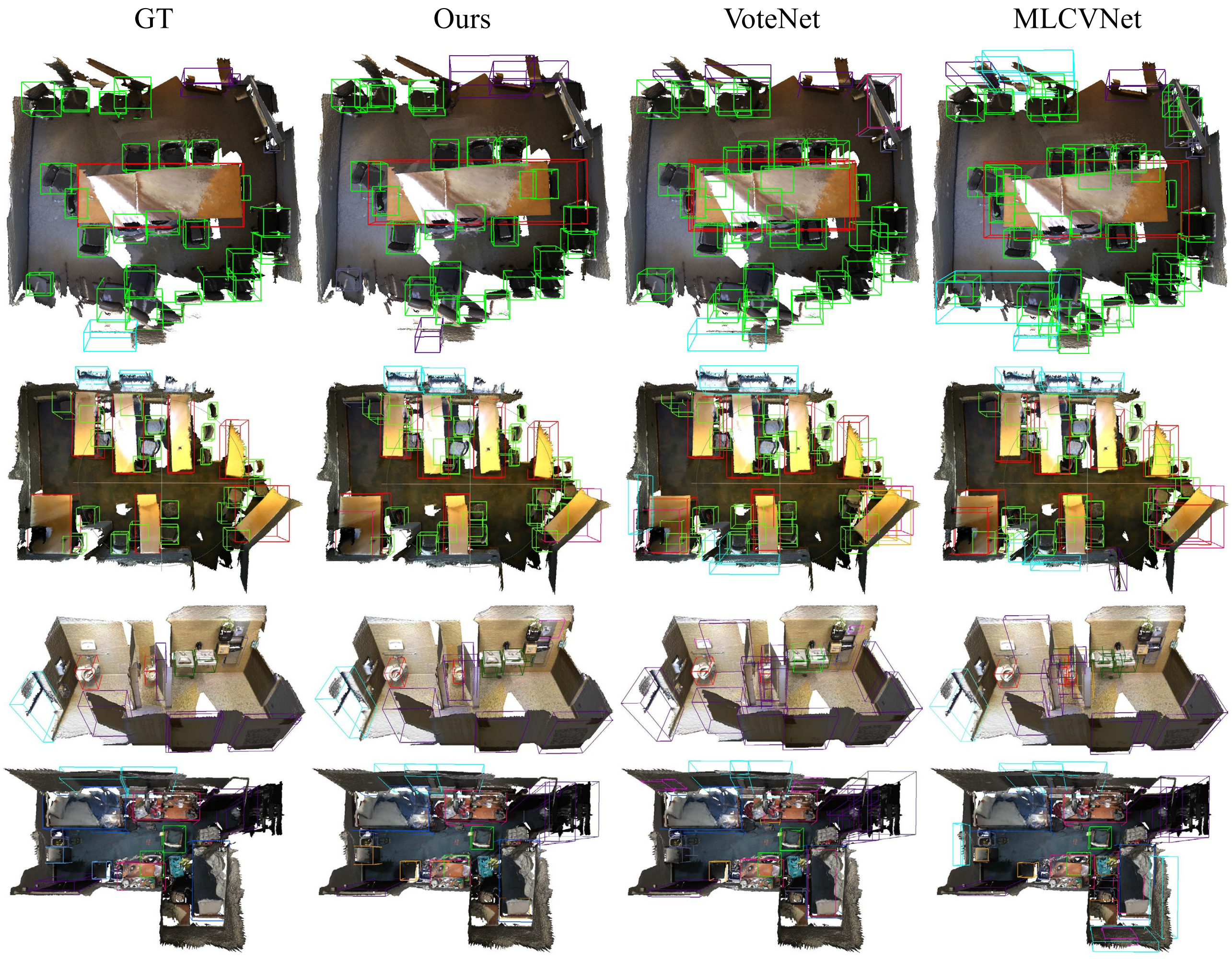

Fig. 4. Visualization comparison of 3D object detection methods in ScanNet V2. Our method leads to less false detection than VoteNet and MLCVNet.

in larger areas, while SUN RGB-D contains only single-view RGB-D images where point clouds projected from the depth map include partial objects and smaller areas. The disparate characteristics may result in the incapability of numerous methods to be reliable in both datasets.

In Table III, it is observed that our method attains superior performance in 12 out of the total 18 categories in ScanNet V2 under mAP@0.5. For example, both the picture category (pic) and the window category (win) contain shallow objects that are inset into large walls. Most of the competitors find it difficult to detect these shallow objects, since the pooling operation aggregates too many features from the background (i.e., wall) while discarding important target features. Fortunately, our dynamic points interaction module preserves the target features effectively. Compared to 3DETR [58], our method improves the detection accuracy by 4.6% and 4.7% of mAP on windows and pictures respectively.

2) Qualitative Results: We visualize the representative detection results from ScanNet V2 and SUN RGB-D in Figure 4 and Figure 6, from which we observe that VoteNet [4] and MLCVNet [30] demonstrate false detection regarding the object's number and category. For example, in the first row of Figure 4, VoteNet [4] and MLCVNet [30] wrongly recognize that many chairs are on the table and wall. In contrast, our method produces more accurate bounding boxes in terms of both the location and category.

C. Ablation Study

1) Residual Points Learning Module: We first evaluate the effect of the number of ResMLP blocks in the Residual Points

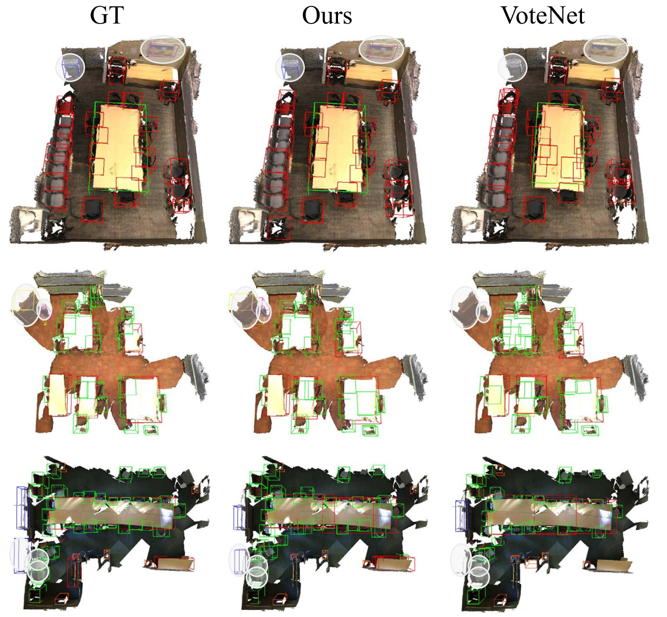

Fig. 5. Visualization comparison showcasing the effect of DPI on recovering lost local features of small objects due to pooling in ScanNet V2.

TABLE IV

Ablation Study About the Number of ResMLP Blocks in RPL. '0*' Means Using the Traditional MLPs

ResMLP	ScanNet V2	
	mAP@0.25	mAP@0.5
0* block	63.9	45.4
1 block	64.9	47.1
2 blocks	**66.7**	**48.5**
3 blocks	65.2	47.0

Learning (RPL) module. We modify the depth of RPL by setting the number of ResMLP blocks to 0, 1, 2, and 3, respectively. 0 means using the traditional MLP layer for feature extraction. The experiment results are reported in Table IV, from which we observe an increase in detection performance as RPL becomes deeper. However, increasing the number of ResMLP blocks would not always lead to better performance. When setting the number of ResMLP blocks to 3, the detection accuracy decreases by 1.5% of $mAP@0.25$ and 1.5% of $mAP@0.5$. Thus, two ResMLP blocks achieve the best performance.

2) Dynamic Points Interaction Module: DPI is the essential component in our model, which significantly improves detection accuracy. The quantitative results are reported in Table V. We can see that without DPI, the performance drops 3.1% and 3.8% in terms of $mAP@0.5$ in ScanNet V2 and SUN RGB-D, respectively. Visualization of the object detection results is in Figure 8. After removing DPI, multiple chairs (green boxes) instead of the table (red boxes) are incorrectly detected. This is due to that the pooling operation aggregates features from the neighboring regions instead of object features. The sampled points of the table integrate with the points from the chairs beside the table, resulting in the mistake of recognizing the table as multiple chairs. Additionally, DPI also aids in the recovery of local features for small objects, especially those situated at the boundaries of the scene. This enhancement significantly contributes to improved detection performance. (see Figure 5). Thus, our DPI enables the grouped features to engage with the pooled features to preserve local features and thus ensures correct table detection.

TABLE V

Ablation Study About DPI and GCA. '-DPI' and '-GCA' Indicate SimLOG Without DPI and GCA Respectively

Methods	ScanNet V2		SUN RGB-D	
	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
VoteNet	58.6	33.5	57.7	32.9
-DPI	64.7	45.4	58.5	35.1
-GCA	65.3	46.1	60.1	37.6
SimLOG	**66.7**	**48.5**	**61.6**	**38.9**

TABLE VI

Ablation Study About DPI and Self-Attention

Methods	ScanNet V2		SUN RGB-D	
	mAP@0.25	mAP@0.5	mAP@0.25	mAP@0.5
self-attention	64.7	45.0	60.1	36.3
DPI	**66.7**	**48.5**	**61.6**	**38.9**

3) DPI vs Self-attention: Self-attention takes the same or similar features as input while DPI takes grouped and pooled features as the query term and key-value pair. As shown in Table VI, when we substitute DPI with self-attention and apply it to the pooled features, we observe a performance decrease of 3.5% and 2.6% under $mAP@0.5$ on ScanNet V2 and SUN RGB-D, respectively. This decrease can be attributed to the fact that self-attention primarily explores internal relationships within features and tends to overlook external cues crucial for compensating for feature loss.

4) Global Context Aggregation Module: GCA plays a substantial role in learning the global contextual information for 3D object detection. As shown in Table V, removing GCA causes the detection accuracy to decrease by 2.4% and 1.3% under $mAP@0.5$ in ScanNet V2 and SUN RGB-D, respectively. The visualization results are shown in Figure 8. The fridge near the sink is erroneously detected as a door by the model without GCA. The global scene context encodes the multi-scale features to generate the scene context information that helps to enhance object detection.

5) Efficiency Analysis: For 3D object detection, balancing efficiency and accuracy is crucial. In this analysis, we emphasize the inference runtime and model parameters of our method. As indicated in Table VII, the GCA module excels by necessitating a minimal number of parameters (0.2 Mb) while delivering satisfactory performance. On the other hand, the DPI module introduces 4.1 Mb of parameters and 29 ms of runtime, yet it markedly enhances the accuracy of 3DOD. In conclusion, when utilizing both DPI and GCA, our method introduces an additional 4.3 Mb of parameters and 34 ms of extra runtime compared to the baseline VoteNet. Nevertheless, it leads to a significant performance improvement, especially in comparison to the transformer-based method

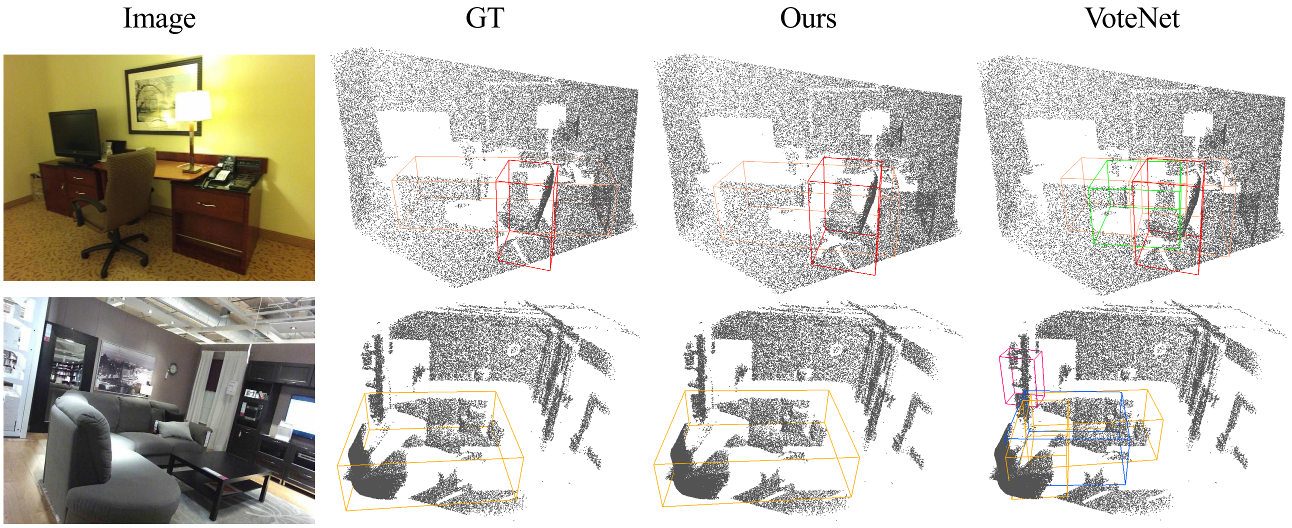

Fig. 6. Visualization comparison of 3D object detection methods in SUN RGB-D. Our method leads to less false detection than VoteNet.

TABLE VII

INFERENCE RUNTIME (IR) AND MODEL PARAMETERS (MP) OF DIFFERENT MODULES ON THE SCANNET V2 VALIDATION SET

ScanNet V2	mAP@0.25	mAP@0.5	IR (ms)	MP (Mb)
VoteNet [4]	58.6	33.5	120	3.6
PointFormer [20]	64.1	42.6	286	12.6
Ours (w/o DPI)	64.7	45.4	126	3.8
Ours (w/o GCA)	65.3	46.1	149	7.7
Ours	**66.7**	**48.5**	154	7.9

TABLE VIII

COMPARISON OF THE ORIGINAL VERSION OF CUTTING-EDGE MODELS AND THEIR IMPROVED VERSION WITH DPI

ScanNet V2	mAP@0.25	mAP@0.5
Group-Free [45]	68.2	52.6
Group-Free+DPI	68.9	54.6
PQ-Transformer [46]	66.9	54.8
PQ-Transformer+DPI	67.7	55.8

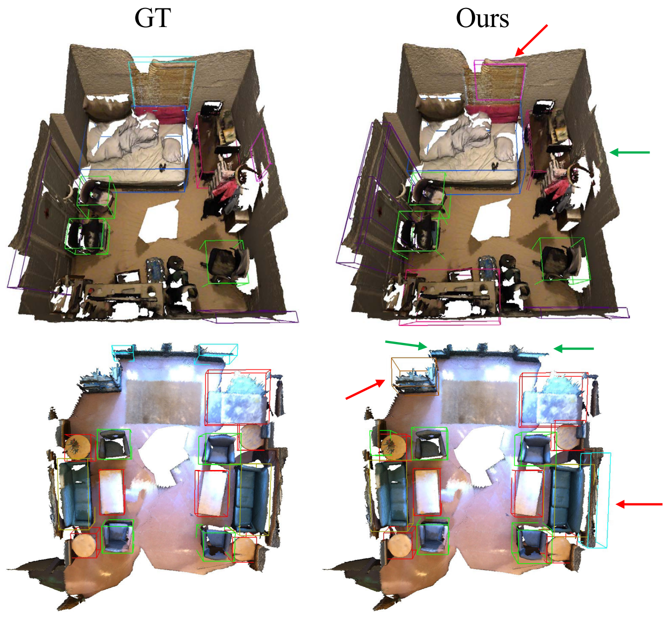

Fig. 7. Failure cases in ScanNet V2. The red and green arrows denote the false positive bounding boxes and the missed objects respectively.

PointFormer [20]. It is important to highlight that the inference runtime and model parameters of our method are notably lower than those of PointFormer.

D. Generalizability

Our DPI is a standalone module and can be conveniently integrated into existing models to improve performance. As can be seen from Table VIII, integrating our DPI module into Group-Free [45] and PQ-Transformer [46] has led to significant performance gain.

E. Limitations

Despite showing promising improvement across two benchmark datasets, SimLOG struggles to achieve accurate object detection in several challenging scenarios in Figure 7. The common failures are false-positive bounding boxes of objects (see the red arrows) and missed object detection (see the green arrows). The most difficult objects to detect are often those that are extraordinarily slim and stick close to the wall, such as the picture and window in the smooth wall example shown in Figure 7. The false-positive bounding boxes also tend to occur when multiple objects share similar shapes and features, making it difficult for the global context to differentiate between them. It is worth noting that these types of issues plague a majority of existing methods.

Moreover, multi-modal large models such as GPT-4V (GPT-4 with vision) have seen rapid development recently. However, our method is still limited to single-modal point clouds. Compared to these models, our method lacks a thorough exploration of the intrinsic geometric properties of point

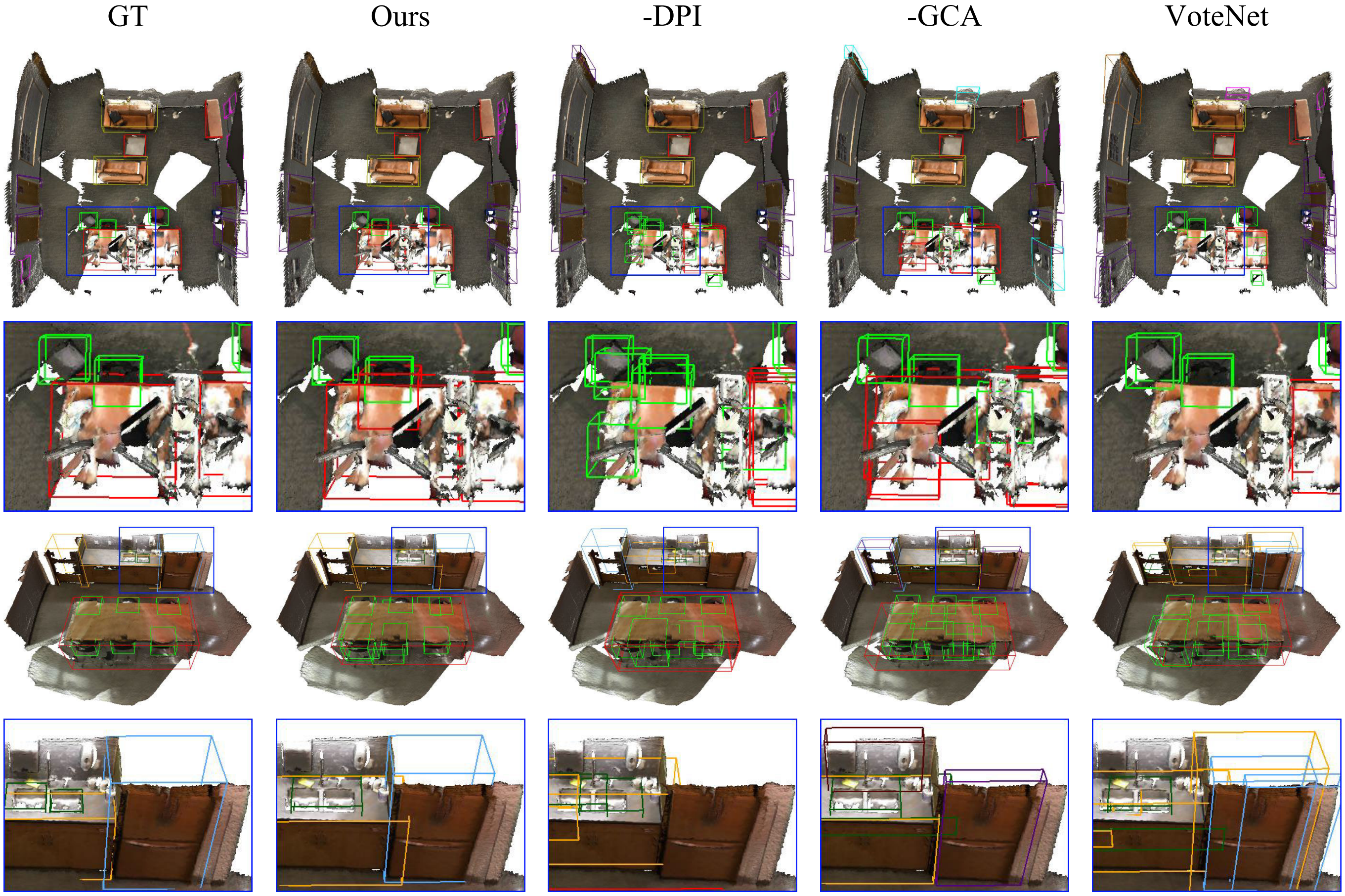

Fig. 8. Visual results of ablation study in ScanNet V2. '-DPI' and '-GCA' denote SimLOG without DPI and GCA respectively. The first and third rows demonstrate the whole scene, and the second and fourth rows are close-up views.

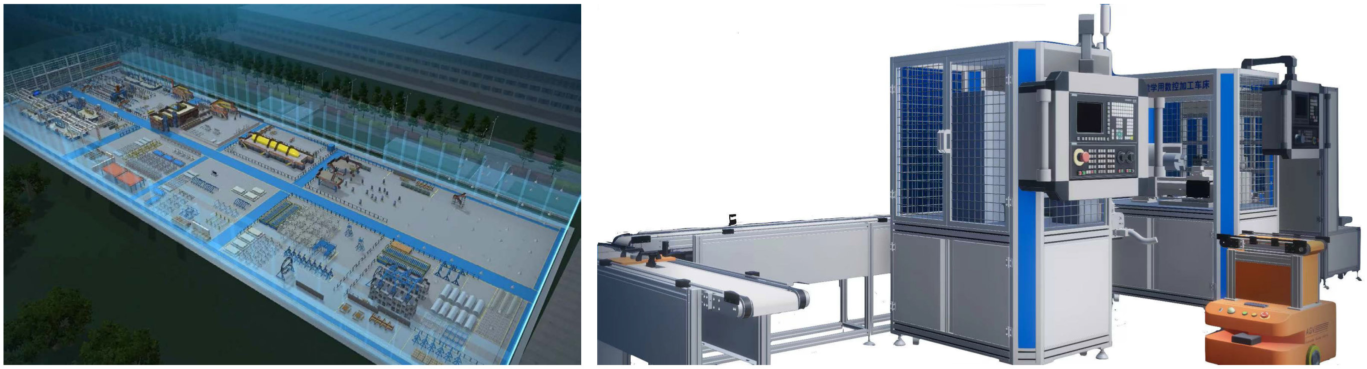

Fig. 9. **SimLOG serves our digital twins system well.** The system is developed by the authors and SimLOG can detect the main objects in the scene for subsequent decisions.

clouds, such as the curvature, normals, and even triangle meshes and umbrella curvature within the local surface.

F. Applications and Ethical Considerations

Our indoor 3D object detection method, SimLOG, has wide-ranging applications in real-world scenarios, especially in indoor robotics, such as floor-cleaning robots, hotel delivery robots, etc. It is also applicable in the industrial sector, where it can be employed by robotic arms for tasks such as automatically grasping items on workstations or handling parts in assembly processes. Specially, SimLOG serves our digital twins system well. The system (see Figure 9) is developed by the authors and SimLOG can detect the main objects in the scene for subsequent decisions.

In these applications, the primary ethical concern arises when robots experience malfunctions, potentially leading to unexpected harm to humans. In such situations, determining responsibility becomes highly challenging, as it is unclear whether the malfunction is due to a failure in the 3D object detection model or mechanical issues, making it difficult to attribute responsibility to either the machine manufacturer or the user.

V. Conclusion

It is fundamental to capture both local and global features of irregular point clouds for 3D object detection (3DOD). Mainstream 3D detectors, e.g., VoteNet and its variants, either discard a considerable amount of local features during pooling operations or ignore global features of the whole scene context. Thus, the models often generate a large number of false positives and false negatives. In this paper, we present SimLOG, a local-global feature learning approach to upgrade voting-based 3DOD networks. It is equipped with an effective feature learning module that recovers local features lost in pooling via the DPI and adds global information to the point features through GCA. SimLOG is shown to outperform its competitors in most cases in two benchmark datasets. Both the feature learning module and DPI are detachable and can be incorporated into existing point-based networks to boost their performance.

In the future, we will explore the possibility of leveraging cross-modal data as an additional component to enhance the perception and feature learning of potential objects. For instance, building upon the successes of F-PointNet [34] and 3D-SIS [33], we can leverage the capabilities of a 2D detector to search for regions in images where objects might exist (i.e., 2D bounding boxes) along with corresponding image features, and then map them into the point cloud space. Through the elimination of a substantial number of background points outside these identified regions and the fusion of image features with point cloud features, the network can concentrate on areas in the point cloud where objects are likely to exist, facilitating the learning of richer multimodal features. This is anticipated to reduce false positives, refine the regression to more accurate 3D bounding boxes, and ultimately lead to more precise 3D detection results. Furthermore, we aim for it to function as a plug-and-play feature-enhancement module, seamlessly integrated into multimodal 3D trackers, such as EagerMOT [66], which focuses solely on bounding box interactions without considering image or point cloud features. This integration is intended to enhance the robustness of tracking 3D trajectories.

References

[1] C. Yi et al., "Hierarchical tunnel modeling from 3D raw LiDAR point cloud," *Computer-Aided Design*, vol. 114, pp. 143–154, Sep. 2019.

[2] J. Beltrán, C. Guindel, A. de la Escalera, and F. García, "Automatic extrinsic calibration method for LiDAR and camera sensor setups," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17677–17689, Oct. 2022.

[3] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 770–779.

[4] C. R. Qi, O. Litany, K. He, and L. Guibas, "Deep Hough voting for 3D object detection in point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9276–9285.

[5] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11037–11045.

[6] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "STD: Sparse-to-dense 3D object detector for point cloud," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1951–1960.

[7] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.

[8] D. Rukhovich, A. Vorontsova, and A. Konushin, "ImVoxelNet: Image to voxels projection for monocular and multi-view general-purpose 3D object detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 1265–1274.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12689–12697.

[10] G. Shi, R. Li, and C. Ma, "PillarNet: Real-time and high-performance pillar-based 3D object detection," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 35–52.

[11] Y. Wang et al., "Pillar-based object detection for autonomous driving," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 18–34.

[12] A. Paigwar, D. Sierra-Gonzalez, Ö. Erkent, and C. Laugier, "Frustum-PointPillars: A multi-stage approach for 3D object detection using RGB camera and LiDAR," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2926–2933.

[13] Z. Liu et al., "BEVFusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 2774–2781.

[14] T. Liang et al., "Bevfusion: A simple and robust LiDAR-camera fusion framework," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 10421–10434.

[15] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11784–11793.

[16] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[17] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 5099–5108.

[18] X. Ma, C. Qin, H. You, H. Ran, and Y. Fu, "Rethinking network design and local geometry in point cloud: A simple residual mlp framework," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–15.

[19] G. Qian et al., "Pointnext: Revisiting pointnet++ with improved training and scaling strategies," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 1–13.

[20] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3D object detection with pointformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 7459–7468.

[21] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst. (NIPS)*, 2017, pp. 5998–6008.

[22] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 667–675.

[23] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Computer Vision—ECCV 2020*. Springer, 2020, pp. 282–298, doi: 10.1007/978-3-030-58452-8.

[24] P. Sun et al., "Sparse R-CNN: End-to-end object detection with learnable proposals," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14449–14458.

[25] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2432–2443.

[26] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 567–576.

[27] S. Wang, S. Suo, W. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2589–2597.

[28] Y. Wang et al., "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 1145.

[29] S. Shi et al., "PV-RCNN: Point-voxel feature set abstraction for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10526–10535.

[30] Q. Xie et al., "MLCVNet: Multi-level context VoteNet for 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10444–10453.

[31] J. Noh, S. Lee, and B. Ham, "HVPR: Hybrid voxel-point representation for single-stage 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14600–14609.

[32] Y. Zheng, Y. Duan, J. Lu, J. Zhou, and Q. Tian, "HyperDet3D: Learning a scene-conditioned 3D object detector," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 5575–5584.

[33] J. Hou, A. Dai, and M. Nießner, "3D-SIS: 3D semantic instance segmentation of RGB-D scans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 4416–4425.

[34] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 918–927.

[35] J. Lahoud and B. Ghanem, "2D-driven 3D object detection in RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4632–4640.

[36] L. Yi, W. Zhao, H. Wang, M. Sung, and L. J. Guibas, "GSPN: Generative shape proposal network for 3D instance segmentation in point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3942–3951.

[37] S. Song and J. Xiao, "Deep sliding shapes for amodal 3D object detection in RGB-D images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 808–816.

[38] Z. Ren and E. B. Sudderth, "Three-dimensional object detection and layout prediction using clouds of oriented gradients," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1525–1533.

[39] Z. Wang, Q. Xie, M. Wei, K. Long, and J. Wang, "Multi-feature fusion VoteNet for 3D object detection," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 18, no. 1, pp. 1–17, Jan. 2022.

[40] J. Sun, H. Fang, X. Zhu, J. Li, and C. Lu, "Correlation field for boosting 3D object detection in structured scenes," in *Proc. AAAI Conference Artificial Intelligence*, 2022, pp. 2298–2306.

[41] Z. Liu, T. Huang, B. Li, X. Chen, X. Wang, and X. Bai, "EPNet++: Cascade bi-directional fusion for multi-modal 3D object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 7, pp. 1–18, Jun. 2022.

[42] J. Liang, P. An, and J. Ma, "Distribution aware votenet for 3D object detection," in *Proc. AAAI Conference Artificial Intelligence*, 2022, pp. 1583–1591.

[43] W. Wei et al., "Semantic consistency reasoning for 3-D object detection in point clouds," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 1, no. 1, pp. 1–14, May 2023.

[44] Z. Li, H. Yu, Z. Yang, T. Chen, and N. Akhtar, "AShapeFormer: Semantics-guided object-level active shape encoding for 3D object detection via transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 1012–1021.

[45] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3D object detection via transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2929–2938.

[46] X. Chen, H. Zhao, G. Zhou, and Y.-Q. Zhang, "PQ-transformer: Jointly parsing 3D objects and layouts from point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2519–2526, Apr. 2022.

[47] Y. Shen et al., "V-DETR: DETR with vertex relative position encoding for 3D object detection," 2023, *arXiv:2308.04409*.

[48] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "Oneformer3D: One transformer for unified point cloud segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 20943–20953.

[49] Y.-Q. Yang et al., "Swin3D: A pretrained transformer backbone for 3D indoor scene understanding," 2023, *arXiv:2304.06906*.

[50] Y. Wang, X. Chen, L. Cao, W. Huang, F. Sun, and Y. Wang, "Multimodal token fusion for vision transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12176–12185, doi: 10.1109/CVPR52688.2022.01187.

[51] H. Wang, "Cagroup3D: Class-aware grouping for 3S object detection on point clouds," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 29975–29988.

[52] D. Rukhovich, A. Vorontsova, and A. Konushin, "FCAF3D: Fully convolutional anchor-free 3D object detection," in *Proc. 17th Eur. Conf. Comput. Vis. (ECCV)*. Tel Aviv, Israel: Springer Oct. 2022, pp. 477–493.

[53] D. Rukhovich, A. Vorontsova, and A. Konushin, "TR3D: Towards real-time indoor 3D object detection," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2023, pp. 281–285.

[54] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 18920–18930.

[55] G. Fan, Z. Qi, W. Shi, and K. Ma, "Point-GCC: Universal self-supervised 3D scene pre-training via geometry-color contrast," 2023, *arXiv:2305.19623*.

[56] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3DNet: 3D object detection using hybrid geometric primitives," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 311–329.

[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[58] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2886–2897.

[59] M. Feng, S. Z. Gilani, Y. Wang, L. Zhang, and A. Mian, "Relation graph network for 3D object detection in point clouds," *IEEE Trans. Image Process.*, vol. 30, pp. 92–107, 2021.

[60] J. Chen, B. Lei, Q. Song, H. Ying, D. Z. Chen, and J. Wu, "A hierarchical graph network for 3D object detection on point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 389–398.

[61] M. Najibi et al., "DOPS: Learning to detect 3D objects and predict their 3D shapes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11910–11919.

[62] T. Tu et al., "ImGeoNet: Image-induced geometry-aware voxel representation for multi-view 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 6973–6984.

[63] Z. Chen, Q. Xu, and R. Cong, "Global context-aware progressive aggregation network for salient object detection," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10599–10606.

[64] B. Cheng, L. Sheng, S. Shi, M. Yang, and D. Xu, "Back-tracing representative points for voting-based 3D object detection in point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 8959–8968.

[65] Y. Wang, V. C. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3D object detection from multi-view images via 3D-to-2D queries," in *Proc. Conf. Robot Learn.*, vol. 164, 2022, pp. 180–191.

[66] A. Kim, A. Osep, and L. Leal-Taixé, "Eagermot: 3D multi-object tracking via sensor fusion," in *Proc. IEEE International Conference Robotics Automation*, Aug. 2021, pp. 11315–11321.