2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Hierarchical Point Attention for Indoor 3D Object Detection

Manli Shu[1,2,*] Le Xue[2] Ning Yu [2] Roberto Martín-Martín[2,3]
Caiming Xiong[2] Tom Goldstein[1] Juan Carlos Niebles[2,4] and Ran Xu[2]

***Abstract*— 3D object detection is an essential vision technique for various robotic systems, such as augmented reality and domestic robots. Transformers as versatile network architectures have recently seen great success in 3D point cloud object detection. However, the lack of hierarchy in a plain transformer restrains its ability to learn features at different scales. Such limitation makes transformer detectors perform worse on smaller objects and affects their reliability in indoor environments where small objects are the majority. This work proposes two novel attention operations as generic hierarchical designs for point-based transformer detectors. First, we propose Aggregated Multi-Scale Attention (MS-A) that builds multi-scale tokens from a single-scale input feature to enable more fine-grained feature learning. Second, we propose Size-Adaptive Local Attention (Local-A) with adaptive attention regions for localized feature aggregation within bounding box proposals. Both attention operations are model-agnostic network modules that can be plugged into existing point cloud transformers for end-to-end training. We evaluate our method on two widely used indoor detection benchmarks. By plugging our proposed modules into the state-of-the-art transformer-based 3D detectors, we improve the previous best results on both benchmarks, with more significant improvements on smaller objects.**

I. INTRODUCTION

3D computer vision models (*e.g.*, object detectors) help robotic and control systems perceive and understand the environment from 3D data (*e.g.*, point cloud), which provides more accurate geometric and spatial information and is robust to illumination and domain shifts. Since point clouds do not have a grid-like structure as images, previous works have proposed various neural network architectures for point cloud understanding [1]–[13]. With the success of attention-based architectures (*i.e.*, transformers) in other learning regime [14]–[16], it has recently been applied to point clouds [17]–[23]. Some properties of transformers make them ideal for modeling point clouds. For example, their permutation-invariant property is necessary for modeling unordered sets like point clouds, and their attention mechanism helps learn long-range relationships and capture global context.

Despite the advantages of transformers for point clouds, we find that the state-of-the-art transformer detectors have imbalanced performance across different object sizes, with lower average precision on smaller objects (see Section IV-B). Such imbalanced performance can affect the robustness of downstream applications, especially for indoor scenarios where the environments are cluttered with small objects [24], [25]. We speculate such bias against small objects can be attributed to two factors. First, for efficiency, existing models are trained on downsampled point clouds with far fewer points than the raw data. The extensive downsampling loses geometric details and impacts more significantly on smaller objects. Second, plain transformers [14], [15] only learn features at the global scale, whereas smaller objects may require more fine-grained feature extraction.

Motivated by the observations, we expect point cloud transformers to benefit from hierarchical feature learning strategies [26], [27], *e.g.*, multi-scale and localized feature learning. Nonetheless, considering the computation intensity of point cloud transformers, using higher-resolution (*i.e.*, higher point density) point cloud features can be inefficient. Furthermore, due to the irregularity of point clouds, it is non-trivial to integrate hierarchical designs into transformers for point-based 3D object detection.

Our approach. We propose two point-based attention modules for hierarchical feature learning on point clouds. Both modules are model-agnostic and can be plugged into any existing point-based transformers for end-to-end training.

We first propose *Aggregated Multi-Scale Attention* (MS-A), which builds higher resolution features from a single-scale input. It then uses the multi-scale features for cross-attention via multi-scale token aggregation [28] with little parameter overhead. The second proposed module is *Size-Adaptive Local Attention* (Local-A), where the attention regions are defined by the detector's box proposals for each object candidate. It thus allows adaptive local feature learning.

We evaluate our method on two widely used indoor 3D detection benchmarks, ScanNetV2 [24] and SUN RGB-D [25]. We plug our attention modules into existing transformer-based 3D detectors and perform end-to-end training. Our method improves the previous best results with little parameter overhead and with more significant improvements on smaller objects. We summarize our main contributions as follows:

- We identify that point-based 3D transformer detectors have performance imbalance issues and perform worse on smaller objects.
- Motivated by our observation, we propose two generic point attention operations that enable multi-scale and localized feature learning.
- We apply our method to various point-based transformer detectors and improve the previous best result on two

[1] Department of Computer Science, University of Maryland, College Park, MD, U.S.A. {manlis, tomg}@umd.edu
[2] Salesforce Research, Palo Alto, CA, U.S.A. {manli.shu, lxue, ning.yu, cxiong, ran.xu}@salesforce.com
[3] Department of Computer Science at the University of Texas at Austin, TX, U.S.A. robertomm@cs.utexas.edu
[4] Department of Computer Science, Stanford University, CA, U.S.A. jniebles@cs.stanford.edu
[*] Work done while at University of Maryland.

widely used indoor 3D detection benchmarks.

II. Related Works

Deep neural networks for 3D point cloud. Existing network architectures for point cloud learning can be roughly divided into two categories based on their point cloud representation: *grid-based* and *point-based*, yet in between, some hybrid architectures operate on both representations [29]–[33]. *Grid-based* methods project the irregular point clouds into grid-like structures, such as 3D voxels or pillars [34]–[38]. *Point-based* methods, on the other hand, directly learn features from the raw point cloud. Within this category, graph-based methods [39]–[42] use graphs to model the relationships among the points. Other works regard the point cloud as a set and learn features through set abstraction [4], [6], [43], [44]. Recent works explore the transformer architecture for point-based learning [12], [13], [17]–[19], [45] by feeding individual points as tokens into a transformer, where the attention mechanism learns point features at a global scale. While most previous methods improve point cloud learning by developing new backbones, our work aims to provide a generic model-agnostic solution. PAConv [46] proposes a generic convolution operation while we study cross-attention operations that can be integrated into the emerging transformer models.

Hierarchical designs for 3D vision transformers. Inspired by the literature in 2D vision [16], [28], [47], [48] especially convolutional neural networks [49] hierarchical designs for 3D transformers also seek to learn features at different granularities. Some methods [19], [50] propose attention mechanisms that can do multi-scale feature learning, but they only work on voxels and cannot be applied to point-based 3D models. Lai et al. [51] proposed stratified self-attention that can learn multi-scale features directly on point clouds, but it only supports downsampling, which limits more fine-grained feature learning. Our multi-scale attention, on the other hand, can produce features of arbitrary resolutions. Another line of work [12], [17], [18] proposes to learn localized features by applying self-attentions to local regions specified by k nearest neighbors or a fixed radius/window. Instead of a fixed region, our localized attention is size-adaptive, which uses adaptive local regions that match the approximated object sizes.

III. Method

A. Background

Point cloud object detection. Given a point cloud $\mathcal{P}_{\text{raw}}$ with a set of P points $\mathcal{P} = \{\mathbf{p_i}\}_{i=1}^{P}$, each point $p_i \in \mathbb{R}^3$ is represented by its 3-dimensional coordinate. 3D object detection on point cloud aims to predict a set of bounding boxes for the objects in the scene, including their locations (as the center of the bounding box), size and orientation of the bounding box, and the semantic class of the corresponding object. Note that due to the computation limit, the point cloud is downsampled at the early stage of a model to a subset of $\mathcal{P}_{\text{raw}}$, which contains N ($N << P$) points. $\mathcal{P} = \text{SA}(\mathcal{P}_{\text{raw}}) = \{\mathbf{p_i}\}_{i=1}^{N}$ contains the aggregated groups of points around N group centers, where SA (set abstraction) is the aggregation function, and the group centers are sampled from the raw point cloud using *Farthest Point Sample (FPS)* [4], a random sampling algorithm that provides good coverage of the entire point cloud.

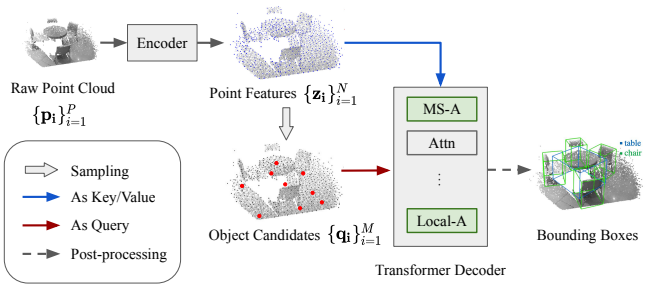


Fig. 1. **A point-based 3D transformer detectors with our proposed modules (MS-A and Local-A).** Detector overview: the raw point cloud is downsampled during encoding to obtain the point features, which serve as the key and value in the transformed decoder. Object candidates are sampled from the point features (e.g., using FPS). The transformer decoder learns object features via alternating self- and cross-attentions. The proposed MS-A and Local-A are cross-attention modules that can be plugged into the transformer. See Fig. 2 and Fig. 3 for the design details of each module.

Point-based 3D transformer detectors. Our method is built on point-based 3D object detectors [7], [13], [52], which detect 3D objects in point clouds in a bottom-up manner. Compared to other 3D detectors that generate box proposals in a top-down manner on the bird's-eye view or voxelized point clouds [53], [54], point-based methods work directly on the irregular point cloud without quantization errors.

We illustrate the general point-based 3D transformer detector in Fig. 1. The features of the input point cloud $\{\mathbf{z_i}\}_{i=1}^{N}, \mathbf{z_i} \in \mathbb{R}^d$ is obtained using a backbone model (*e.g.*, PointNet++ [6]), where d is the feature dimension. Point-based detectors generate bounding box predictions starting with M ($M < N$) initial object *candidates* $\{\mathbf{q_i}\}_{i=1}^{M}, \mathbf{q_i} \in \mathbb{R}^C$, sampled from the point cloud as object centers. A common candidate sampling approach is the Farthest Point Sample (FPS). Once the initial candidates are obtained, the detector extracts features for every object candidate. Attention-based methods [13] learn features by doing self-attention among the object candidates and cross-attention between the candidates (*i.e.*, query) and point features $\{\mathbf{z_i}\}_{i=1}^{N}$.

The learned features of the object candidates will then be passed to prediction heads, which predict the attributes of the bounding box for each object candidate. The attributes of a 3D bounding box include its location (box center) $\hat{\mathbf{c}} \in \mathbb{R}^3$, size $\hat{\mathbf{d}} \in \mathbb{R}^3$, orientation (heading angles) $\hat{\mathbf{a}} \in \mathbb{R}$, and the semantic label of the object $\hat{\mathbf{s}}$. With these parameterizations, we can represent a bounding box proposal as $\hat{\mathbf{b}} = \{\hat{\mathbf{c}}, \hat{\mathbf{d}}, \hat{\mathbf{a}}, \hat{\mathbf{s}}\}$.

Attention mechanism is the basic building block of transformers. The attention function takes in query (Q), key (K), and value (V) as the input. The output of the attention function is a weighted sum of the value, with the attention weight being the scaled dot-product between the key and query:

$$\text{Attn}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_h}})V, \tag{1}$$

where d_h is the hidden dimension of the attention layer. For self-attention, $Q \in \mathbb{R}^{d_h}$, $K \in \mathbb{R}^{d_h}$ and $V \in \mathbb{R}^{d_v}$ are transformed from the input $X \in \mathbb{R}^d$ via linear projection with parameter matrix $W_i^Q \in \mathbb{R}^{d \times d_h}$, $W_i^K \in \mathbb{R}^{d \times d_h}$, and $W_i^V \in \mathbb{R}^{d \times d_v}$ respectively. For cross-attention, Q, K, and V can have different sources.

In practice, transformers adopt the **multi-head attention** design, where multiple attention functions are applied in parallel across different attention *head*s. The input of each attention head is a segment of the layer's input. Specifically, the query, key, and value are split along the hidden dimension into $(Q_i, K_i, V_i)_{i=1}^h$, with $Q_i \in \mathbb{R}^{d_h/h}, K_i \in \mathbb{R}^{d_h/h}, V_i \in \mathbb{R}^{d_v/h}$, where h is the number of attention heads. The final output of the multi-head attention layer is the projection of the concatenated outputs of all attention heads:

$$\begin{aligned}\text{MultiHead}(Q,K,V) = \text{Concat}(\{\text{Attn}(Q_0, K_0, V_0);\\ ...; \text{Attn}(Q_{h-1}, K_{h-1}, V_{h-1})\})W^O,\end{aligned} \tag{2}$$

where the first term denotes the concatenation of the output and W^O is the output projection matrix.

B. Aggregated Multi-Scale Attention

In existing point-based transformer detectors, the cross-attention modules are applied between object candidates and all other points of the point cloud. However, due to the computation overhead of the attention function, the actual point cloud that the model uses is a downsampled set of 1024 points [12], [13], whereas the raw point cloud usually contains tens of thousands points [24], [25]. Such extensive downsampling causes a loss of detailed geometric information and fine-grained features essential for dense prediction tasks like object detection.

To this end, we propose *Aggregated Multi-Scale Attention* (MS-A), which learns to build arbitrary higher-resolution (*i.e.*, higher point density) feature maps from the single-scale feature input. It then uses features of both features as the multi-scale key/value in the cross-attention between object candidates and other points. The multi-scale feature aggregation is realized via multi-head token aggregation, where we use the key and value of different scales in different subsets of attention heads. We aim to provide fine-grained geometric details for object-level feature learning.

The first step of MS-A is to build a higher-resolution feature map from the single-scale input. We propose a learnable upsampling procedure. Given the layer's input point cloud feature $\{\mathbf{z_i}\}_{i=1}^N, \mathbf{z_i} \in \mathbb{R}^d$, MS-A can create a feature map with an arbitrary number (N') of points. To get the locations (*i.e.*, coordinates) of the N' points, we use FPS to sample N' points from the *raw point cloud* and obtain $\{\mathbf{p_i}\}_{i=1}^{N'}, \mathbf{p_i} \in \mathbb{R}^3$. Next, for each sampled point $\mathbf{p_i}$ in the N'-point cloud, we initialize their feature via three-nearest-neighbor weighted interpolation [6], a generic operation adopted by many point-based methods [17], [46], [52]. Unlike previous work, which uses the interpolation for the skip connection between the encoder and decoder, our N' points are sampled arbitrarily from the raw point cloud. We can sample a N'-point feature to have more points than any feature map in the model, thus providing more fine-grained geometry details.

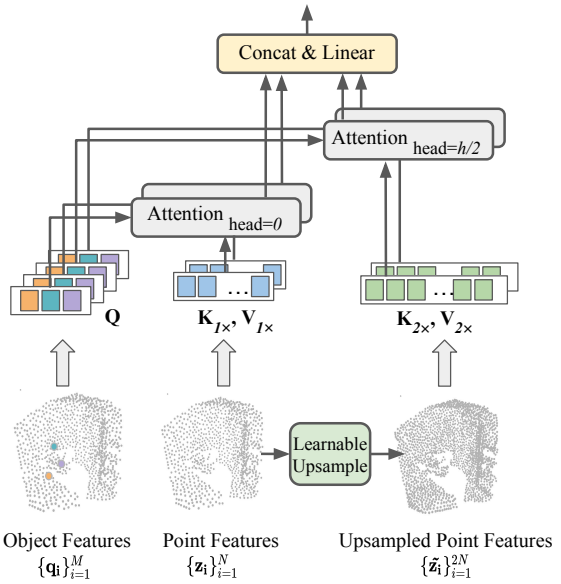


Fig. 2. **Aggregated Multi-Scale Attention (MS-A)** learns features at different scales within the multi-head cross-attention design. It constructs higher resolution (*i.e.*, higher point density) point features from the single-scale input point features and uses keys and values of both scales.

From the interpolated feature initialization, we obtain the final feature representation of the N'-point feature map via a learnable module Φ_θ. We use MLP as the learnable projection function. The learned N'-point feature map is:

$$\{\tilde{\mathbf{z}}_\mathbf{i}\}_{i=1}^{N'},\ \tilde{\mathbf{z}}_\mathbf{i} = \Phi_\theta(\text{interpolate}(\{z_i^0, z_i^1, z_i^2\})) \tag{3}$$

After the upsampling, we have two sets of point features of different scale $\{\mathbf{z_i}\}_{i=1}^N, \{\tilde{\mathbf{z}}_\mathbf{i}\}_{i=1}^{N'}$. To avoid computation increase, we perform multi-head cross-attention on both sets of point features in a single pass. We use features of different scales on different attention heads. We divide attention heads evenly into two groups and use $\{\mathbf{z_i}\}_{i=1}^N$ to derive K and V in the first group while using $\{\mathbf{z_i}\}_{i=1}^{N'}$ for the other. Both groups share the same set of queries derived from object candidates $\{\mathbf{q_i}\}_{i=1}^M$. Since the input and output of this module are the same as a plain attention module, we can plug MS-A into any attention-based model to enable feature learning at different scales. In practice, we set $N' = 2N$ and apply MS-A only at the first layer of a transformer decoder to introduce minimal computation overhead. Further analyses on design choices are in Section IV-D.

C. Size-Adaptive Local Attention

Although the attention mechanism can effectively model the long-range relationship between points, it cannot guarantee that the learned module will pay more attention to points that are important to a particular object (*e.g.*, those belonging to the object). On the other hand, the lack of hierarchy in plain transformers does not support explicit localized feature extraction. While some work addresses this issue by restricting

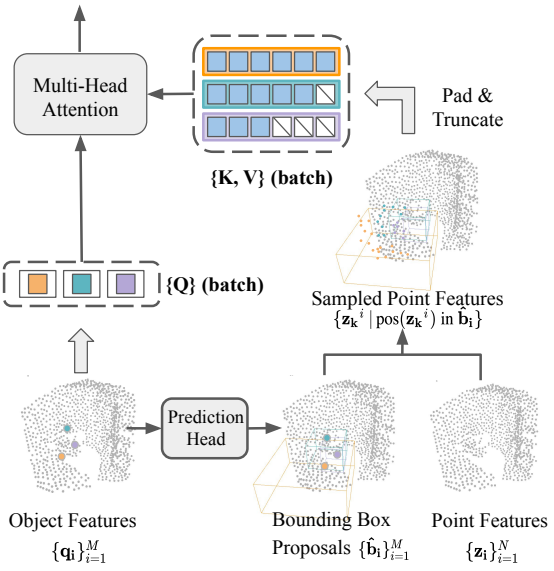


Fig. 3. **Size-Adaptive Local Attention (Local-A)** performs adaptive local attention between each object candidate (query) and the points inside its corresponding bounding box proposal. The attention range (the token lengths of keys and values) varies across different object candidates, so perform padding/truncating to allow batch processing.

attention to fixed local regions [12], [19], we propose *Size-Adaptive Local Attention* (Local-A) that defines local regions adaptively based on the size of each bounding box proposal.

Point-based 3D detectors can generate intermediate bounding box proposals $\{\hat{\mathbf{b}}_\mathbf{i}\}_{i=1}^M$ from the object features ($\{\mathbf{q_i}\}_{i=1}^M$) of a decoder layer. Given the bounding boxes generated for each object candidate $\mathbf{q_i}$, we perform cross-attention between $\mathbf{q_i}$ and points sampled from within its corresponding box $\hat{\mathbf{b}}_\mathbf{i}$. We thus have customized size-adaptive attention regions for every query point. For every input object candidate $\mathbf{q_i}^l \in \mathbb{R}^d$ at decoder layer l, it is updated by Local-A via (i denotes the index of a query point):

$$\mathbf{q_i}^{l+1} = \text{Attn}(Q_i^l, K_i, V_i), \text{where} \tag{4}$$

$$Q_i^l = \mathbf{q_i}^l W^Q, K_i = Z_i W^K, V_i = Z_i W^V \text{ with} \tag{5}$$

$$Z_i = \{\mathbf{z_j}^i \mid \texttt{pos}(\mathbf{z_j}^i) \text{ in } \hat{\mathbf{b}}_\mathbf{i}\}, \hat{\mathbf{b}}_\mathbf{i} = \text{Pred}_{box}^l(\mathbf{q_i}^l). \tag{6}$$

In Eq(6), $\texttt{pos}(\cdot)$ obtains the 3D-coordinate (x, y, z) of a point feature $\mathbf{z}$, and $\hat{\mathbf{b}}_\mathbf{i}$ is the model's bounding box prediction for query point $\mathbf{q_i}^l$ (generated via the prediction head Pred_{box}^l). Z_i hereby denotes a set of points *inside* box $\hat{\mathbf{b}}_\mathbf{i}$. K_i and V_i are derived from this set of point features. Note that $\{\mathbf{z_i}\}_{i=1}^N$ is the entire set of point features extracted by the encoder and is not updated during decoding.

Since each $\mathbf{q_i}$ has its own sets of keys and values depending on the size of its bounding box predictions, the size of K_i/V_i differs across queries. To allow batch computation, we set a maximum number of points (N_{local}) for the sampling process and use N_{local} as a fixed token length for every query point. For bounding boxes that contain less than N_{local} points, we pad the point sequence with an unused token to N_{local} and mask the unused tokens in the cross-attention function; for those containing more than N_{local} points, we randomly discard them and truncate the sequence to have N_{local} points as keys and values. Lastly, when the bounding box is empty, we perform ball query [4] around the object candidate to sample N_{local} points.

Like MS-A, Local-A does not pose additional requirements on a module's input. Therefore, we can apply it to any transformer layer. We apply Local-A at the end of a transformer where bounding box proposals are generally more accurate. We empirically set $N_{local} = 16$ with related ablation study in Section IV-D.

IV. Experiments

In this section, we first evaluate our method on two widely used indoor point cloud detection datasets, ScanNetV2 and SUN RGB-D. Next, we provide qualitative analyses of visualizations of bounding boxes and attention weights. We also conduct evaluations with our proposed size-aware metrics. Lastly, we include ablation studies on the design choices of our method.

Datasets. *ScanNetV2* [24] consists of 1513 reconstructed meshes of hundreds of indoor scenes, with rich annotations for 3D scene understanding tasks, including classification, semantic segmentation, and object detection. For object detection, it provides axis-aligned bounding boxes with 18 object categories. We follow the official dataset split by using 1201 samples for training and 312 samples for testing. *SUN RGB-D* [25] is a single-view RGB-D dataset with 10335 samples. For 3D object detection, it provides oriented bounding box annotations with 37 object categories, while we follow the standard evaluation protocol [52] and only use the ten common categories. The training and testing split contains 5285 and 5050 samples, respectively.

Evaluation metrics. For both datasets, we follow the standard evaluation protocol [52] and use the mean Average Precision (mAP) as the evaluation metric. We report mAP scores under two different Intersection over Union (IoU) thresholds: mAP@0.25 and mAP@0.5. In addition, in Section IV-B, to evaluate model performance across different object sizes, we follow the practice in 2D vision [55] and implement our size-aware metrics that measure the mAP on small, medium, and large objects, respectively. Because of the randomness of point cloud training and inference, we train a model 5 times and test each model 5 times. We report both the best and the average results among the 25 trials.

Baselines. We validate our method and demonstrate its model-agnostic property by applying it to three different transformer-based point cloud detectors: Group-Free [13], RepSurf [56], and 3DETR [12]. Group-Free encodes point features with a PointNet++ [6] backbone and learns object features using a transformer decoder with plain attention. We consider two Group-Free configurations: Group-Free^{6,256} and Group-Free^{12,512}, where Group-Free^{L,O} denotes the variant with L decoder layers and O object candidates. RepSurf-U learns point features from a novel multi-surface representation explicitly describing local geometry. It uses a similar transformer decoder as [13] and has two configurations.

The official implementation and the averaged results of RepSurf-U for object detection are not publicly available, so we include the results of our reproduction of RepSurf-U. 3DETR is another end-to-end 3D transformer detector. Unlike other baselines, 3DETR's encoder and decoder are both transformers. The official result of 3DETR is not over multiple runs, so we report our reproduced results for this baseline.

We also include previous point-based 3D detectors for comparison. VoteNet [52] aggregates features for object candidates through end-to-end optimizable Hough Voting. H3DNet [57] proposes a hybrid set of geometric primitives for object detection and trains multiple individual backbones for each primitive. Pointformer [18] proposes a hierarchical architecture as the backbone and adopts the voting algorithm of VoteNet for object detection.

Implementation details. For a baseline model with L transformer layers, we enable multi-scale feature learning by replacing the cross-attention of the 1-st layer with MS-A. At the L-th layer (*i.e.*, the last decoder layer), we replace its cross-attention with Local-A. We follow the original training settings of the baseline models [12], [13], [56].

TABLE I

PERFORMANCE OF OBJECT DETECTION ON SCANNETV2.

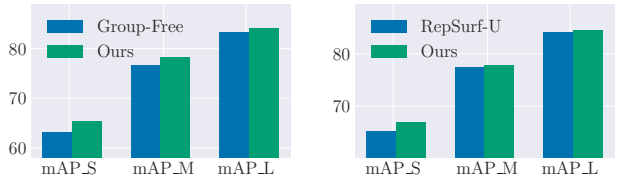
We follow the standard protocol [52] by reporting the best results over 25 trials (5 trainings, each with 5 testings) with the averaged results in the bracket. Backbone stands for the encoder model architecture, and PN++ denotes PointNet++. Group-Free^{L,O} denotes the variant with L decoder layers and O object candidates. The same notation applies to RepSurf-U. Note that the detection code of RepSurf is not published, so we implement our version of RepSurf-U and report the reproduced results (repd.).

Methods	#Params	Backbone	ScanNet V2	
			mAP@0.25	mAP@0.50
VoteNet [52]	-	PN++	62.9	39.9
H3DNet [57]	-	PN++	64.4	43.4
H3DNet [57]	-	4×PN++	67.2	48.1
Pointformer [18]	-	transformer	64.1	42.6
3DETR-m [12] (repd.)	7.4M	transformer	64.1 (62.9)	45.8 (44.1)
w/ MS + Local (Ours)	7.5M	transformer	**64.8 (63.7)**	**46.4 (45.2)**
Group-Free^{6,256} [13]	14.5M	PN++	67.3 (66.3)	48.9 (48.5)
w/ MS + Local (Ours)	14.6M	PN++	67.8 (66.8)	50.8 (49.5)
RepSurf-U^{6,256} [56]	14.5M	PN++	68.8 (-)	50.5 (-)
RepSurf-U^{6,256} (repd.)	14.5M	PN++	68.0 (67.4)	50.2 (48.7)
w/ MS + Local (Ours)	14.6M	PN++	**69.9 (68.7)**	**53.0 (51.3)**
Group-Free^{12,512} [13]	29.6M	PN++w2x	69.1 (68.6)	52.8 (51.8)
w/ MS + Local (Ours)	29.6M	PN++w2x	70.3 (69.0)	53.5 (52.3)
RepSurf-U^{12,512} [56]	29.7M	PN++w2x	71.2 (-)	54.8 (-)
RepSurf-U^{12,512} (repd.)	29.7M	PN++w2x	70.8 (70.2)	54.4 (53.6)
w/ MS + Local (Ours)	29.8M	PN++w2x	**71.4 (70.6)**	**55.6 (54.3)**

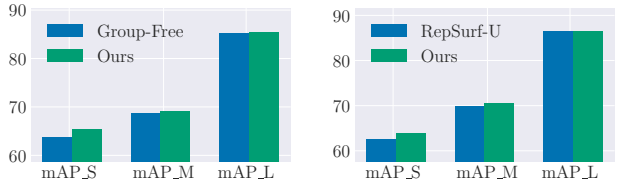
TABLE II

PERFORMANCE OF OBJECT DETECTION ON SUN RGB-D.

Methods	mAP@0.25	mAP@0.50
VoteNet [52]	59.1	35.8
H3DNet [57]	60.1	39.0
3DETR-m [12]	59.1	32.7
Pointformer [18]	61.1	36.6
Group-Free^{6,256} [13]	63.0 (62.6)	45.2 (44.4)
w/ MS + Local (Ours)	63.3 (62.8)	45.8 (44.9)
RepSurf-U^{6,256} [56]	64.3 (-)	45.9 (-)
RepSurf-U^{6,256} (repd.)	64.0 (63.3)	45.7 (45.2)
w/ MS + Local (Ours)	**64.2 (63.6)**	**47.0 (45.7)**



(a) Per size-category (S/M/L) mAPs on ScanNetV2.



(b) Per size-category (S/M/L) mAPs on SUN-RGBD.

Fig. 4. **Performance on object of different sizes**. We define the S/M/L thresholds based on each dataset's statistics (volume distribution).

A. Main Results

In Table I, we observe consistent improvements in baseline models when equipped with our attention modules. On ScanNetV2, we perform on par with the state-of-the-art RepSurf-U detector by applying MS-A and Local-A to Group-Free, and we can further improve RepSurf-U across varying model configurations with little parameter overhead. Table II shows a similar trend on SUN RGB-D. Our method has larger improvements on ScanNetV2 than SUN RGB-D, which can be attributed to the different complexity of the two datasets: a ScanNetV2 sample contains the entire scan of a scene, whereas a SUN RGB-D sample only contains a single view. The benefit of our hierarchical attention is more prominent when handling complex scenes with fine-grained raw data.

B. Performance on objects of different sizes.

In addition to the standard evaluation metrics, we also examine the models' performance across different object sizes. We define the size-aware metric for 3D detection following the tradition in 2D detection [55]. We set the threshold for mAP_S as the $30th$ percentile of the volume of all objects and use the $70th$ percentile as the threshold for mAP_L.

We evaluate Group-Free and RepSurf models on both benchmark datasets. From the results in Fig. 4, we can see that mAPs on smaller objects (mAP_S) are much lower than on larger objects for both models. When applied with our proposed attention modules, we observe the largest performance gains in mAP_S .

C. Qualitative Results

In Figure 5, we provide qualitative results on both datasets. The visualized results are of our methods applied to the Group-Free detectors. The qualitative results suggest that our model can detect and classify objects of different scales even in complex scenarios containing more than ten objects (*e.g.*, the example in the bottom row). By looking into cross-attention weights in the transformer detector, we find that

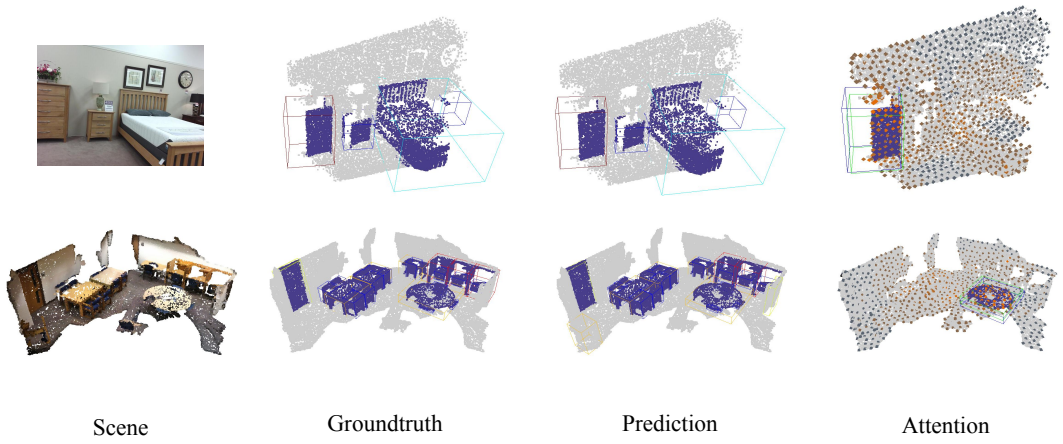


Fig. 5. **Qualitative results on SUN RGB-D (top) and ScanNetV2 (bottom)**. The color of a bounding box in the middle two columns stands for the semantic label of the object. In the last column, we draw both the ground truth (in green) and the prediction (in blue) of the object. We highlight the points that belong to an object for better visualization. In the last column, we visualize the attention weight of the last transformer layer (before applying Local-A). We visualize the cross-attention weight between an object candidate and the point cloud.

object candidates tend to have higher correlations with points that belong to their corresponding objects.

D. Ablation Study

The maximum number of points (N_{local}) in Local-A. In Local-A, for each object candidate (i.e., query), we sample a set of points within its corresponding bounding box proposal and use the point features as the key and value for this object candidate in the cross-attention function. As introduced in Section III-C, we cap the number of sampled points with N_{local} to allow batch computation.

TABLE III

THE EFFECT OF N_{local} IN LOCAL-A.

When there are enough points, a larger N_local means the points are sampled more densely within each bounding box proposal.

N_{local}	mAP@0.25	mAP@0.50	mAP_S	mAP_M	mAP_L
8	67.8	51.1	64.3	77.2	82.8
16	**68.8**	**52.3**	65.1	**77.9**	83.4
24	68.7	**52.3**	**65.2**	77.7	83.5
32	68.3	52.1	64.7	77.3	**83.8**

From Table III, we find that too little number of points (e.g., $N_{local} = 8$) for Local-A results in a performance drop. As N_{local} increases, we do not observe a significant performance gain when it exceeds $N_{local} = 16$. Intuitively, a small N_{local} means the points within each bounding box are sampled sparsely, which can be too sparse to provide enough information about any object. This explains why $N_{local} = 8$ does not work well. On the other hand, a large N_{local} may only benefit large objects and have little effect on smaller objects because the latter are padded with unused tokens.

MS-A with different feature resolutions. Similar to how the learnable upsampling in MS-A produces higher-resolution features, we can implement learnable *downsampling* using conventional set abstraction [4], which aggregates point features within local groups and produce feature maps with fewer points (*i.e.*, lower resolution). Intuitively, a higher-resolution feature map provides more fine-grained geometry details, while a more coarse one may provide a more global context. We conduct an empirical analysis on MS-A with different sampling ratios to study the effects of feature maps of different granularity. We choose a sampling ratio of 0.5 and 2.0 to represent coarse and fine-grained features, respectively.

TABLE IV

MS-A WITH DIFFERENT FEATURE SCALES.

Feature scale $= s$ means the feature map contains $s \times N$ points. A larger s denotes a feature map with higher point density (*i.e.*, resolution)

Scales s	mAP@0.25	mAP@0.50	mAP_S	mAP_M	mAP_L
[1]	68.6	51.8	63.1	76.6	83.2
[1, 2]	**68.9**	**52.5**	**65.0**	**77.5**	**83.9**
[0.5, 1, 2]	67.9	51.7	64.6	76.7	**83.9**

Results in Table IV suggest that coarse features ($s = 0.5$) do not benefit transformer detectors. This is expected because transformers do not have limited receptive fields and thus do not need coarse-grained feature maps to learn global context.

V. CONCLUSION

This work presents Aggregated Multi-Scale Attention (MS-A) and Size-Adaptive Local Attention (Local-A), two generic point-based attention operations that can be applied to various 3D transformer detectors and enable fine-grained feature learning. We improve point-based transformer detectors on two challenging indoor 3D detection benchmarks, with the largest improvement margin on smaller objects. As our method promotes fine-grained 3D feature learning, which is important to many 3D vision systems, future work will adapt the proposed attention modules to other applications, such as segmentation. Another future direction is to apply our method to improve outdoor detectors for applications like autonomous vehicles. Considering that mainstream outdoor detectors are usually not point-based, one may need to adapt the attention operations to other 3D representations.

REFERENCES

[1] B. Graham, M. Engelcke, and L. van der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *CVPR*, 2018.

[2] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *CVPR*, 2018.

[3] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from RGB-D data," in *CVPR*, 2018.

[4] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *CVPR*, 2017.

[5] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas, "Volumetric and multi-view cnns for object classification on 3d data," in *CVPR*, 2016.

[6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *NIPS*, 2017.

[7] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *CVPR*, 2019.

[8] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M. Yang, and J. Kautz, "Splatnet: Sparse lattice networks for point cloud processing," in *CVPR*, 2018.

[9] Y. Xu, T. Fan, M. Xu, L. Zeng, and Y. Qiao, "Spidercnn: Deep learning on point sets with parameterized convolutional filters," in *ECCV*, 2018.

[10] Y. Yang, C. Feng, Y. Shen, and D. Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," in *CVPR*, 2018.

[11] T. Guan, J. Wang, S. Lan, R. Chandra, Z. Wu, L. Davis, and D. Manocha, "M3DETR: multi-representation, multi-scale, mutual-relation 3d object detection with transformers," in *WACV*, 2022.

[12] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in *ICCV*, 2021.

[13] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-free 3d object detection via transformers," in *ICCV*, 2021.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NLPS*, 2017.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR*, 2021.

[16] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in *ICCV*, 2021.

[17] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, "Point transformer," in *ICCV*, 2021.

[18] X. Pan, Z. Xia, S. Song, L. E. Li, and G. Huang, "3d object detection with pointformer," in *CVPR*, 2021.

[19] C. Zhang, H. Wan, X. Shen, and Z. Wu, "Patchformer: An efficient point transformer with patch attention," in *CVPR*, 2022.

[20] X. Yu, L. Tang, Y. Rao, T. Huang, J. Zhou, and J. Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," in *CVPR*, 2022.

[21] J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, "Modeling point clouds with self-attention and gumbel subset sampling," in *CVPR*, 2019.

[22] Z. Liu, Z. Chen, S. Xie, and W. Zheng, "Transgrasp: A multi-scale hierarchical point transformer for 7-dof grasp detection," in *ICRA*. IEEE, 2022, pp. 1533–1539.

[23] J. Kini, A. Mian, and M. Shah, "3dmodt: Attention-guided affinities for joint detection & tracking in 3d point clouds," in *ICRA*. IEEE, 2023, pp. 841–848.

[24] A. Dai, A. X. Chang, M. Savva, M. Halber, T. A. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *CVPR*, 2017.

[25] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *CVPR*, 2015.

[26] S. Feng, Z. Zhou, J. S. Smith, M. Asselmeier, Y. Zhao, and P. A. Vela, "GPF-BG: A hierarchical vision-based planning framework for safe quadrupedal navigation," in *ICRA*. IEEE, 2023, pp. 1968–1975.

[27] C. Wen, H. Huang, Y. Liu, and Y. Fang, "Pyramid learnable tokens for 3d lidar place recognition," in *ICRA*. IEEE, 2023, pp. 4143–4149.

[28] S. Ren, D. Zhou, S. He, J. Feng, and X. Wang, "Shunted self-attention via multi-scale token aggregation," in *CVPR*, 2022.

[29] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *CVPR*, 2018.

[30] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *CVPR*, 2019.

[31] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, "PV-RCNN: point-voxel feature set abstraction for 3d object detection," in *CVPR*, 2020.

[32] M. Ye, S. Xu, and T. Cao, "Hvnet: Hybrid voxel network for lidar based 3d object detection," in *CVPR*, 2020.

[33] H. Wang, L. Ding, S. Dong, S. Shi, A. Li, J. Li, Z. Li, and L. Wang, "Cagroup3d: Class-aware grouping for 3d object detection on point clouds," in *NeurIPS*, 2022.

[34] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[35] G. Riegler, A. O. Ulusoy, and A. Geiger, "Octnet: Learning deep 3d representations at high resolutions," in *CVPR*, 2017.

[36] P. Wang, Y. Liu, Y. Guo, C. Sun, and X. Tong, "O-CNN: octree-based convolutional neural networks for 3d shape analysis," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 72:1–72:11, 2017.

[37] C. He, R. Li, S. Li, and L. Zhang, "Voxel set transformer: A set-to-set approach to 3d object detection from point clouds," in *CVPR*, 2022.

[38] X. Wang, J. Lei, H. Lan, A. Al-Jawari, and X. Wei, "Dueqnet: Dual-equivariance network in outdoor 3d object detection for autonomous driving," in *ICRA*. IEEE, 2023, pp. 6951–6957.

[39] M. Simonovsky and N. Komodakis, "Dynamic edge-conditioned filters in convolutional neural networks on graphs," in *CVPR*, 2017.

[40] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 146:1–146:12, 2019.

[41] Q. Xu, X. Sun, C. Wu, P. Wang, and U. Neumann, "Grid-gcn for fast and scalable point cloud learning," in *CVPR*, 2020.

[42] H. Zhou, Y. Feng, M. Fang, M. Wei, J. Qin, and T. Lu, "Adaptive graph convolution for point cloud analysis," in *ICCV*, 2021.

[43] 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 *ICLR*, 2022.

[44] H. Wang, S. Shi, Z. Yang, R. Fang, Q. Qian, H. Li, B. Schiele, and L. Wang, "Rbgnet: Ray-based grouping for 3d object detection," in *CVPR*, 2022.

[45] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer V2: grouped vector attention and partition-based pooling," in *NeurIPS*, 2022.

[46] M. Xu, R. Ding, H. Zhao, and X. Qi, "Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds," in *CVPR*, 2021.

[47] Y. Li, C. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *CVPR*, 2022.

[48] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable DETR: deformable transformers for end-to-end object detection," in *ICLR*, 2021.

[49] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, 1989.

[50] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, "Deformable feature aggregation for dynamic multi-modal 3d object detection," in *ECCV*, 2022.

[51] X. Lai, J. Liu, L. Jiang, L. Wang, H. Zhao, S. Liu, X. Qi, and J. Jia, "Stratified transformer for 3d point clouds segmentation," in *CVPR*, 2022.

[52] C. R. Qi, O. Litany, K. He, and L. J. Guibas, "Deep hough voting for 3d object detection in point clouds," in *ICCV*, 2019.

[53] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in *CVPR*, 2017.

[54] K. Tian, Y. Ye, Z. Zhu, P. Li, and G. Huang, "Efficient and hybrid decoder for local map construction in bird'-eye-view," in *ICRA*. IEEE, 2023, pp. 8378–8385.

[55] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *ECCV*, 2014.

[56] H. Ran, J. Liu, and C. Wang, "Surface representation for point clouds," in *CVPR*, 2022.

[57] Z. Zhang, B. Sun, H. Yang, and Q. Huang, "H3dnet: 3d object detection using hybrid geometric primitives," in *ECCV*, 2020.