OcTr: Octree-based Transformer for 3D Object Detection

Chao Zhou\textsuperscript{1,2}, Yanan Zhang\textsuperscript{1,2}, Jiaxin Chen\textsuperscript{2}, Di Huang\textsuperscript{1,2,3}\textsuperscript{*}

\textsuperscript{1}State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
\textsuperscript{2}School of Computer Science and Engineering, Beihang University, Beijing, China
\textsuperscript{3}Hangzhou Innovation Institute, Beihang University, Hangzhou, China

\{zhouchaobeing, zhangyanan, jiaxinchen, dhuang\}@buaa.edu.cn

Abstract

A key challenge for LiDAR-based 3D object detection is to capture sufficient features from large scale 3D scenes especially for distant or/and occluded objects. Albeit recent efforts made by Transformers with the long sequence modeling capability, they fail to properly balance the accuracy and efficiency, suffering from inadequate receptive fields or coarse-grained holistic correlations. In this paper, we propose an Octree-based Transformer, named OcTr, to address this issue. It first constructs a dynamic octree on the hierarchical feature pyramid through conducting self-attention on the top level and then recursively propagates to the level below restricted by the octants, which captures rich global context in a coarse-to-fine manner while maintaining the computational complexity under control. Furthermore, for enhanced foreground perception, we propose a hybrid positional embedding, composed of the semantic-aware positional embedding and attention mask, to fully exploit semantic and geometry clues. Extensive experiments are conducted on the Waymo Open Dataset and KITTI Dataset, and OcTr reaches newly state-of-the-art results.

1. Introduction

3D object detection from point clouds has received extensive attention during the past decade for its ability to provide accurate and stable recognition and localization in autonomous driving perception systems. In this task, feature learning plays a very fundamental and crucial role; yet it is rather challenging due to not only the disordered and sparse nature of data sampling, but also to insufficient acquisition under occlusion or at a distance. To address this issue, many methods have been proposed, which can be taxonomized into two major classes, \textit{i.e.} grid-based and point-based. The former first regularize point clouds into multi-view images or voxels and then apply 2D or 3D CNNs to build shape representations \cite{4, 52}, while the latter directly conduct MLP based networks such as PointNet++ \cite{33} and DGCNN \cite{50} on original points for geometry description \cite{32, 40, 42, 60}. Unfortunately, they fail to capture necessary context information through the small receptive fields in the deep models, leading to limited results.

Witnessing the recent success of Transformers in NLP, many studies have investigated and extended such architectures for 3D vision \cite{24, 29, 58, 61}. Transformers are reputed to model long-range dependencies, delivering global receptive fields, and to be suitable for scattered inputs of arbitrary sizes. Meanwhile, in contrast to those static weights that are learned in convolutions, Transformers dynamically aggregate the input features according to the relationships
between tokens. Regarding the case in 3D object detection, compared to point-based Transformers [11,61], voxel-based ones show the superiority in efficiency. However, they tend to suffer heavy computations when dealing with large scale scenes because of the quadratic complexity of Transformers, with the underlying dilemma between the grid size and the grid amount in voxelization. Taking the KITTI dataset as an example, it is unrealistic for Transformers to operate on the feature map with the spatial shape of $200 \times 176 \times 5$, which is commonly adopted in most of the detection heads [38,46,52,55].

More recently, there have appeared an influx of efficient self-attention model variants that attempt to tackle long sequences as input. They generally sparsify the attention matrix by fixed patterns [7,23,34], learned patterns [21,45] or a combination of them [1,56]. Fixed patterns chunk the input sequence into blocks of local windows or dilation windows, whilst learned patterns determine a notion of token relevance and eliminate or cluster outliers. Specific to 3D object detection from point clouds, VoTr [24] modifies self-attention with pre-defined patterns including local windows and stride dilation ones in a sparse query manner, and the dilation mechanism enlarges the receptive field by sampling attending tokens in a radius. SST [9] splits input tokens into non-overlapping patterns in a block-wise way and enables window shifting to capture cross-window correlation. Despite some improvements reported, they both only achieve bigger local receptive fields rather than the expected global ones, and computations still increase rapidly with the expansion of receptive fields.

Another alternative on self-attention is to take advantage of a proxy memory bank which has the access to the entire sequence tokens [1,2,56]. By using a small number of induced proxies to compress the whole sequence, it diffuses the global context efficiently. VoxSet [12] adapts Set Transformer [19] to 3D object detection and exploits an induced set to model a set-to-set point cloud translation. With the help of the compressed global proxies and Conv-FPN, it obtains a global receptive field; nevertheless, as they admit, it is sub-optimal to set only a few latent codes as proxies for a large 3D scene, prone to impairing the representation of different point cloud structures and their correlations. Therefore, there remains space for a stronger solution.

In this paper, we present a novel Transformer network, namely Octree-based Transformer (OcTr), for 3D object detection. We firstly devise an octree-based learnable sparse pattern, i.e. OctAttn, which meticulously and efficiently encodes point clouds of scenes as shown in Fig. 1. The OctAttn module constructs a feature pyramid by gathering and applies self-attention to the top level of the feature pyramid to select the most relevant tokens, which are deemed as the octants to be divided in the subsequent. When propagating to the level below, the key/value inputs are restricted by the octants from the top. Through recursively conducting this process, OctAttn captures rich global context features by a global receptive field in a coarse-to-fine manner while reducing the quadratic complexity of vanilla self-attention to the linear complexity. In addition, for better foreground perception, we propose a hybrid positional embedding, which consists of the semantic-aware positional embedding and attention mask, to fully exploit geometry and semantic clues. Thanks to the designs above, OcTr delivers a competitive trade-off between accuracy and efficiency.

Our contribution is summarized in three-fold:

1. We propose OcTr for voxel-based 3D object detection, which efficiently learns enhanced representations by a global receptive field with rich contexts.

2. We propose an octree-based learnable attention sparsification scheme (OctAttn) and a hybrid positional embedding combining geometry and semantics.

3. We carry out experiments on the Waymo Open Dataset (WOD) and the KITTI dataset and report state-of-the-art performance with significant gains on far objects.

2. Related Work

2.1. 3D Object Detection from Point Clouds

There exist two prevailing point-cloud representations in 3D object detection, i.e. point-based and voxel-based.

The point-based methods [31,40,42,60] directly process raw point clouds in the irregular 3D space. As a pioneering attempt, F-Pointnet [32] employs instance segmentation in frustums to extract proposals. VoteNet [31] clusters objects from the surface in a deep Hough voting manner. PointRCNN [40] generates 3D RoIs with foreground segmentation and applies an RCNN-style [35] two-stage refinement. Different from PointRCNN, some existing methods [3,53,59] build a lightweight and efficient single stage 3D object detection framework. However, the current point-based methods still suffer from a large computation burden, which is not suitable for large-scale point cloud scenes.

The voxel-based ones [4,6,13,52,55,62,63] conduct voxelization on entire point clouds to construct regular grids. VoxelNet [63] exploits the voxel feature encoding layer with 3D convolutions to extract the feature of each voxel. SECOND [32] improves the model with sparse 3D convolutions, significantly increasing the speed of both training and inference. Pointpillars [18] compacts point clouds into vertical columns and encodes them with 2D CNNs. Several recent methods [13,38,54] also explore merging point-based and voxel-based networks into one framework for complementary features from different representations of point clouds. Unfortunately, they all use small convolution kernels with limited receptive fields, which are not competent to capture global context that is important to 3D detection.
2.2. Transformer in 3D Vision

Inspired by the great success of the self-attention mechanism in NLP [47] and CV [8, 23], Transformers have been adapted to 3D vision for their ability to capture long-range dependencies. For instance, Point Transformer [61] brings in vector attention that modulates individual feature channels for point cloud classification and segmentation; PCT [11] presents offset-attention with the implicit Laplace operator and normalization refinement which is more suitable for point cloud learning. To address the high latency, some methods [30, 57] adopt voxels or patches for acceleration.

For 3D detection, 3DETR [28] treats and predicts bounding boxes as sequences in an end-to-end manner. CT3D [37] leverages a channel-wise Transformer architecture to refine the RoI head. To learn context-aware representations, some studies [9, 12, 24, 29] introduce Transformers into a point- or voxel-based encoder. PointFormer [29] stacks local, global and local-global Transformers based on the point-based encoder; VoTr [24] exploits dilated attention with fast query to enlarge receptive fields; VoxSet [12] builds an induced point set as proxies of global context and applies point-to-point translation using voxels as mediums; and SST [9] embraces single strides without down-sampling and conducts window attention combining with its shifted version. Even though they all expand receptive fields by diverse Transformer variants, global context is not adequately involved or efficiently utilized. In contrast, we propose Octree-based Transformer (OcTr) for voxel-based 3D object detection, achieving a true global receptive field that balances accuracy and efficiency.

3. Method

3.1. Framework

This sub-section describes the overall framework of the proposed OcTr model as shown in Fig. 2. Specifically, as inspired by [51], we first voxelize the point cloud into regular grids and adopt the sparse 3D convolution for patch embedding, where the grids are regarded as the “tokens” and are passed through the Octree Transformer Blocks (OTB). Compared with the vanilla Transformer block, the self-attention module is substituted by the proposed octree-attention OctAttn, which encodes global context in a more efficient way. After applying a hybrid semantic embedding on multi-scale features, we sequentially stack two OTBs, tailored by a down-sampling layer. The voxel features are then projected into the BEV view by point-wise convolution and are passed through a multi-scale dense 2D backbone. Ultimately, an anchor-based or anchor-free RPN head is used for 3D proposal generation, and the RoI head is optional for refinement. Note that our OcTr can be adopted for most of the voxel-based detection frameworks by simply altering 3D backbones.

3.2. Self-attention Revisit

According to [47], Transformer encoder blocks typically include a multi-head self-attention (MHSA) mechanism, a feed-forward network (FFN), a normalization function, and the residual connections [14]. Given an input sequence \( X \), the principle part of MHSA is formulated as

\[
\text{MHSA}(X) = \sum_{h=1}^{H} W_h [\sigma(W_q X W^T + W_k X^T) \cdot X W_v],
\]

where \( h \) denotes the index of the head, and \( H, \sigma, W \) and \( d \) are the amount of heads, softmax function, learnable weight and feature dimension, respectively. The subscripts of \( q, k \) and \( v \) indicate query, key and value. The inputs and outputs of the MHSA module are connected by residual connectors and normalization layers. The MLP-based FFN connects its inputs/outputs in a similar manner.

3.3. Octree Attention

An octree is a multi-scale asymmetric and efficient representation for unstructured 3D data such as point clouds. To build an octree for the input point cloud, we recursively sub-divide it in the breadth-first order until the pre-defined octree depth is reached. Whether to sub-divide an octant is determined by the occupancy [36, 48], the surface approximation [44, 49] or a learning algorithm [26].
Despite of being resource-friendly, the octree representation is discrete and non-differentiable, which motivates us to ameliorate previous cumbersome pre-processing division by introducing a novel octree-based attention mechanism. As depicted in [21,45], the attention matrices calculated by self-attention imply the relevance of input tokens and guide feature selection. We thus prune the dense attention matrices of the multi-scale feature pyramid to sparse octree attention in an adaptive and parallel manner, namely OctAttn.

As shown in Fig. 3, let the output feature map and coordinates of convolutional patch embedding be $F_0 \in \mathbb{R}^{M \times d}$ and $I_0 \in \mathbb{R}^{M \times 3}$; $M$ and $d$ indicate the amount of the non-empty grids in a batch and feature dimension, respectively. Based on $F_0$, we generate a multi-scale feature pyramid as

$$C = \{F_n, I_n\}_n, n \in [0, N),$$

(2)

where

$$I_n = \lfloor I_0 \frac{1}{2^n} \rfloor, \quad F_n = \text{BN}(S_{\text{max}}(F_0, I_n)),$$

(3)

$n$, $N$, BN and $S_{\text{max}}$ are the index of the level of the multi-scale feature pyramid, the height of the pyramid, batch normalization [16] and the max scatter function, respectively.

The pruning begins from the top of the pyramid. The top feature map, i.e. $F_{N-1}$, is reorganized to dense input tokens $\bar{F}_{N-1} \in \mathbb{R}^{B \times m_{N-1} \times d}$, where $B$ and $m_{N-1}$ are the batch size and maximum number of non-empty voxels per batch, respectively. The voxel is padded with 0 if it is empty. As shown in Eq. (4), the MHSA takes $\bar{F}_{N-1}$ as input and outputs the attention score matrices $A_{N-1} \in \mathbb{R}^{B \times m_{N-1} \times m_{N-1}}$ and attentive features $\bar{F}_{N-1}'$ as

$$A_{N-1} = \sum_{h=1}^{H} \sigma(\bar{F}_{N-1} W_q W^T_{k} F_{N-1}' \sqrt{d}),$$

$$\bar{F}_{N-1}' = \sum_{h=1}^{H} W_h \sigma(\bar{F}_{N-1} W_q W^T_{k} F_{N-1}' \sqrt{d}) \cdot \bar{F}_{N-1} W_e).$$

(4)

For each query token, we select the top$k$ attention scores as its most relevant token group in a row-wise way, denoted by $O_{N-1} \in \mathbb{R}^{B \times m_{N-1} \times k}$.

When propagating from top to bottom through the pyramid and reaching the $n$-th level, we uniformly sample limited $K$ attending octants from the selected regions with features in $\bar{F}_n$ and top$k$ indices in $O_{n+1}$. We conduct cross-attention instead of self-attention with the dense query sequence of the $n$-th level, $\bar{Q}_n \in \mathbb{R}^{B \times m_n \times d}$, and the compact sampled key/value sequence of the $n$-th level, $\bar{K}_n/\bar{V}_n \in \mathbb{R}^{B \times m_n \times K \times d}$, which is formulated as below

$$A_n = \sum_{h=1}^{H} \sigma(\bar{Q}_n W_q W^T_{k} \bar{K}_n \sqrt{d}),$$

$$\bar{F}_n' = \sum_{h=1}^{H} W_h \sigma(\bar{Q}_n W_q W^T_{k} \bar{K}_n \sqrt{d}) \cdot \bar{V}_n W_e).$$

(5)

Back ing off the sampling, this can be treated as an attention mask on self-attention matrices. The above process is run recursively until reaching the bottom level of the pyramid.

Furthermore, as the top$k$ selection is a hard decision that disables gradient back-propagation, we adopt the Gumbel-top$k$ technique [17] to perform a differentiable and continuous approximation by replacing the vanilla top$k$ selection.
The normalized scores used in topk are derived from the distribution in Eq. (6) during training, maintaining the original ones during inference. , , denote the noise sampled from the Gumbel distribution, the temperature and the amount of non-empty voxels in layer , respectively.

\[
p_i = \frac{\exp((A_i^i + g_i)/\tau)}{\sum_i \exp((A_i^i + g_i)/\tau)} \in [0, 1],
\]

In order to leverage the multi-scale features in distinct spatial shapes, we concatenate them by upsampling, which is implemented by inverse indices of the scatter function, followed by a linear projection layer for aligning the input feature dimension.

As the local context is generally critical for object detection, inspired by [7], we additionally introduce a Locally enhanced Positional Embedding (LePE) which enables local neighbor interactions on the value sequence. With sub-manifold sparse convolutions, we replace the residual connections in the attention mechanism with LePE.

Finally, OTB is formulated as follows

\[
\begin{align*}
\hat{F} &= FC(\{F_{N-1}', F_{N-2}', \ldots, F_0'\}) + LePE(F_0), \\
\tilde{F} &= BN(FFN(\hat{F})) + \hat{F},
\end{align*}
\]

where \( F_{N,i}' \in \mathbb{R}^{m_o \times d} \) denotes the compact tensor of the up-sampled \( \tilde{F}_{N,i} \), \( \Gamma \) indicates concatenation, and FC denotes the fully-connected layer.

Besides, we analyze the time complexity of the octree attention as below

\[
O((\omega^{N-1})^2 + \sum_{n=0}^{N-2} KM \omega^n) = \mathcal{O}(\omega^{N-1})^2 + \omega - 1 KM (1 - \omega^{1-N}),
\]

where \( \omega \) is the average down-sampling ratio in the sparse voxel representation.

### 3.4. Semantic Positional Embedding

Due to the large proportion of background grids in point clouds, the attention matrices are dominated by background grid pairs, leading to a sub-optimal solution. To fully leverage the local 3D shape patterns and original voxel coordinates, we propose a hybrid positional embedding to capture both the geometry and semantic clues as displayed in Fig. 4.

Specifically, we first segment foreground grids using the supervision from the ground truth. Segmentation scores are predicted by a sub-manifold sparse convolution branch with a sigmoid function; and the focal loss [22] is applied to balance the foreground and background.

![Figure 4. Illustration of semantic positional embedding, where an extra foreground segmentation branch is adopted with supervision and semantic scores are concatenated as absolute positional embeddings and serve as relative masks on attention matrices.](image-url)
Table 1. Performance on WOD with 202 validation sequences for vehicle (IoU=0.7), pedestrian (IoU=0.5) and cyclist (IoU=0.5), using 20% samples for training. All the results are achieved by the models simultaneously trained for 3 classes on single frames, except the ones of the model marked by †, which is only trained for a single class. Refer to Supp. B for the results trained with 100% samples.

| Model                  | Vehicle (L1) mAP|mAP/mAPH | Vehicle (L2) mAP|mAP/mAPH | Pedes. (L1) mAP|mAP/mAPH | Pedes. (L2) mAP|mAP/mAPH | Cyclist (L1) mAP|mAP/mAPH | Cyclist (L2) mAP|mAP/mAPH |
|------------------------|----------------|----------|-----------------|----------------|------------------|----------------|------------------|-----------------|----------------|----------------|
| SECOND [52]            | 70.96/70.34    | 62.58/62.02 | 65.23/54.24     | 57.22/47.49   | 57.13/55.62      | 54.97/53.53   |                  |                 |                 |                 |
| PointPillar [18]       | 70.43/69.83    | 62.18/61.64 | 66.21/46.32     | 58.18/40.64   | 55.26/51.75      | 53.18/49.80   |                  |                 |                 |                 |
| PartA²Net [41]         | 74.82/74.32    | 65.88/55.42 | 71.76/63.64     | 62.53/55.30   | 67.35/66.15      | 65.05/63.89   |                  |                 |                 |                 |
| PVR-CNN [38]           | 75.41/74.74    | 67.44/66.80 | 71.98/61.24     | 63.70/53.95   | 65.88/64.25      | 63.39/61.82   |                  |                 |                 |                 |
| CenterPoint [55]       | 73.55/70.70    | 63.16/62.65 | 72.09/65.49     | 64.27/58.23   | 68.67/67.39      | 66.11/64.87   |                  |                 |                 |                 |
| LiDAR-RCNN [20]        | 73.57/73.00    | 64.76/64.24 | 71.25/58.71     | 63.15/51.7    | 68.66/66.9       | 66.11/64.4    |                  |                 |                 |                 |
| Voxel-RCNN [6]         | 75.59/-        | 66.59/-    | -/-             | -/-           | -/-              | -/-           |                  |                 |                 |                 |
| PVR-CNN++ [39]         | 77.82/77.32    | 69.07/68.62 | 77.99/71.36     | 69.92/63.74   | 71.80/70.71      | 69.31/68.26   |                  |                 |                 |                 |
| SST† [9]               | 76.22/75.79    | 68.04/67.64 | 81.39/74.05     | 72.82/65.93   | -/-              | -/-           |                  |                 |                 |                 |
| PDV [15]               | 76.85/67.33    | 69.30/68.81 | 74.19/65.96     | 65.05/58.28   | 68.71/67.55      | 66.49/65.36   |                  |                 |                 |                 |
| Ours                   | 78.12/77.63    | 69.79/69.34 | 80.76/74.39     | 72.48/66.52   | 72.58/71.50      | 69.93/68.90   |                  |                 |                 |                 |

Table 2. Results on the WOD validation set in different ranges for vehicle detection.

4.1. Datasets and Implementation Details

WOD [43] is a large dataset of autonomous driving scenes. It totally contains 798 training sequences with around 160K LiDAR samples and 202 validation sequences with 40K LiDAR samples, with the mean Average Precision (mAP) and mAP weighted by heading accuracy (mAPH) as evaluation metrics. There are also two levels of difficulty describing the sparsity in each bounding box, and LEVEL_1 (L1) and LEVEL_2 (L2) denote more than 5 points and 1-5 points, respectively. For detection performance along distance, it provides mAP/mAPH on 0-30m, 30m-50m and 50m-inf.

KITTI [10] is a widely used benchmark for 3D object detection, which includes 3,712, 3,769 and 7,518 frames for training, validation and testing, respectively. mAP is used as the official metric with 11 recall points for the val set and 40 for the test set, and the IoU thresholds are set to 0.7, 0.5, and 0.5 for car, pedestrian and cyclist. We use the official setting in all experiments.

Implementation Details The total loss for optimizing the overall two-stage detection is formulated as Eq. (11), where $\mathcal{L}_{rcnn}$ can be omitted if there is no RoI head. Refer to Supp. A.1 for more information.

$$\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{rcnn} + \mathcal{L}_{seg}. \tag{11}$$

4.2. Results on WOD

The results on the validation set are displayed in Tab. 1, and it can be seen that we achieve new state-of-the-art performance on all the three classes. In particular, for pedestrian, we outperform the baseline model PV-RCNN++ [39] by 2.77%/2.56% in terms of both L1 and L2 mAP, which indicates the effectiveness of the proposed model in handling hard examples.

In comparison with other Transformer-based models, we focus on vehicle since the counterparts only report the performance on it. As Tab. 2 shows, our OcTr achieves the best mAP among all these convolution- and Transformer-based backbones. It also outperforms the Transformer-based detection head network CT3D [37] by 2.52% and 1.46% in L1 and L2 mAP. Regarding the accuracies at different distances, OcTr ranks the first place in the range of 30m-50m and 50m-inf, which surpasses the previous best by 0.45%, 2.66% in L1 mAP and 1.18%, 2.26% in L2 mAP respectively. It clearly illustrates that OcTr has the advantage in capturing long-range fine-grained context, which facilitates dealing with objects far away. Indeed, far objects generally have much more sparse points than near ones and heavily rely on context for detection, thus benefiting more from OcTr. Refer to Fig. 5 for visualization.
### 4.3. Results on KITTI

The performance of the single-stage detectors is shown in Tab. 3. We take SECOND [52], a commonly used anchor-based model, as the baseline, and compare OcTr with another two advanced Transformer-based variants VoxSet [12] and VoTR [24]. We can see that OcTr achieves comparable results with SECOND and VoxSet in car, while it significantly outperforms all the counterparts and reports the best performance both in pedestrian and cyclist. Benefiting from the large receptive field and fine-grained global context, OcTr exceeds SECOND by 4.19% and 3.29% for pedestrians and cyclists respectively, where hard samples often appear.

We summarize the performance of the two-stage models on the KITTI test set in Tab. 4. With the help of the multi-scale backbone features and rich global context, OcTr reaches a leading mAP in car at the moderate level, surpassing the state-of-the-art Focals-Conv [4] by 0.36%. We also evaluate OcTr on the KITTI val set, and OcTr again delivers the best performance in the average score, outperforming the second-best by 0.49%. One can observe that we rank the best or the second best in all the cases.

### 4.4. Ablation study

**Scalability on various detectors** As summarized in Tab. 5, we conduct experiments on three different and representative detectors, SECOND (single-stage, anchor-based), PV-RCNN (two-stage, anchor-based) and PV-RCNN++ (two-stage, anchor-free). Regardless of the number of stages or region proposal network, we acquire sound improvements...
LEPE | SAPE | SAM | Veh. mAP (L1/L2) | Pedes. mAP (L1/L2)
---|---|---|---|---
✓ | 71.35/63.30 | 65.75/57.89
✓ | 72.34/64.32 | 66.56/58.62
✓ | 72.64/64.46 | 66.62/58.83
✓ | 72.86/64.40 | 67.79/59.90
✓ | 73.28/65.05 | 68.08/60.36

Table 7. Ablation on semantic positional embedding on the WOD validation set.

| top: number | Veh. mAP (L1/L2) | Pedes. mAP (L1/L2)
---|---|---
1 | 70.38/62.20 | 64.19/56.43
4 | 72.58/64.42 | 66.21/58.42
8 | 73.28/65.05 | 68.08/60.36
16 | 73.25/65.01 | 67.89/60.10

Table 8. Results of different $k$ values on the WOD validation set.

| Method | #Param. (M) | Latency (ms) | Memory (GB)
---|---|---|---
SECOND [52] | 5.3 | 48 | 2.3
VoTR-SSD [24] | 4.8 | 67 | 3.0
VoxSeT-SSD [12] | 3.0 | 37 | 3.6
OcTr-SSD | 2.9 | 64 | 2.5

Table 9. Resource costs of different backbones with single-stage detectors on the KITTI dataset, test on GTX2080Ti.

compared to the baselines of the sparse convolution backbones, highlighting its scalability.

**Ablation on OctAttn** We carry out additional experiments to make apple-to-apple comparison with several representative linear Transformer methods, including Performer [5] (kernel-based linear attention), ACT [27] (cluster-based linear attention), VoTr [24] (fixed patterns) and the Nearest-$K$ strategy. Tab. 6 lists the results, and our OctAttn clearly performs the best, showing its ability.

**Ablation on semantic positional embedding** We individually evaluate the contributions of LePE, SAPE and SAM with SECOND as the baseline detector on WOD in Tab. 7. By incorporating LePE, the L1/L2 performance is boosted by 0.99%/1.02% and 0.81%/0.73% on vehicle and pedestrian, illustrating its necessity. Furthermore, we separately verify the validity of SAPE and SAM. With semantic clues, we observe that SAPE increases by 0.3% L1 mAP on vehicle, while SAM provides an L1/L2 mAP improvement of 0.52%/0.08% and 1.23%/1.28% on vehicle and pedestrian. Finally, we simultaneously apply SAPE and SAM and construct the full model of OcTr, which gains 0.94%/0.73% and 1.52%/1.74% L1/L2 mAP on vehicle and pedestrian, showing its impact.

**Influence by top: $k$** Tab. 8 shows the performance of OcTr with various values of $k$. As we can see, the performance improves when $k$ becomes larger but quickly saturates. Due to the redundancy of the scanned scenes, we argue that only a few tokens with high relevance need to be subdivided to embrace fine-grained features.

**Analysis on model complexity** We compare OcTr with two recent Transformer-based models in terms of resource cost by keeping the same detection head. Tab. 9 shows that with the learnable octree attention mechanism, OcTr consistently maintains less model parameters and less memory occupancies than the counterparts. Regarding the inference speed, VoxSeT runs faster, but it should be noted that VoxSeT inputs with pillars which discard the height dimension, leading to inferior results. As for VoTR, we deliver a mild improvement in efficiency while bringing a large gain in performance.

![Figure 5. Visualization of results by OcTr on the WOD validation split. Blue/red indicates predicted/ground-truth bounding boxes](image)

### 5. Conclusion

This paper proposes a novel voxel-based approach to 3D object detection, namely OcTr. It aims to balance the fine-grained global representation and efficiency with acceptable resource costs. To this end, we propose a learned sparsification attention mechanism, OctAttn, which adaptively prunes the octants from the multi-scale feature pyramid in a top-to-bottom manner. Furthermore, we adopt a hybrid semantic-aware positional embedding based on foreground segmentation. Extensive experiments are conducted on WOD and KITTI, and OcTr reaches the state-of-the-art performance, validating its effectiveness.

**Acknowledgment**

This work is partly supported by the National Natural Science Foundation of China (No. 62022011 and No. 62202034), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2021ZX-04), and the Fundamental Research Funds for the Central Universities.
References


Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. 1, 2, 6, 7, 8


Yifan Zhang, Qingyong Hu, Guoquan Xu, Yanxin Ma, Jianwei Ma, and Yulan Guo. Not all points are equal: Learning highly efficient point-based detectors for 3d lidar point clouds. In *CVPR*, pages 18953–18962, 2022. 2


Hualian Sheng, Sijia Cai, Yuan Liu, Bing Deng, Jianqiang Huang, Xian-Sheng Hua, and Min-Jian Zhao. Improving 3d object detection with channel-wise transformer. In *ICCV*, pages 2743–2752, 2021. 3, 6, 7


