A Du-Octree based Cross-Attention Model for LiDAR Geometry Compression

Mingyue Cui[1], Mingjian Feng[1], Junhua Long[1,2], Daosong Hu[1], Shuai Zhao[1], and Kai Huang[1,2,†]

Abstract— **Point cloud compression is an essential technology for efficient storage and transmission of 3D data. Previous methods usually use hierarchical tree data structures for encoding the spatial sparseness of point clouds. However, the node context within the tree is not fully discovered since the feature space among nodes varies significantly. To address this problem, we innovatively represent the LiDAR points in a two-octree structure instead of using traditional single-octree coding, and then design the cross-attention model to capture the hierarchical features between different octrees, of which each octree incorporates a transformer-based deep entropy model and an arithmetic encoder. Besides, we introduce the untied cross-aware position encoding with principal component analysis and different projection matrices, which enhances the correlations over two octrees' attention feature embeddings. Experimental results show that our method outperforms the previous state-of-the-art works, achieving up to 8.2% Bpp savings on point cloud benchmark datasets with different lasers.**

I. Introduction

LiDAR is a remote sensing technology that can accurately capture the surface structure of the terrain. Benefiting from the high accuracy and resolution of 3D geometry information, LiDAR has been widely used in various fields [1], [2], [3], [4], such as virtual reality, autonomous vehicles, intelligent robotics, etc. However, due to the sheer volume of point cloud data, processing such data poses a serious challenge. Taking only the single Velodyne LiDAR of HDL64 as an example, it generates over 100,000 points per sweep and about 84 billion points per day [5]. Thus, how to achieve efficient point cloud compression has become a crucial issue.

Eliminating structural redundancy in point cloud compression is a fundamental and effective way. The MPEG group develops a standard point cloud compression method GPCC [6] that adapts a hand-crafted context-adaptive arithmetic encoder for bit allocation. Recently, Kaya *et al.* [7] introduce the voxel-based deep learning-based convolutional transforms for lossy point cloud geometry compression by refining the bounding volumes of voxelized point clouds geometry. Overall, these methods achieve notable compression results, but there still remains a massive quantity of redundant information hidden in these representations, which provides an opportunity for reduction in the bitrate.

In recent years, the octree-based compression method has attracted great attention, due to its higher coding efficiency. However, how to fully consider the node context information in the octree is not easy. A highly refined octree with deep layers can code more point cloud data, but it leads to longer computational overhead. By contrast, a coarser octree with fewer layers can reduce memory usage and improve coding efficiency but may result in more loss of fine-grained details in the reconstructed point cloud. Besides, in the octree encoding process, spatial context information needs to be fully considered, especially for the neighbor geometric relationship between sibling nodes. Different from ancestor nodes, the neighbor geometric relationship of sibling nodes provides lower-level local geometry features, which are difficult to obtain by directly traversing the octree. Therefore, researchers have to consider the trade-off between reconstruction quality and coding time [8].

This paper proposes a novel du-octree (dual octree) based cross-attention model called DuOct, which can compress the LiDAR geometry efficiently. Considering the significant spatial feature differences between point cloud data, we divide it into coarse-grained inner and outer point cloud clusters and organize them as a two-octree structure. Due to the separation of point clouds with significant differences, the feature spaces within each tree (cluster) are more similar, reducing the complexity of recursive tree building. Subsequently, we develop the cross-attention transformer to better capture the hierarchical geometry features between two octrees. Finally, we propose the untied cross-aware position encoding to further extract the correlation between two octrees' self-attention embedding and position embedding with principal component analysis (PCA) and different projection matrices.

We compare the our method with state-of-the-art methods such as GPCC [6], VoxelContext [9], SibContext [10], OctAttention [11], SparsePCGC [12], and OctFormer [13] on SemanticKITTI [14] and NuScenes [15] large-scale datasets. Experimental results demonstrate that our method outperforms those methods, which are designed for the specific category of point clouds. Also, our method adapts to the downstream task with almost no impact on performance. Our main contributions are summarized as follows:

- We propose a du-octree structured cross-attention mechanism called DuOct to model the dependency of octree nodes, which efficiently exploits the spatial context

*This work was supported partly by the Grants from Natural Science Foundation of China (No. 62232008), partly by the Grants from Shenzhen Science and Technology Program (JCYJ20210324122203009), and partly by the Open Project Program for the Engineering Research Center of Software/Hardware Co-design Technology and Application (East China Normal University).

[1] The authors are with the School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China. (email:{cuimy, fengmj8, longjh7, huds, zhaosh56, huangk36} @mail2.sysu.edu.cn).

[2] J. Long and K. Huang are also with the Shenzhen Institute, Sun Yat-sen University, Guangzhou 510275, China.

[†] Corresponding author.

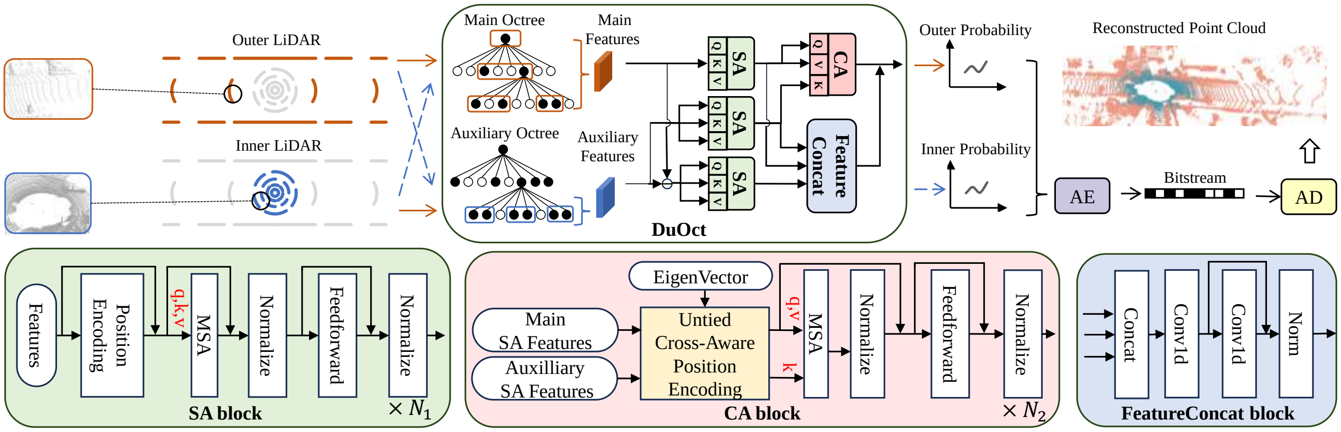


Fig. 1: The framework of DuOct, which mainly consists of three self-attention (SA) blocks for learning the octree's self feature, a cross-attention (CA) block for extracting two octrees correlation, and a feature concatenate block to fuse features.

information of point clouds.

- We innovatively design the du-octree coding structure, which can provide more geometric priors and reduce the coding time compared with the single-octree.
- To further capture the hierarchical geometry features, we develop the cross-attention Transformer to fuse prior knowledge features of the traversal format from sibling.
- Our DuOct uses untied cross-aware position encoding with PCA and different projection matrices to extract the correlations, which enhances generalization ability.

II. Related Work

A. Voxel-based Methods

Voxel-based methods [16], [17], [18], [19] represent point clouds as voxel grids and apply transform coding techniques to compress the voxel values and classify the voxel occupancy. Wang *et al.* [20] propose a voxel-based end-to-end neural network that voxelizes point clouds into non-overlapped 3D cubes and applies a 3D-convolution-based variational autoencoder to compress the point clouds. VoxelFPN [21] further proposes a one-stage end-to-end trainable deep architecture for multi-scale voxel partitions. Compared with octree coding, although the voxel can naturally retain geometric patterns of point clouds, it is sensitive to density variation and also brings high computational costs.

B. Octree-based Methods

Octree-based methods have higher coding efficiency through hierarchical representation. Garcia *et al.* [22] propose an intra-frame context-based octree coding method utilizing incorporating spatial correlation and an adaptive rate-distortion optimization algorithm. MuSCLE [23] reduces the temporal redundancy by exploiting spatio-temporal relationships across LiDAR sweeps and traverses octree from both top and bottom for better compression ratio. Huang *et al.* [5] propose the first octree-based deep learning entropy model OctSqueeze modeling the dependency among the node and its multiple ancestor nodes. The method avoids high computational complexity, yet the strong dependency among sibling nodes is ignored.

Recently, VoxelContext [9] and SibContext [10] extract sibling context information by combining voxel with octree coding. The former utilizes a 3D convolution-based deep entropy model to encode the neighboring spatial information for each node in the constructed octree. The latter fits quadratic surfaces with a voxel-based geometry-aware module to provide geometric priors in entropy encoding. OctAttention [11] employs a conditional entropy model to discover neighboring node dependencies, with a large receptive field and a mask operation for encoding multiple nodes. SparsePCGC [12] further adopts the convolutional representation of multiscale sparse tensor for point cloud geometry compression, which exploits correlations through cross-scale context modeling. OctFormer [13] optimizes the frequent multi-head self-attention (MSA) operation from OctAttention by sharing the results within constructed non-overlapped windows.

Overall, compared with the above state-of-the-art methods, our proposed DuOct has the following advantages: **a)** Compared with VoxelContext and SibContext, our proposed method does not need to introduce computationally expensive 3D convolution on generated voxel grids and has a bigger receptive field of neighborhood information. **b)** Compared with OctAttention, SparsePCGC, and OctFormer, our proposed cross-attention model with a du-octree based structure can further exploit spatial context information and capture the hierarchical geometry features, benefiting from the separation of point clouds with significant differences.

III. Method

As shown in Fig.1, we propose a du-octree based cross-attention entropy model DuOct for LiDAR geometry compression. First, the LiDAR data is divided into inner and outer point clouds based on the differences in feature space, and two trees are simultaneously constructed level by level. The outer and inner octrees take turns to be represented as the main octree. Each non-leaf node contains the feature of its xyz coordinate, depth, parent occupancy, and index. Then we apply several multi-head self-attention models to learn the content information for two octrees. To discover the correlation between the two octrees and achieve accurate prediction, we further propose a cross-attention block with untied cross-aware position encoding, which further captures the hierarchical geometry features between the octrees. During entropy coding, the arithmetic encoder (AE) synthesizes them into a bitstream. Finally, the arithmetic decoder (AD) decompresses the bitstream into two octrees, from which the point clouds can be concatenated back into a complete octree and reconstructed.

A. *Du-octree Coding*

Octree divides the 3D space into 8 cubes along the maximum side length of the bounding box, and then recursively divides each non-empty cube in the same way until reaching the set maximum depth. Each non-leaf node uses an 8-bit occupancy to represent the distribution of children nodes and each bit corresponds to one specific child.

In the octree construction procedure, the coordinate of the cube center is used to represent each node. However, there is a quantization error e between the coordinate $\hat{p_i}$ of the cube center and the corresponding point p_i in the raw point cloud P, which can be described as:

$$e = \max_i \|p_i, \hat{p_i}\|_\infty \leq \frac{L}{2^h} \tag{1}$$

where L is the length of the bounding box and h represents the maximum depth level in the octree structure. For octree-based methods, the geometric compression loss only comes from quantization error in Eq. 1. Researchers usually increase the octree depth h to achieve highly accurate compression quality. However, limited by coding time, single-tree coding based methods [10], [11] usually set the maximum depth to 12. In this paper, we divide the point cloud into coarse-grained inner and outer point clusters and use different partition scales to construct octrees. Actually, we explore the possibility of improving the partitioning resolution by reducing the size of the bounding box (for inner point clusters), as opposed to simply increasing the octree depth.

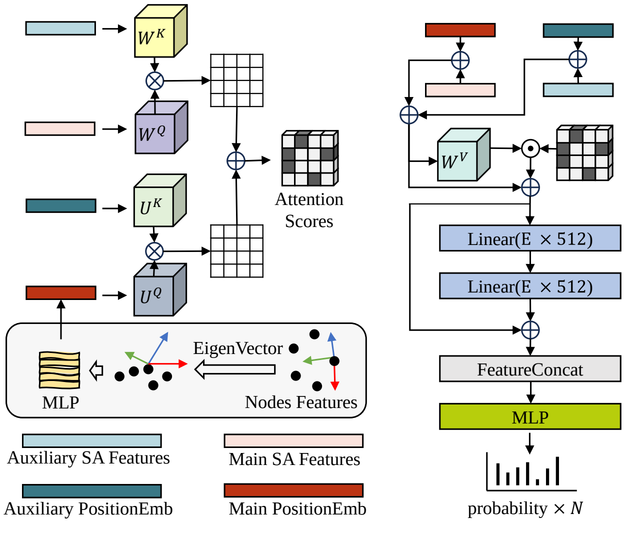
B. *Context Model*

We use $X = [x_1, x_2, ..., x_i, ..., x_n]$ to represent a sequence of occupancies for octree nodes, where x_i is the occupancy of i_{th} octree node. Each occupancy $(1-255)$ is represented with an 8-bit code. Our DuOct constructs a parametric probability distribution $Q(X)$ to approximate the actual distribution $P(X)$ of the sequence.

According to the entropy coding theory [24], the smaller distortion between the actual probability distribution $P(X)$ and the predicted $Q(X)$, the fewer actual compressed bitrate $\mathbb{E}_{X\sim P}[-log_2 Q(X)]$ is to its lower bound. Thus the goal of the entropy model is to minimize the cross-entropy loss between the $P(X)$ and $Q(X)$. $Q(X)$ is factorized into a product of predicted probability distributions of each node's occupancy x_i:

$$Q(X) = \prod_i q_i(x_i|\mathbf{f_i}, \widetilde{\mathbf{f_j}}; w) \tag{2}$$

where $q_i(x_i|\mathbf{f_i}, \widetilde{\mathbf{f_j}}; w)$ is the estimated distribution of octree nodes occupancy x_i. w is the weight of the entropy model.



(a) Untied Cross-Aware Position Encoding

(b) Cross-attention

Fig. 2: The structure of the cross-attention based Transformer model. The model uses the node features to generate queries and key-value pairs for the cross-attention operation. The cross-aware embeddings are projected with different weight matrices to form the attention scores matrices.

Benefiting from the du-octree structure, we assume that the predicted distribution x_i depends on the local octree nodes sequence feature $\mathbf{f_i}$ and the sequence feature $\widetilde{\mathbf{f_j}}$ from the auxiliary octree. $\mathbf{f_i}$ is composed of a sequence of node features $[f_{i-j}, ..., f_i, ...f_{i+k}]$, where f_i denotes the feature of the i_{th} octree node. Specifically, f_i contains the xyz coordinates, index (0-7), depth (1-12), and parent occupancy (1-255). $j + k - 1$ is the local sequence size.

C. *Du-octree based Cross-Attention Model*

1) Self-Attention Model: We design the self-attention model for aggregating the node features among the same depth. By traversing the octree with a set sequence size E, the input of the self-attention model can be organized as several sequences. We increase the sequence $\mathbf{f_i} \in \mathbb{R}^{E\times 6}$ feature dimension from 6 to 512 after the embedding layer, and add positional encoding $pe \in \mathbb{R}^{E\times 512}$ to the input sequence to provide the position information within a sequence. Based on the MSA mechanism [25], several Transformer blocks are applied to extract features for every node and generate the self-attention feature $\mathbf{s_i} \in \mathbb{R}^{E\times 512}$. $\mathbf{s_i}$ reflects the nodes' salient characteristics among local siblings with the same spatial splitting scale.

2) Untied Cross-Aware Position Encoding: Considering that the self-attention feature usually lacks the description of the geometric splitting information between multiple levels of the overall octree, we propose an untied cross-aware position encoding module to learn the correlation of features between multiple depths of octrees. As shown in Fig 2, we use PCA to compute eigenvectors, for which the basis vectors correspond to the maximum-variance directions in the auxiliary geometry feature space [26]. These eigenvectors are then employed to project the initial features into the feature subspaces of the auxiliary octree to generate the significant cross-aware position embeddings pe_{cross}.

Instead of adding the cross-aware position embeddings and self-attention features to generate attention score matrices, our position encoding is untied. Considering that the position embeddings and self-attention features have significantly different concepts, we apply different projection matrices to them for generating different query-key attention score matrices, which is beneficial for extracting correlations and enhancing generalization ability [27]. The details of the untied cross-aware position encoding are shown in Fig 2. We adopt different weights W^K, W^Q, U^K, and $U^Q \in \mathbb{R}^{512 \times 512}$ to generate two matrices and add to form the attention score matrices of $\mathbb{R}^{E \times E}$.

3) Cross-Attention Transformer: As shown in Fig 2, we design the cross-attention based Transformer model to fuse the features between the main octree and auxiliary octree. The attention score matrices are multiplied with values to generate the cross-attention feature. In the head t, the attention score is calculated as follows:

$$\begin{aligned} AttentionScore^{(t)} = softmax(& \frac{W^Q \mathbf{s_i} \cdot W^K \widetilde{\mathbf{s_j}}}{\sqrt{H}} \\ & + \frac{U^Q pe_{cross} \cdot U^K pe_{aux}}{\sqrt{H}}) \end{aligned} \tag{3}$$

where $\mathbf{s_i}$, $\widetilde{\mathbf{s_j}}$ are the self-attention feature from the main octree and the auxiliary octree respectively, pe_{cross} and pe_{aux} are position embeddings, and H denotes the dimension of the keys. According to the attention mechanism, the cross-attention context can be described as:

$$\begin{aligned} \mathbf{s_i}' &= \mathbf{s_i} + pe_{cross} \\ \widetilde{\mathbf{s_j}}' &= \widetilde{\mathbf{s_j}} + pe_{aux} \\ C^{(t)} &= AttentionScore^{(t)} \cdot W^V(\mathbf{s_i}' + \widetilde{\mathbf{s_j}}') \end{aligned} \tag{4}$$

The cross-attention model expands the receptive field from a layer in the octree to the auxiliary octree and establishes feature associations between hierarchical multi-scale features. Besides, after performing cross-attention calculation, a forward propagation with two linear layers and a residual connection is used to generate cross-attention features. We further apply feature concatenate block and multi-layer perceptrons (MLP) with 2 linear layers to obtain the octree node's occupancy probability distribution. The output of the cross-attention-based Transformer model is as follows:

$$\begin{aligned} F = \mathbf{s_i}' + \widetilde{\mathbf{s_j}}' + Norm(W^M([C^{(1)}, C^{(2)}, \ldots, C^{(t)}])) \\ F' = Norm(Linear(Linear(F)) + F) \\ F' = FeatureConcat(F', \mathbf{s_j}, \widetilde{\mathbf{s_j}}) \\ q_i(x_i | \mathbf{f_i}, \widetilde{\mathbf{f_j}}; w) = MLP(F') \end{aligned} \tag{5}$$

where W^V and W^M represent the weight matrices, $Norm$ is layer normalization [28] for faster convergence.

D. Loss

The proposed deep entropy model is optimized with the cross entropy between the real and predicted occupancy of the non-leaf node $X = [x_1, x_2, ..., x_i, ..., x_n]$. The loss function is as follows:

$$Loss = -\sum_i \log q_i(x_i | \mathbf{f_i}, \widetilde{\mathbf{f_j}}; w) \tag{6}$$

where $q_i(x_i | \mathbf{f_i}, \widetilde{\mathbf{f_j}}; w)$ is the estimated occupancy distribution of octree node occupancy x_i.

IV. Experiments

A. Dataset

1) SemanticKITTI: SemanticKITTI [14] is a famous large-scale outdoor dataset, collected by 64-lasers LiDAR. The dataset consists of 22 sequences with a total of 43,504 LiDAR scans. The point clouds are captured at a rate of 10 Hz, and each scan contains 128,000 points. We use sequences 00-10 for training and 11-21 for testing, following the standard split.

2) NuScenes: NuScenes [15] is a commonly used large-scale dataset for autonomous driving, collected by 32-lasers LiDAR. The dataset consists of 1,000 scenes with a total of 390,000 LIDAR scans. The point clouds are captured at a rate of 2 Hz, and each scan contains 34,700 points. We randomly sample 1200 frames from each of the first five data batches (total 6,000) for training, and randomly sampled 100 frames from each of the last five data batches (total 500) for testing, following the split from SibContext [10].

B. Experimental Details

1) Baseline Methods: We evaluate our proposed method by comparing it against state-of-the-art methods GPCC [6], VoxelContext [9], SibContext [10], OctAttention [11], SparsePCGC [12], and our previous work (OctFormer [13]). These methods are also designed for a specific category of point clouds. We keep our training/testing setting consistent with them for a fair comparison. For some results, we use the results reported in their respective papers directly.

2) Evaluation Metrics: We use the point-to-point PSNR (D1 PSNR) and point-to-plane PSNR (D2 PSNR) to evaluate the point cloud reconstruction quality and bits per point (Bpp) as the compression ratio metric [29]. We also report chamfer distance (CD) [30], [31]. We use the official metric calculating tool *pc_error* provided by MPEG's GPCC and set the PSNR peak value *r* = 1 following [9] and [11]. We normalize the point cloud data to $[-1, 1]^3$. Unless otherwise specified, all distortion curves and bitrates are obtained by averaging over sequence.

3) Implementation Details: DuOct is implemented in Pytorch and train/test on a machine with Xeon Gold 6234 CPU and a single NVIDIA RTX 8000 GPU (48GB Memory). The Adam optimizer is adopted and the learning rate is 1e-4 for the entropy model. It takes 5 days on SemanticKITTI and 3 days on NuScenes to train. We set the embedding size and the feed-forward dimension in the Transformer block to 512, and the output dimension to 256, where the default Transformer context window size E is set to 1024. We use 14,2,2 layers and 8 heads for each MSA block and 8 layers and 8 heads for the multi-head cross-attention.

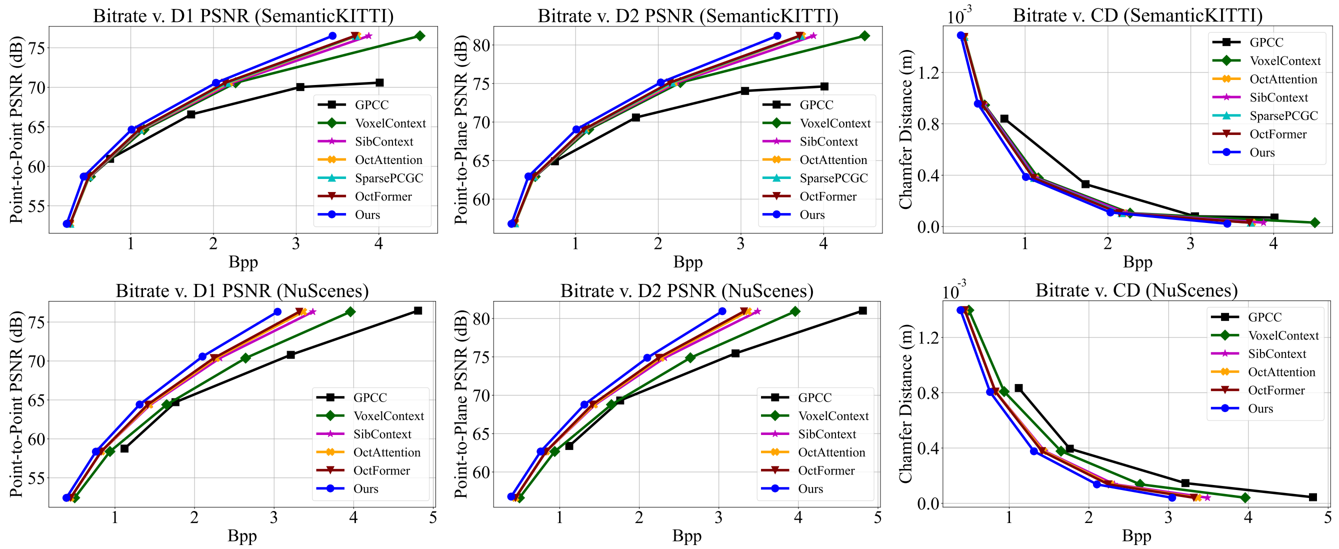


Fig. 3: Quality results of different compression methods on SemanticKITTI and NuScenes datasets at different bitrates. For a fair comparison, we only compare with SparsePCGC on SemanticKITTI dataset, in which data in their paper is used.

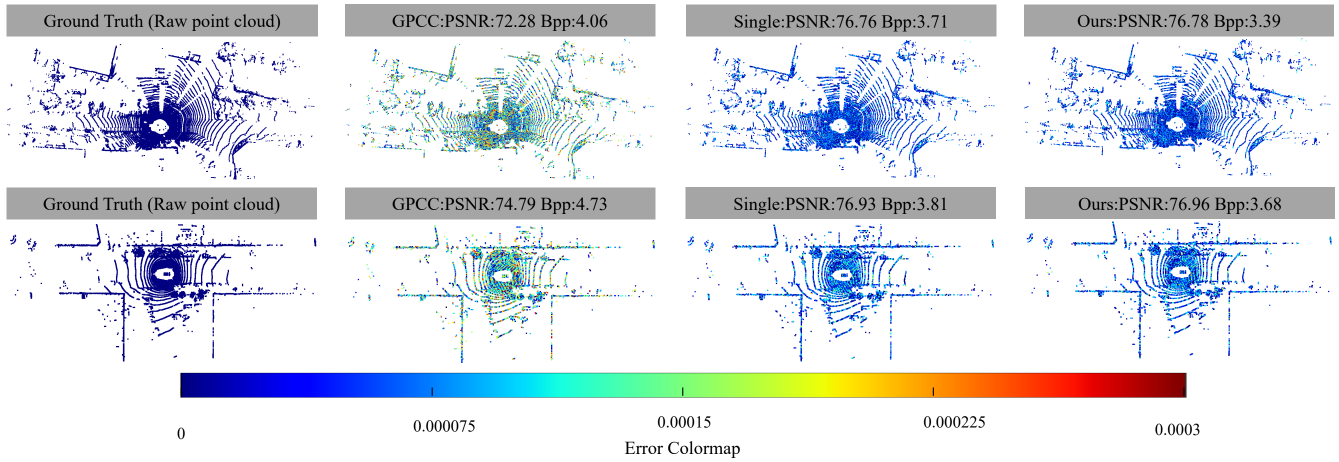


Fig. 4: Visualization of DuOct and other methods under different Bpps on SementicKITTI (upper) and NuScenes (bottom) datasets. **Considering that** under the same octree depth, octree-based methods have the same reconstruction qualities, we only list different categories from left to right: Ground Truth, GPCC, Ours with single-octree coding (Single), and Ours.

C. Main Results

The rate-distortion curves of LiDAR compression are shown in Fig. 3. For a fair comparison, we do not introduce the coordinate refinement module (CRM) for each baseline method, which is a module for post-processing and does not affect the compression ratio. From the figure, we can observe that our method outperforms other methods on both the SemanticKITTI and NuScenes datasets. Specifically, compared with GPCC and VoxelContext, we achieve up to 49.3% and 23.5% Bpp savings on SemanticKITTI and 36.7% and 23.1% Bpp savings on NuScenes. These methods do not fully consider the correlations within octree nodes, which limits the further reduction in the bitrate. Other baselines (OctAttention, SparsePCGC, and OctFormer) achieve similar compression performance. However, our method still achieves up to 8.2% Bpp savings compared with them. The reason is that our du-octree structure splits the point cloud data into coarse-grained inner and outer point cloud clusters to build octrees, and our cross-attention transformer efficiently captures the hierarchical geometry features between two octrees. Experimental results indicate the effectiveness of our DuOct model with a large receptive field. From Fig. 4, it can also be seen that our method is closer to the color of the raw point cloud, which means that our method maintains high compression quality while achieving higher compression rates. For the inference time, our model takes 0.017 seconds per 1000 octree nodes with the implementation settings.

D. Ablation Study

1) Effectiveness of Du-Octree Coding: As shown in Fig. 6 and Table. I, we compare the two octree coding methods “Single” and our Du-octree. As expected, compared with the single-octree coding, the du-octree achieves significant coding time performance improvement, under the condition of coding the same number of points. For example, compared

TABLE I: Perform ablation study on coding structure and cross attention, in which ‘Y’ and ‘N’ denote retention and removal, respectively. For the depths n, the results of backbone ‘Single’ are from the octree with depth n, while the backbone ’Du-octree’ are from the outer octree with depth n and inner octree with depth $m = n - 2$.

Backbone	Single				Du-octree				
Cross Attention	-				Y				N
Metrics / Depths	Bpp	D1 PSNR	D2 PSNR	CD ($\times 10^{-4}$)	Bpp	D1 PSNR	D2 PSNR	CD ($\times 10^{-4}$)	Bpp
8	0.231	52.73	56.86	14.893	**0.228**	52.72	56.86	14.758	0.230
9	0.457	58.70	62.93	9.569	**0.432**	58.70	62.93	9.460	0.448
10	1.061	64.62	69.04	3.863	**1.014**	64.64	69.04	3.784	1.051
11	2.179	70.57	75.12	1.102	**2.031**	70.57	75.13	1.046	2.140
12	3.709	76.51	81.19	0.220	**3.443**	76.52	81.20	0.177	3.637

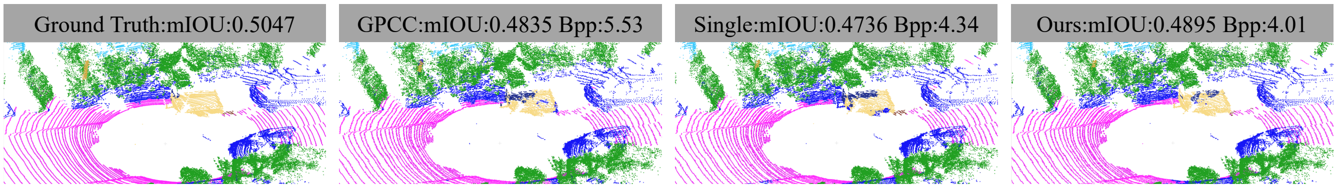

Ground Truth:mIOU:0.5047
GPCC:mIOU:0.4835 Bpp:5.53
Single:mIOU:0.4736 Bpp:4.34
Ours:mIOU:0.4895 Bpp:4.01

Fig. 5: Qualitative results of semantic segmentation between different methods.

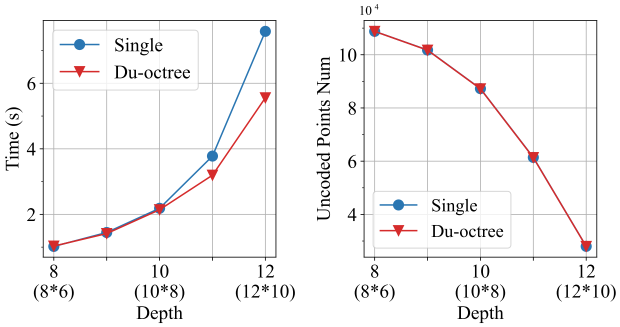


Fig. 6: Comparison of time performance and number of uncoded point clouds between single and du-octree structures on SemanticKITTI dataset. For du-octree coding, we represent the depth of two trees in the form of $n * m$, where n is the depth of the outer point cloud and m is the other.

with the single octree with a depth of 12, the du-octree with setting n and m to 12 and 10 obtains 36% tree-building time savings, and 7% Bpp savings. The main reason is that the point cloud of the divided inner tree is more compact, which means that high-precision point coding can be achieved with only shallow-depth trees.

2) Effectiveness of Untied Cross-Aware Position Encoding: As shown in Table. I, we also perform an ablation study on our proposed untied cross-aware position encoding. It can be seen that the Bpp of our model is obviously lower than the model without untied cross-aware position encoding. Specifically, we obtain up to 5% (from 3.637 to 3.443) Bpp savings when adding the untied cross-aware position encoding. The reason is that we use PCA to interact features between octrees and employ different projection matrices, which enhances better correlation extraction.

E. Performance on Application

Another important metric for compression is its effects on the performance of downstream tasks. As shown in Fig. 5 and Table. II, we quantify the effects for semantic segmentation [32], [33]. We use RandLA-Net [34] to evaluate the segmentation performance and apply intersection-over-union (IOU) as the metric. From the figure, our method can achieve segmentation performance close to the ground truth when the Bpp is 4.02 for the mean IOU and class ’car’ IOU metrics respectively. At any given Bpp, our method obtains higher IOU than GPCC and single-octree based method, which means our reconstructed point clouds preserve more fine-grained details. Overall, experimental results demonstrate the effectiveness of our method on the downstream task.

TABLE II: Comparison of segmentation performance between different methods at different bitrates.

Method	Bpp	Mean IOU	IOU in ’car’
Ground Truth	-	0.5047	0.9451
GPCC	0.8226	0.1763	0.5053
	1.8008	0.2819	0.6593
	3.2511	0.3907	0.8173
	5.5325	0.4835	0.9076
Single	0.5981	0.1748	0.3890
	1.3356	0.2763	0.6924
	2.5616	0.4319	0.8773
	4.3498	0.4736	0.9350
Ours	0.5553	0.1775	0.4078
	1.2770	0.2856	0.7163
	2.5159	0.4325	0.9021
	4.0178	0.4895	0.9375

V. Conclusion

In this paper, we propose a du-octree based cross-attention model called DuOct, which can compress the LiDAR geometry efficiently. In this model, we innovatively represent the LiDAR points in a two-octree structure instead of using traditional single-octree coding, which reduces the coding time. On this basis, we develop the cross-attention Transformer to further capture the hierarchical geometry features between two octrees. Experimental results demonstrate that our method outperforms other methods and we hope our DuOct can provide a new perspective for point cloud compression.

References

[1] Qin Zou, Qin Sun, Long Chen, Bu Nie, and Qingquan Li. A comparative analysis of lidar slam-based indoor navigation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6907–6921, 2022.

[2] You Li and Javier Ibanez-Guzman. Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4):50–61, 2020.

[3] Jian Zhao, Yaxin Li, Bing Zhu, Weiwen Deng, and Bohua Sun. Method and applications of lidar modeling for virtual testing of intelligent vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(5):2990–3000, 2021.

[4] Stefano Gasperini, Mohammad-Ali Nikouei Mahani, Alvaro Marcos-Ramiro, Nassir Navab, and Federico Tombari. Panoster: End-to-end panoptic segmentation of lidar point clouds. *IEEE Robotics Autom. Lett.*, 6(2):3216–3223, 2021.

[5] Lila Huang, Shenlong Wang, Kelvin Wong, Jerry Liu, and Raquel Urtasun. Octsqueeze: Octree-structured entropy model for lidar compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1313–1323, 2020.

[6] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai. An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc). *APSIPA Transactions on Signal and Information Processing*, 9:e13, 2020.

[7] Emre Can Kaya, Sebastian Schwarz, and Ioan Tabus. Refining the bounding volumes for lossless compression of voxelized point clouds geometry. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3408–3412, 2021.

[8] Antoine Dricot, Fernando Pereira, and João Ascenso. Rate-distortion driven adaptive partitioning for octree-based point cloud geometry coding. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2969–2973, 2018.

[9] Zizheng Que, Guo Lu, and Dong Xu. Voxelcontext-net: An octree based framework for point cloud compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6042–6051, 2021.

[10] Zhili Chen, Zian Qian, Sukai Wang, and Qifeng Chen. Point cloud compression with sibling context and surface priors. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 744–759, Cham, 2022. Springer Nature Switzerland.

[11] Chunyang Fu, Ge Li, Rui Song, Wei Gao, and Shan Liu. Octattention: Octree-based large-scale contexts model for point cloud compression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[12] Jianqiang Wang, Dandan Ding, Zhu Li, Xiaoxing Feng, Chuntong Cao, and Zhan Ma. Sparse tensor-based multiscale representation for point cloud geometry compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(7):9055–9071, 2023.

[13] Mingyue Cui, Junhua Long, Mingjian Feng, Boyang Li, and Huang Kai. Octformer: Efficient octree-based transformer for point cloud compression with local enhancement. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(1):470–478, Jun. 2023.

[14] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2020.

[15] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019.

[16] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015.

[17] Simone Limuti, Enrico Polo, and Simone Milani. A transform coding strategy for voxelized dynamic point clouds. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2954–2958, 2018.

[18] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. Learning convolutional transforms for lossy point cloud geometry compression. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4320–4324, 2019.

[19] Jianqiang Wang, Hao Zhu, Haojie Liu, and Zhan Ma. Lossy point cloud geometry compression via end-to-end learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12):4909–4923, 2021.

[20] Bhaskar Anand, Vivek Barsaiyan, Mrinal Senapati, and P. Rajalakshmi. Real time lidar point cloud compression and transmission for intelligent transportation system. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5, 2019.

[21] Hongwu Kuang, Bei Wang, Jianping An, Ming Zhang, and Zehan Zhang. Voxel-fpn: Multi-scale voxel feature aggregation for 3d object detection from lidar point clouds. *Sensors*, 20(3), 2020.

[22] Diogo C. Garcia and Ricardo L. de Queiroz. Intra-frame context-based octree coding for point-cloud geometry. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 1807–1811, 2018.

[23] Sourav Biswas, Jerry Liu, Kelvin Wong, Shenlong Wang, and Raquel Urtasun. Muscle: Multi sweep compression of lidar using deep entropy models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[24] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

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

[26] A.M. Martinez and A.C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[27] Guolin Ke, Di He, and Tie-Yan Liu. Rethinking positional encoding in language pre-training. In *International Conference on Learning Representations*, 2021.

[28] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[29] Sebastian Schwarz, Marius Preda, Vittorio Baroncini, Madhukar Budagavi, Pablo Cesar, Philip A. Chou, Robert A. Cohen, Maja Krivokuća, Sébastien Lasserre, Zhu Li, Joan Llach, Khaled Mammou, Rufael Mekuria, Ohji Nakagami, Ernestasia Siahaan, Ali Tabatabai, Alexis M. Tourapis, and Vladyslav Zakharchenko. Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148, 2019.

[30] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[31] Tianxin Huang and Yong Liu. 3d point cloud geometry compression on deep learning. In *Proceedings of the 27th ACM international conference on multimedia*, pages 890–898, 2019.

[32] Anh Nguyen and Bac Le. 3d point cloud segmentation: A survey. In *2013 6th IEEE conference on robotics, automation and mechatronics*, pages 225–230. IEEE, 2013.

[33] Qingyong Hu, Bo Yang, Guangchi Fang, Yulan Guo, Aleš Leonardis, Niki Trigoni, and Andrew Markham. Sqn: Weakly-supervised semantic segmentation of large-scale 3d point clouds. In *European Conference on Computer Vision*, pages 600–619. Springer, 2022.

[34] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11108–11117, 2020.